EXPERT INSIGHT

Mastering Palo Alto Networks

The complete journey to firewall mastery from setup to advanced security

Third Edition

Tom Piens aka 'reaper'

packt

Mastering Palo Alto Networks

Third Edition

The complete journey to firewall mastery from setup to advanced security

Tom Piens aka 'reaper'

<packt>

Mastering Palo Alto Networks

Third Edition

Portfolio Director: Vijin Boricha
Relationship Lead: Rahul Nair
Program Manager and Growth Lead: Ankita Thakur
Project Manager: Gandhali Raut
Content Engineer: Rebecca Youé
Technical Editor: Arjun Varma
Copy Editor: Safis Editing
Indexer: Rekha Nair
Proofreader: Safis Editing
Production Designer: Ajay Patule

First published: September 2020
Second edition: June 2022

Production reference: 1290525

Published by Packt Publishing Ltd.
Grosvenor House
11 St Paul's Square
Birmingham
B3 1RB, UK.

ISBN 978-1-83664-481-1
www.packtpub.com

Contributors

About the author

Tom Piens is a network security expert with over two decades of hands-on experience deploying, configuring, and securing enterprise environments. He specializes in Palo Alto Networks technologies, including advanced implementations of Strata and Prisma Access. As the author of multiple technical books, Tom focuses on translating complex security concepts into practical, real-world configurations.

Thanks to Lawrence Curling for the thorough technical review and Andy Soest for for his thoughtful pointers.

Special thanks to the Packt team for their relentless support throughout the process of writing this edition. Your vigilance helped make this edition even better than the last.

That's what I do,
I drink and I know things.

— Tyrion Lannister

About the reviewer

Laurence Curling is a network security specialist with over 16 years of experience spanning both traditional data center environments and cloud-native architectures. He brings a depth of practical knowledge in designing and managing secure infrastructures that meet the demands of modern enterprise networks.

His primary expertise lies within the Palo Alto Networks ecosystem, with a strong focus on the Strata portfolio. Laurence has overseen complex firewall migrations and upgrades, restructured legacy rulebases into application-aware policies, and implemented centralized administration through Panorama to improve visibility and control across distributed environments.

In addition to his work with next-generation firewalls, Laurence has developed proficiency in cloud-delivered security solutions through Prisma Access. His experience includes deploying and managing secure remote access and extending consistent security policy enforcement across hybrid networks.

Laurence is known for a thoughtful, solutions-oriented approach grounded in core security principles, with a track record of adapting to emerging technologies and evolving threat landscapes.

Table of Contents

Chapter 3: Building Strong Policies 89

Chapter 9: Upgrading Firewalls and Panorama 357

Chapter 10: Logging and Reporting 375

Preface

Palo Alto Networks firewalls are powerful, flexible, and deeply capable—but they're not always intuitive, especially when you're just getting started. Whether you're brand new to network security or already working in the field, this book is here to guide you through the ins and outs of Palo Alto's Strata suite, from the ground up.

We start at zero. No assumptions, no skipped steps. You'll learn how the technology works, how to configure it, and—just as important—how to troubleshoot it when things go sideways. The content covers everything up to PAN-OS 11.2, walking through core features, real-world deployment practices, and common pitfalls that tend to trip people up.

Though the focus is on the firewall, the skills you build here will give you a strong foundation for any role involving Palo Alto gear. Experienced readers will find best practices, deeper insights, and practical tips woven throughout, while new users will get a structured path to proficiency.

I've always enjoyed teaching through dialogue—answering real questions with real context. That mindset shaped how this book was written: not as a dry reference, but as a practical guide to help you work through challenges, build confidence, and actually *understand* what's going on under the hood.

If this book helps you solve a problem faster, avoid a bad config, or just see how it all fits together, then it's done its job.

Who this book is for

This book is for anyone looking to build solid, practical skills with Palo Alto Networks firewalls. No prior experience with Palo Alto gear is required—if you're starting from scratch, you'll be fine. A basic understanding of networking concepts (IP addressing, routing, NAT) will help, but even that is reviewed where needed.

It's also a useful resource for experienced firewall admins looking to sharpen their skills, adopt best practices, or get up to speed on PAN-OS 11.2. Whether you're studying for a certification, deploying in production, or just trying to understand what your firewall is actually doing, this book is for you.

What this book covers

Chapter 1, Understanding the Core Technologies, is an introduction to the logic behind the firewall, including the ins and outs of the Palo Alto Networks Strata suite.

Chapter 2, Setting up a New Device, walks you through, step by step, how to set up a fresh firewall.

Chapter 3, Building Strong Policies, helps you find your way around all the policies, objects, and network configuration.

Chapter 4, Taking Control of Sessions, is a deeper dive into understanding how sessions are processed by the firewall.

Chapter 5, Services and Operational Modes, covers fine-tuning the firewall as a device in your network.

Chapter 6, Identifying Users and Controlling Access, looks at applying identification as a core technology to security.

Chapter 7, Managing Firewalls Through Panorama, explores centralized management through the Panorama platform.

Chapter 8, Managing Firewalls Through Strata Cloud Manager, explores centralized management through the Strata Cloud Manager platform.

Chapter 9, Upgrading Firewalls and Panorama, covers planning and preparing for software upgrades.

Chapter 10, Logging and Reporting, explains how to keep track of what's happening on your network through logs and reports.

Chapter 11, Virtual Private Networks (VPNs), is a deep dive into VPN tunnels.

Chapter 12, Advanced Protection, helps you get a firm grasp on security profiles and layer 7 packet inspection.

Chapter 13, Troubleshooting Common Session Issues, covers basic troubleshooting for common issues.

Chapter 14, A Deep Dive Into Troubleshooting, is an advanced troubleshooting walk-through for more complex troubleshooting.

Chapter 15, Cloud-Based Firewall Deployment, explores how to deploy cloud-based firewalls.

The *Appendix* contains bonus material, looking at some standalone advanced features.

To get the most out of this book

Before you dive in, it helps to have a basic grasp of core networking concepts—things like IP addressing, subnets, routing, and how NAT works. You don't need to be an expert, but familiarity with these fundamentals will make the material easier to follow.

No prior experience with Palo Alto Networks products is assumed. We'll walk through everything step by step, from initial setup to advanced features.

Software/hardware covered in the book	Operating system requirements
PAN-OS 11.2 and below	Windows, macOS, or Linux
All Palo Alto software and hardware firewalls, Panorama and Strata Cloud Manager central management platforms, Strata Logging Service (previously Cortex Data Lake)	Windows, macOS, or Linux

Table P.1: Software/hardware and operating system requirements

Download the example code files

The code bundle for the book is hosted on GitHub at `https://github.com/PacktPublishing/Mastering-Palo-Alto-Networks-Third-Edition`.

We also have other code bundles from our rich catalog of books and videos available at `https://github.com/PacktPublishing`. Check them out!

Conventions used

There are a number of text conventions used throughout this book.

`CodeInText`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and social media handles. For example: "On macOS and Linux, a USB serial connection will usually create a new `tty` entry in the `/dev/` directory."

A block of code is set as follows:

```
<uid-message>
<version>1.0</version>
<type>update</type>
<payload>
<login>
<entry user="domain\user" ip="x.x.x.x" timeout="60">
</entry>
</login>
</payload>
</uid-message>
```

Any command-line input or output is written as follows:

```
> request high-availability state suspend
> request high-availability state functional
```

Bold: Indicates a new term, an important word, or words that you see on the screen. For instance, words in menus or dialog boxes appear in the text like this. For example: "When you open the **Network | Interfaces** menu, you will see an assortment of physical interfaces., and routing verdicts are made."

Warnings or important notes appear like this.

Tips and tricks appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have any questions or feedback about this book, please email us at `customercare@packt.com` and mention the book's title in the subject line.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you reported this to us. Please visit `http://www.packtpub.com/submit-errata`, click **Submit Errata**, and fill in the form. We ensure that all valid errata are promptly updated in the GitHub repository at `https://github.com/PacktPublishing/Mastering-Palo-Alto-Networks-Third-Edition`.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `http://authors.packtpub.com/`.

Share your thoughts

Once you've read *Mastering Palo Alto Networks*, *Third Edition*, we'd love to hear your thoughts! Please `click here to go straight to the Amazon review page` for this book and share your feedback.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere?

Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily.

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below:

   `https://packt.link/free-ebook/9781836644811`

2. Submit your proof of purchase.
3. That's it! We'll send your free PDF and other benefits to your email directly.

1

Understanding the Core Technologies

Welcome to the first chapter! In this book, we're going to explore the ins and outs of the Palo Alto Networks Strata suite. We'll start off by learning about all the different features of the firewall and how to configure them before we move on to more complex features and additional services that will help you complete your deployment. On the way, you'll pick up important knowledge nuggets that will help you both understand the technology and pass the PCNSE exam.

In this chapter, we're going to examine the core technologies that make up the Palo Alto Networks firewall. We are going to take a closer look at the way in which security zones control how security, **Network Address Translation** (**NAT**), and routing verdicts are made. We will review the mechanics behind App-ID and Content-ID so you get a deeper understanding of how packets are processed and security decisions are made by the firewall, and we will review how User-ID contributes to a more robust security stance by applying group-based or user-based access control.

This chapter will cover the following topics:

- Understanding the zone-based firewall
- Understanding App-ID and Content-ID
- The management and data planes
- Authenticating users with User-ID

By the end of this chapter, you will have a better understanding of how the core technology is built up and will be able to apply these skills when we start building configuration. If you're preparing for the PCNSE exam, this chapter will also help you understand the fundamentals required to tackle some of the scenario-based questions.

Technical requirements

For this chapter, no physical installation is required. A good understanding of basic networking protocols like UDP and TCP is necessary to fully benefit from the explanations in this chapter. It is helpful if you've already worked with Palo Alto Networks firewalls, but it is not required. Some experience with firewalls or web proxies in general is recommended, as this will make the subject matter more tangible.

Understanding the zone-based firewall

Traditionally, when considering a firewall as an element of your network, most likely you will imagine a network design like the one in *Figure 1.1*, with two to four areas surrounding a box, which represents the firewall. Most of the time, whatever is placed in the north is considered dangerous as it represents the internet; the east and west are somewhat gray areas as they are the **demilitarized zones** (**DMZs**) that are partly exposed to the internet, and the south is the happy place where users do their daily tasks. All these areas will be defined as zones in the firewall:

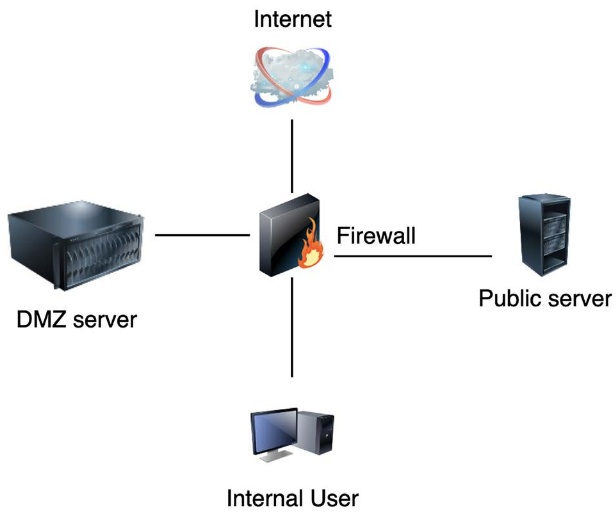


Figure 1.1: Basic network topology

In reality, a network design may look a lot more complex due to network segmentation, remote offices being connected to headquarters via all sorts of different technologies, and the adoption of cloud vendors.

In a **route-based firewall**, zones are simply an architectural or topological concept that helps identify which areas comprise the global network that is used by the company; they are usually represented by tags that can be attached to a subnet object. They have no bearing on any of the security decisions made by the system when processing security policies.

The **zone-based firewall**, on the other hand, will use zones as a means to internally classify the source and destination in its state table.

The following diagram illustrates the phases of packet processing from the first step when the first packet of a new session enters the firewall to the last step where the packet egresses the firewall:

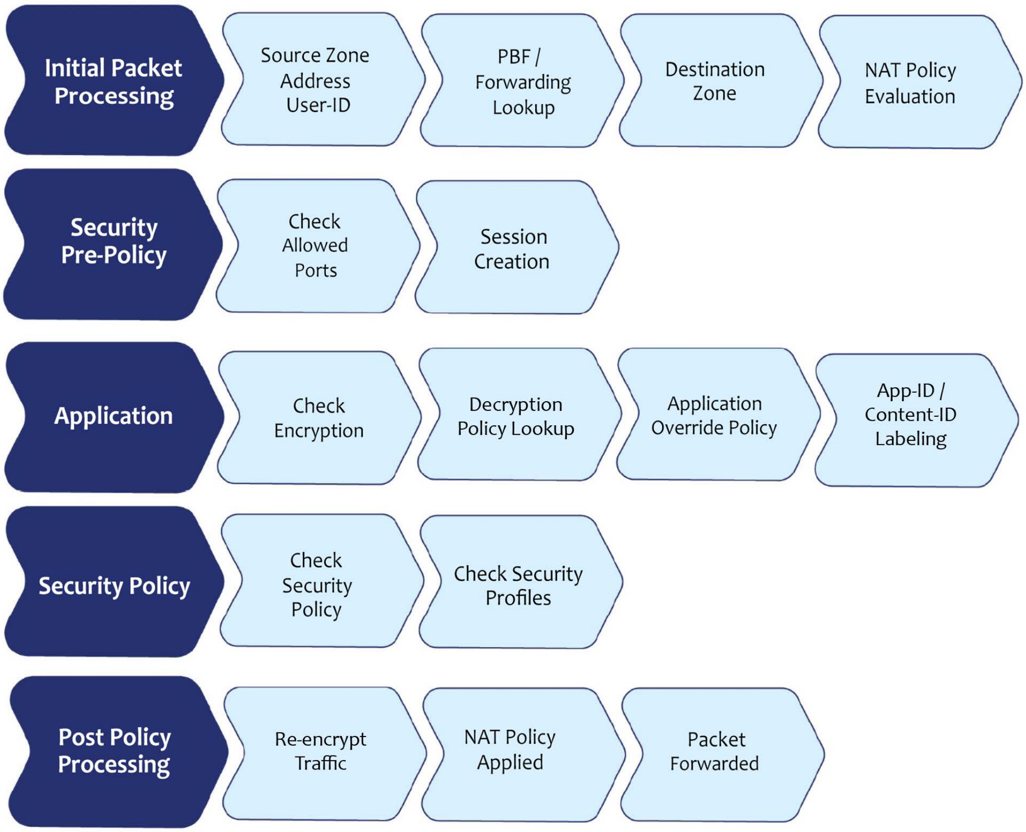


Figure 1.2: Phases of packet processing

Let's look at the process workflow for initial packet processing:

1. When a packet is first received, a source zone lookup is performed. If the source zone has a protection profile associated with it, the packet is evaluated against the profile configuration. If the first packet is a TCP packet, it will also be evaluated against the TCP state where the first packet needs to be a SYN packet, and a SYN cookie is triggered if the protection profile threshold is reached.
2. Then, a destination zone is determined by checking the **policy-based forwarding** (**PBF**) rules and, if no results are found, the routing table is consulted.
3. Lastly, the NAT policy is evaluated as the destination IP may be changed by a NAT rule action, thereby changing the destination interface and zone in the routing table. This would require a secondary forwarding lookup to determine the post-NAT egress interface and zone.

After these zone lookups have been performed in the **initial packet processing**, the firewall will continue to the **security pre-policy** evaluation.

In the pre-policy evaluation, the "six-tuple" (**6-tuple**) is used to match an incoming session against the rule base before establishing or dropping/denying a session. At this stage, the firewall does not consider the application just yet, as this can usually not be determined by the first packet in a session. The six-tuple consists of the following elements and is used in both uni-directional flows of a session:

- Source-address
- Destination-address
- Source-port
- Destination-port
- Protocol
- Security-zone

Zones are attached to a physical, virtual, or sub-interface. Each interface can only be part of one single zone. Zones can be created to suit any naming convention and can be very descriptive in their purpose (**untrust, DMZ, LAN**, and so on), which ensures that, from an administrative standpoint, each area is easily identifiable.

It is best practice to use zones in all security rules, and leveraging a clear naming convention prevents misconfiguration and makes security rules very readable. Networks that are physically separated for whatever reason but are supposed to be connected topologically (for example, users spread over two buildings that come into the firewall on two separate interfaces) can be combined into the same zone, which simplifies policies.

It is important to note that there are implied rules that influence intrazone or interzone sessions. These rules can be found at the bottom of the security policy:

- **Default intrazone connections**: Packets flowing from and to the same zone will be implicitly allowed
- **Default interzone connections**: Packets flowing from one zone to a different zone are implicitly blocked

Security rules can be set to only accept traffic within the same zone, between different zones only, or both. This setting can be changed in the rule **Type** and is set to **Universal** by default. As illustrated in *Figure 1.3*, the **Universal** rule allows sessions to flow from all zones in the **Source** field to all zones in the **Destination** field, from **LAN** to **LAN** and **DMZ**, and from **DMZ** to **LAN** and **DMZ.**

Rules set to the **intrazone** type only allow sessions to flow inside the same zone regardless of whether multiple zones are added to the security rule: from **DMZ** to **DMZ** and from **LAN** to **LAN**, but not from **LAN** to **DMZ** or from **DMZ** to **LAN.**

Rules set to the **interzone** type only allow sessions to flow between different zones: from **DMZ** to **LAN** and from **LAN** to **DMZ**, but not from **DMZ** to **DMZ** or from **LAN** to **LAN**, even though both are listed in the source and destination.

This means that you can perfectly control between which interfaces traffic is allowed to flow to even if you are unable to define subnets in the source or destination, which, for traditional firewalls, means sessions will be allowed to flow everywhere.

			Source		Destination						
	NAME	TYPE	ZONE	ADDRESS	ZONE	ADDRESS	APPLICATION	SERVICE	ACTION	PROFILE	OPTIONS
1	intrazone	intrazone	DMZ LAN	any	(intrazone)	any	allowed web apps	application-default	Allow		
2	interzone	interzone	DMZ LAN	any	DMZ LAN	any	allowed web apps	application-default	Allow		
3	universal	universal	DMZ LAN	any	DMZ LAN	any	allowed web apps	application-default	Allow		
4	intrazone-default	intrazone	any	any	(intrazone)	any	any	any	Allow	none	none
5	interzone-default	interzone	any	any	any	any	any	any	Deny	none	none

Figure 1.3: Different security rule types and default rules

Now that we've seen the important role zones play while making security decisions, let's look at the expected behavior when determining zones.

Expected behavior when determining zones

When a packet arrives on an interface, the PBF policy or routing table will be consulted to determine the destination zone based on the original IP address in the packet header.

Let's consider the following routing table:

```
> show routing route
flags: A:active, ?:loose, C:connect, H:host, S:static, ~:internal, R:rip,
O:ospf, B:bgp,
      Oi:ospf intra-area, Oo:ospf inter-area, O1:ospf ext-type-1, O2:ospf ext-
type-2, E:ecmp, M:multicast
VIRTUAL ROUTER: default (id 1)
  ==========
destination       nexthop       metric flags  interface
0.0.0.0/0         198.51.100.1  10A S    ethernet1/1
198.51.100.0/24   198.51.100.2  0      A C    ethernet1/1
198.51.100.2/32   0.0.0.0       0      A H
192.168.0.0/24    192.168.0.1   0      A C    ethernet1/2
192.168.0.1/32    0.0.0.0       0      A H
172.16.0.0/24     172.16.0.1    0      A C    ethernet1/3
172.16.0.1/32     0.0.0.0       0      A H
total routes shown: 7
```

Let's assume `ethernet1/1` is the external interface with IP address `198.51.100.2` set to zone **external**, `ethernet1/2` is the DMZ interface with IP address `192.168.0.1` set to zone **DMZ**, and `ethernet1/3` is the LAN interface with IP `172.16.0.1` and set to zone **LAN**. The default route is going out of interface `ethernet1/1` to `198.51.100.1` as a next-hop. There are a few scenarios that will influence how the zone is determined:

- **Scenario 1:** A packet is received from client PC `172.16.0.5` with destination IP `1.1.1.1`.

  The firewall quickly determines the source zone is **LAN** and a route lookup determines the destination IP is not a connected network, so the default route needs to be followed to the internet. The destination zone must be **external** because the egress interface is `ethernet1/1`.

- **Scenario 2:** A packet is received from client PC `172.16.0.5` with destination IP `1.1.1.1` but a PBF rule exists that forces all traffic for `1.1.1.1` to the next-hop IP `192.168.0.25`.

  As PBF overrides the routing table, the destination zone will become **DMZ** as the egress interface is now `ethernet1/2`.

- **Scenario 3:** A packet is received from internet IP `203.0.113.1` with destination IP `198.51.100.2`. This is a typical example of what **NAT** looks like to the firewall: it receives a packet with its external IP address as the destination.

  From the perspective of the NAT policy, the source zone will be **external** as the IP is not from a connected network and no static route exists, and the destination zone will *also* be **external** as the IP is connected to that interface.

From a security aspect, however, once NAT is applied, the destination zone will change to the zone that the post-NAT destination IP is connected to (usually **DMZ**).

Important note

Remember that NAT policy evaluation happens after the initial zones have been determined but before the security policy is evaluated. This will cause outbound NAT rules to come from **LAN** and go to **external**, but inbound NAT rules to match as coming from **external** and also going to **external**, while the inbound security rule will use the appropriate destination zone. See *Figure 1.2*.

In this section, we saw how the first round of security decisions relies heavily on zones, which should also reflect any rule base you create going forward: use zones in the source and destination as much as possible to fully control the flow of traffic and prevent unexpected behavior. In the next section, we'll look at what happens in the second round, which also makes a **next-generation firewall** (**NGFW**) "next generation."

Understanding App-ID and Content-ID

App-ID and Content-ID are two technologies that go hand in hand and make up the core inspection mechanism. App-ID relies on decoders to identify and classify flows based on the protocol and layer 7 application. This allows more granular control over what is being allowed or blocked while ensuring an application behaves as expected. Content-ID relies on threat prevention engines to do deep inspection flows for threats, classify URL categories, and prevent data exfiltration.

How App-ID gives more control

Determining which application is contained within a specific data flow is the cornerstone of any NGFW. It can no longer be assumed that any sessions using TCP ports 80 and 443 are simply plaintext or encrypted web browsing. Today's applications predominantly use these ports as their base transport, and many malware developers have leveraged this convergence to well-known ports in an attempt to masquerade their malware as legitimate web traffic while exfiltrating sensitive information or downloading more malicious payloads into an infected host.

The following image illustrates the steps taken by App-ID to identify applications within flows:

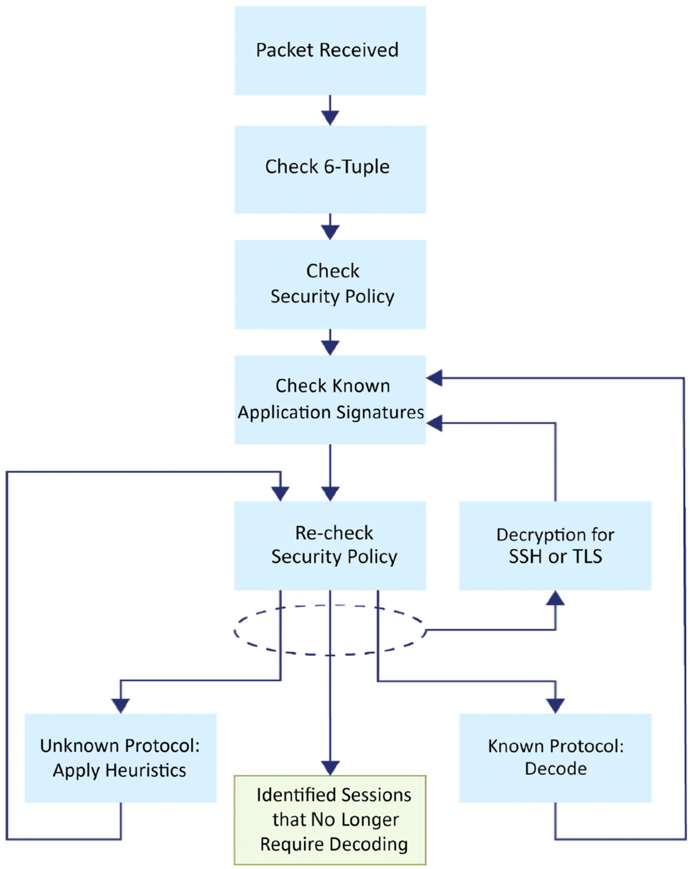


Figure 1.4: How App-ID classifies applications

When a packet is received, App-ID will go through several stages to identify just what something is:

1. First, the **6-Tuple** is checked against the security policy to verify whether a certain source, destination, protocol, and port combination is allowed. This will take care of low-hanging fruit if all the unnecessary ports have been closed off and *unusual* destination ports can already be rejected.
2. Next, the packets will be checked against known application signatures and the app cache to see if the session can be rapidly identified.
3. This is followed by a second security policy check against the application, now adding App-ID to the required set of identifiers for the security policy to allow the session through.
4. If, at this time or in future policy checks, it is determined that the application is SSH, TLS, or SSL, a secondary policy check is performed to verify whether decryption needs to be applied. If a decryption policy exists, the session will go through decryption and will then be checked again for a known application signature, as the session encapsulated inside TLS or SSH may be something entirely different.
5. If, in this step, the application has not been identified (a maximum of 4 packets after the handshake, or 2,000 bytes), App-ID will use the base protocol to determine which decoder to use to analyze the packets more deeply.
6. If the protocol is known, the decoder will go ahead and decode the protocol, then run the payload against the known application signatures again. The outcome could either be a known application (like `ssl` or `web-browsing`) or an unknown generic application, like `unknown-tcp`. This generic application can also be controlled via a security policy to allow or block the session.
7. The session is then re-matched against the security policy to determine whether it is allowed to pass or needs to be rejected or dropped. If the protocol is unknown, App-ID will apply heuristics to try and determine which protocol is used in the session.
8. Once it is determined which protocol is used, another security policy check is performed.
9. Once the application has been identified or all options have been exhausted, App-ID will stop processing the packets for identification.

Throughout the life of a session, the identified application may change several times as more information is learned from the session through inspecting packet after packet. For example, a TCP session may be identified as SSL, which is the HTTPS application as the firewall detects an SSL handshake. The decryption engine and protocol decoders will then be initiated to decrypt the session and identify what is contained inside the encrypted session. Next, it may detect the `web-browsing` application as the decoder identifies typical browsing behavior such as an HTTP GET. App-ID can then apply known application signatures to identify `flickr`. Each time the application context changes, the firewall will quickly check whether this particular application is allowed in its security rule base.

If, at this point, `flickr` is allowed, the same session may later switch contexts again as the user tries to upload a photo, which will trigger another security policy check. The session that was previously allowed may now get blocked by the firewall as the sub-application `flickr-uploading` may not be allowed.

Once the App-ID process has settled on an application, the application decoder will continuously scan the session for expected and deviant behavior, in case the application changes to a sub-application or a malicious actor is trying to tunnel a different application or protocol over the existing session.

App-ID signatures and decoders are regularly (usually once a month around the 15th) updated to account for changes to existing applications or protocols and adding new signatures for previously unknown applications or sub-applications to existing apps to add more depth and control (for example, Facebook chat, file sharing, or games).

App-ID, therefore, allows you to control not only which sessions are allowed to pass through the firewall but also how these applications are allowed to behave. In the next section, we will look at how threats can be prevented and malware blocked.

How Content-ID makes things safe

If the appropriate security profiles have been enabled in the security rules, the Content-ID engine will apply the URL filtering policy and will continuously, and in parallel, scan the session for threats like vulnerability exploits, virus or worm infections, suspicious DNS queries, **command and control** (**C&C** or **C2**) signatures, DoS attacks, port scans, malformed protocols, or data patterns matching sensitive data exfiltration. TCP reassembly and IP defragmentation are performed to prevent packet-level evasion techniques. In the following image, you can see how single-pass pattern matching enables simultaneous scanning for multiple types of threats and how URL filtering is added to the mix:

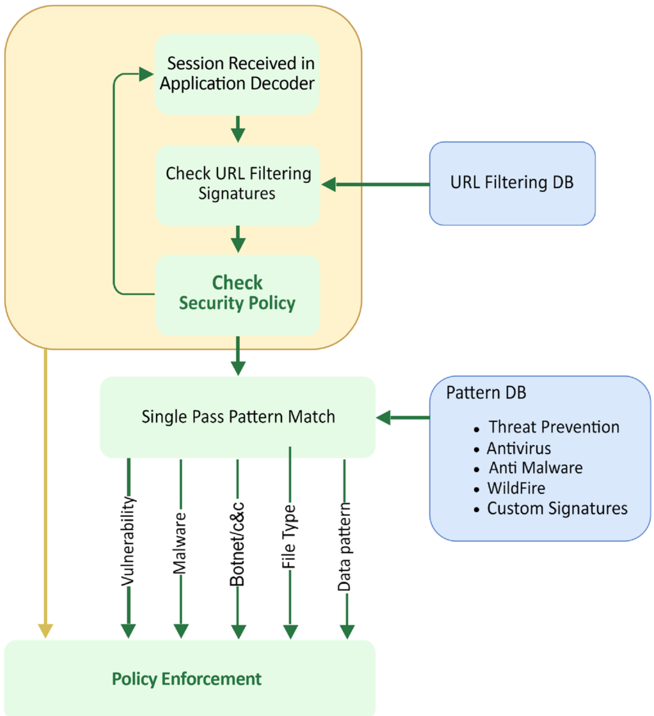


Figure 1.5: How Content-ID scans packets

All of this happens in parallel because the hardware and software were designed so that each packet is simultaneously processed by an App-ID decoder and a Content-ID stream-based engine, each in a dedicated chip on the chassis or through a dedicated process in a **virtual machine** (**VM**). This design reduces latency versus serial processing, which means that enabling more security profiles does not come at an exponential cost to performance as is the case with other firewall and IPS solutions.

Inline evaluation

To extend the signature-based evaluation that we just discussed, the firewall also comes with **Inline Machine Learning** (**ML**) and **Inline Cloud Analysis** capabilities that allow it to evaluate dynamic content independently without needing to wait for a signature update to start blocking malicious content. Some of these are as follows:

- **WildFire** is capable of evaluating **portable executable** (**PE**), **executable and linked format** (**ELF**), MS Office files, PowerShell, and shell scripts in real time by applying ML models.
- Starting from PAN-OS 11.2, certain chassis (**PA-5400** and **PA-VM** at the time of writing) get local deep learning AI that expands the ML capabilities.
- **URL Filtering** can leverage local and cloud Inline ML categorization to evaluate website details to protect users from phishing variants and JavaScript exploits.
- **Anti-Spyware** can tap into live Inline Cloud Analysis to access five analysis engines for C&C-based threats over HTTP, HTTP2, SSL, unknown-UDP, and unknown-TCP.
- **Vulnerability Protection** can leverage Inline Cloud Analysis to analyze SQL and PowerShell code injection.

All cloud inline detection requires the firewall to have an active internet connection, so plan accordingly if internet access is limited for your deployment.

In this section, you learned how all the (OSI) layer 7 content inspection components work together to provide you with more visibility into which applications are traversing the firewall while blocking any malicious payload.

The management and data planes

Hardware and VM design are focused on enabling the best performance for parallel processing while still performing tasks that cost processing power and could impede the speed at which flows are able to pass through the system. For this reason, each platform is split up into so-called *planes*.

There are two main planes that make up a firewall, the **data plane** and the **management plane**, which are physical or logical boards that perform specific functions. While all platforms have a management plane, larger platforms like the PA-5200 have an additional **control plane** and two to three data planes. The largest platforms have replaceable hardware blades (line cards) that have up to three data plane equivalents per line card and can hold up to 10 line cards. Smaller platforms like the PA-220 only have one hardware board that virtually splits up responsibilities among its CPU cores.

The **management plane** is where all administrative tasks happen. It serves the web interfaces used by the system to allow configuration, provide URL filtering block pages, and serve the client VPN portal. It performs cloud lookups for URL filtering and DNS security, and downloads and installs content updates onto the data plane. It also performs the logic part of routing and communicates with dynamic routing peers and neighbors. Authentication, User-ID, logging, and many other supporting functions are not directly related to processing packets.

The **control plane** takes on the task of facilitating communications between multiple data planes and the management plane as well as monitoring processes on the data planes.

The **data plane** is responsible for processing flows and performing all the security features associated with the NGFW. It scans sessions for patterns and heuristics. It maintains IPsec VPN connections and has hardware offloading to provide wire-speed throughputs. Due to its architecture and the use of interconnected specialty chips, all types of scanning can happen in parallel as each chip processes packets simultaneously and reports its findings.

A switch fabric enables communication between planes so the data plane can send lookup requests to the management plane, and the management plane can send configuration updates and content updates.

The following diagram illustrates how these components interact with each other:

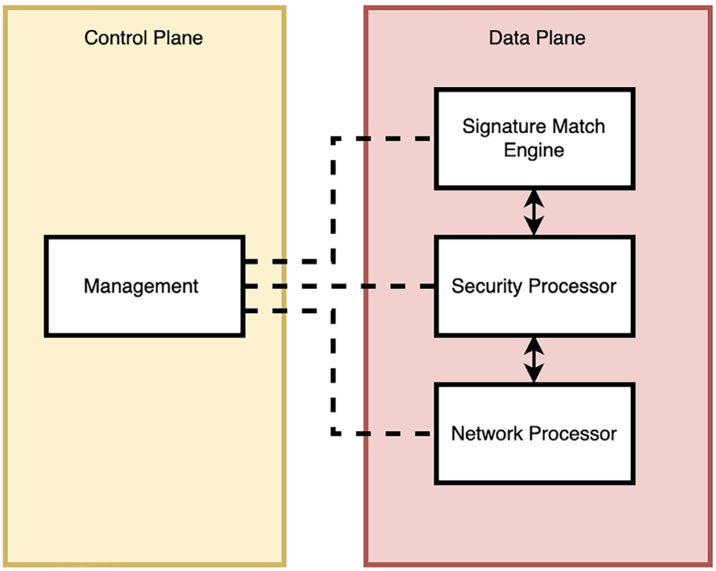


Figure 1.6: Management and data planes

Now that we've covered the most basic functions and you have a firm grasp of how the hardware is organized, let's look at identity-based authorization. The ability to identify users and apply different security policies based on identity or group membership is an important feature of the NGFW as it allows more dynamic security rules that don't rely on static access lists but, instead, allow users to roam inside and outside the campus and still have all the access they need without exposing internal resources.

Authenticating and authorizing users with User-ID

Frequently neglected but very powerful when set up properly is a standard (no additional license required) feature called User-ID. Through several mechanisms, the firewall can learn who is initiating which sessions, regardless of their device, operating system, or source IP. Additionally, security policies can be set so users are granted access or restricted in their capabilities based on their individual ID or group membership.

User-ID expands functionality with granular control of who is accessing certain resources and provides customizable reporting capabilities for forensic or managerial reporting.

Users can be identified through several different methods:

- **Server monitoring**:
    - Microsoft Active Directory security log reading for log-on and authentication events
    - Microsoft Exchange Server log-on events
    - Novell eDirectory log-on events
- The interception of **X-Forwarded-For** (**XFF**) headers, forwarded by a downstream proxy server
- **Client probing** using NetBIOS and WMI probes
- **Direct user authentication**:
    - The Captive Portal to intercept web requests and serve a user authentication form or transparently authenticate using Kerberos
    - GlobalProtect VPN client integration
- **Port mapping** on a multiuser platform such as Citrix or Microsoft Terminal Server where multiple users will originate from the same source IP
- The **Extensible Markup Language** (**XML**) API
- A **syslog listener** to receive forwarded logs from external authentication systems

You will have noticed there are many ways to leverage User-ID, so we will revisit this topic in depth in *Chapter 6*, *Identifying Users and Controlling Access*.

Summary

Now that you've completed this chapter, you are able to identify the strengths of using a zone-based firewall versus a route-based one. You understand how applications can be identified even though they may all be using the same protocol and port, and you understand how deep packet inspection is achieved in single-pass parallel processing. Most importantly, you have a firm grasp of which phases a packet goes through to form a session. It's okay if this information seems a bit overwhelming; we will see more practical applications, and implications, in the next two chapters. We will be taking a closer look at how security and NAT rules behave once you start playing with zones, and how to anticipate expected behavior by simply glancing at the rules.

If you are preparing for the PCNSE exam, this chapter covered parts of the *Planning and Core Concepts* and *Deploy and Configure* domains. Make note of *Figure 1.2* regarding packet processing, remember that route lookups and PBF form the basis of zoning, and take note of how App-ID and Content-ID interoperate.

In the next chapter, we will learn how to set up a firewall from scratch and get up and running in no time. We will glance over the physical and virtual components and how to configure them so traffic can flow through, and NAT can be applied where needed.

2

Setting up a New Device

In this chapter, we will cover how you can gain access to the console and web interface of a fresh-out-of-the-box firewall appliance or a cleanly staged **virtual machine** (**VM**). You will learn how to license, update, and upgrade the firewall so that the latest features are available when you start building your security policy, and the latest signatures are always loaded onto the device to protect your users and infrastructure from malware and vulnerability exploits.

We are going to harden your management configuration to ensure a rigid security stance, and we will also look at the different types of network interface modes—aggregated interfaces and routing.

In this chapter, we're going to cover the following main topics:

- Gaining access to the user interface
- Adding licenses and setting up dynamic updates
- Upgrading the firewall
- Hardening the management interface
- Understanding the interface types

By the end of this chapter, you'll be able to quickly set up a fresh firewall, register it, and upgrade it to a desirable level in a short amount of time. You'll be able to apply best practices and leverage strong authentication for your administrative access, and you will be able to quickly identify which interface configuration will suit any given network topology that the firewall needs to be placed in.

Technical requirements

For this chapter, a basic understanding of network appliances is required as we will be looking at physically connecting to a device, configuring the management environment, and choosing the data plane interface's deployment mode. Basic knowledge of standing up a virtual appliance in a virtual environment, including connecting it to virtual switches or virtual interfaces and providing it with network access on a hypervisor, is also required.

Gaining access to the user interface

> If you are deploying your firewall on a cloud provider like Azure or AWS, take a look at *Chapter 15*, *Cloud-Based Firewall Deployment*.

When taking a new device out of the box or setting up a VM on a local hypervisor, such as VMware ESXi, Fusion, NSX, Hyper-V, KVM, and so on, one of the first things that needs to be accomplished is making it possible to gain access to the user interface, be it a web interface or a console connection. We'll go over the most common ways of gaining this access in this section.

One of the first things you *may* need to do is to connect a console cable to gain access to the **command-line interface** (**CLI**). Older models only come with an RJ45 console port, so for those, you will need a standard DB9-to-RJ45 console cable, optionally patched through a serial-to-USB cable so a modern laptop is able to interface with the port. The pinout for the DB9 should be as follows:

```
1 - Empty - Data Carrier Detect (DCD)
2 - 3 - Receive Data (RXD)
3 - 6 - Transmit Data - (TXD)
4 - 7 - Data Terminal Ready (DTR)
5 - 4 - Ground (GND)
6 - 2 - Data Set Ready (DSR)
7 - 8 - Request To Send (RTS)
8 - 1 - Clear to Send - (CTS)
9 - Empty - Ringing Indicator (RI)
```

Luckily, there are USB-to-RJ45 cables available as well that will save you the trouble of figuring out the correct pinouts:

Figure 2.1: RJ45-to-USB console cable

All but the very old models also come with a micro-USB port, which allows a console connection to be made using a standard USB-A-to-micro-USB cable, as in the following picture:

Figure 2.2: PA-460 RJ45 and the micro-USB console ports

In all cases, you will need to find which COM (hardware interface) or **TTY** (**TeleTYpewriter**) port is being used on your computer's operating system.

On a Windows machine, the first time you plug in the cable, a driver may need to be installed. Once the installation has been completed, you need to find the virtual COM port number that has been assigned to the console cable. In most cases, you can determine this virtual COM port number by following these steps:

1. Open the **Device Manager**.
2. Click **Start | Control Panel | Hardware and Sound | Device Manager** (under **Devices and Printers**).
3. In the **Device Manager** list, look in **Ports** and find the virtual COM port assigned to the USB port. This entry will look similar to **USB to Serial Port (COM#)**, where COM# is the number to be used in the following step.

Next, you will need a terminal emulation client to connect to the console. You can use a free client for this, such as PuTTY from `https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html`.

Besides the COM port, you may need to provide more settings to be able to connect. If asked, use these settings:

```
Bits per second: 9600
Data Bits: 8
Parity: none
Stop bits: 1
Flow control: none
```

On macOS and Linux, a USB serial connection will usually create a new `tty` entry in the `/dev/` directory; a USB-to-DB9 dongle may create a **call-up** (**CU**) entry in the `/dev/` directory.

Find the proper device by searching with either of these commands:

```
ls /dev/tty.*
ls /dev/cu.*
```

You will find `/dev/cu.usbserialxxxxx` or `/dev/tty.usbmodemxxxxx`, where `xxxxx` is the serial device name.

Once you determine the appropriate device, you can connect to the console port by using the `screen` command set to `9600` bits per second:

```
screen /dev/tty.usbmodemxxxxx 9600
```

Now, go ahead and connect the console cable or micro-USB to your laptop and appliance. If you have a port free on your management network, go ahead and connect the firewall's MGT port to the switch. If you don't have a management connection available yet, you will need to connect your laptop directly to the MGT port for easier access once the IP is set up on the management interface. Lastly, plug in the power cable.

If the firewall is loaded in a VM or cloud entity, hit the **Start** button to boot up the virtual appliance.

Once you've logged on to the console, you will see the operating system boot up, and if the firewall is already connected to a DHCP-enabled management network, you will see something similar to the following, where the DHCP address is already listed for your convenience:

Figure 2.3: PA-VM post-boot DHCP information

If you missed this information, you can log on and use the following command to see the DHCP information:

```
admin@PA-440> show system info
hostname: PA-440
ip-address: 192.168.27.116
public-ip-address: unknown
netmask: 255.255.255.0
default-gateway: 192.168.27.1
ip-assignment: dhcp
```

If, for some reason, you have not received a DHCP address yet from your DHCP server, you can initiate a `renew` action from the CLI by using a `> request dhcp client management-interface renew` command.

Important note

The default username and password for a factory settings appliance or VM are as follows:

`Username: admin`

`Password: admin`

The first time you log on, you will be asked to change this default password. Remember to commit the configuration to save your new password.

Accessing the management interface

Now that you have access to the user interface, you will need to make an important decision right away: whether you will be using **zero-touch provisioning** (**ZTP**) or not (we'll see a bit more about that later in *Chapter 7*, *Managing Firewalls Through Panorama*). In short, ZTP enables a device to automatically receive coordinates for a central management Panorama system, if you have one, and immediately download its configuration from there. By default, this option is enabled on all new devices and is bound to interface `ethernet1/1`. If you do not need this functionality, you must turn it off before you can start using this interface. You can do so by issuing the command `set system ztp disable`, as follows:

```
admin@PA-440> set system ztp disable
Executing this command will disable Zero Touch Provisioning (ZTP), and remove
all logs and configuration. The system will restart in standard mode for
regular configuration of the firewall. Are you sure you want to continue?  (y
or n)
```

If you hit the *Y* key, the system will reboot and ZTP will be disabled. If you change your mind later and want to enable it again, you can do so by issuing the following command: `set system ztp enable`.

Once your system has rebooted, continue with the initial configuration.

If your network does not have a DHCP server, or you connected the firewall directly to your laptop, you will need to set an IP address manually. Copy and paste the following into a text file and alter the `<IP>` entries with the appropriate IP for your management interface, the default gateway it will use to reach out to the internet, and the DNS servers it will use to resolve the domain names. Type the netmask in quad decimals, not in CIDR (slash notation subnet, such as `/16` and `/24`):

```
configure
set deviceconfig system type static
set deviceconfig system ip-address <IP>
set deviceconfig system netmask <x.x.x.x>
set deviceconfig system default-gateway <IP>
set deviceconfig system dns-setting servers primary <IP>
set deviceconfig system dns-setting servers secondary <IP>
commit
```

You can chain `set` commands that belong in the same path and class so that you do not need to set each attribute in individual `set` commands; instead, you can add all the desired settings at once.

In the next example, we go into configuration mode, switch the management interface from DHCP to static configuration, and then combine all the configuration parameters for the management interface into one `set` command. If you haven't done so already, start by changing the default password to a new one, and then add the interface configuration:

```
admin@PA-440> set password
Enter old password :
Enter new password :
```

```
Confirm password :
Password changed
admin@PA-440> configure
Entering configuration mode
[edit]
admin@PA-440# set deviceconfig system type static
[edit]
admin@PA-440# set deviceconfig system ip-address 192.168.27.5 netmask
255.255.255.0 default-gateway 192.168.27.1 dns-setting servers primary 1.1.1.1
secondary 1.0.0.1
[edit]
admin@PA-440# commit
Commit job 2 is in progress. Use Ctrl+C to return to command prompt
...........................................55%....75%.....98%.................
.....100%
Configuration committed successfully
[edit]
admin@PA-440#
```

You may need to log back in after running the commit job when the admin password is changed.

Important note

The > prompt in username@hostname> indicates that you are in *operational* mode and can execute runtime commands. The # prompt in username@hostname# indicates that you are in *configuration* mode and can add configuration parameters.

Operational commands can be run from configuration mode by prefixing run to a command—for example, user@host# run show clock.

Once the commit job finishes, you will be able to connect to the web interface through https://<IP> or by using an SSH client, such as PuTTY or the ssh command in Linux or macOS.

You are now able to get onto a freshly started firewall and configure it, so we can move on to the next step and gain access to the web interface.

Connecting to the web interface and CLI

Now that your device has an IP address, you can connect to its web interface via any browser using https://<IP>.

You will be met with an unfriendly error message, as in the following screenshots. This is due to the web interface using a self-signed certificate that has not been validated by any authority. For now, this can be safely ignored:

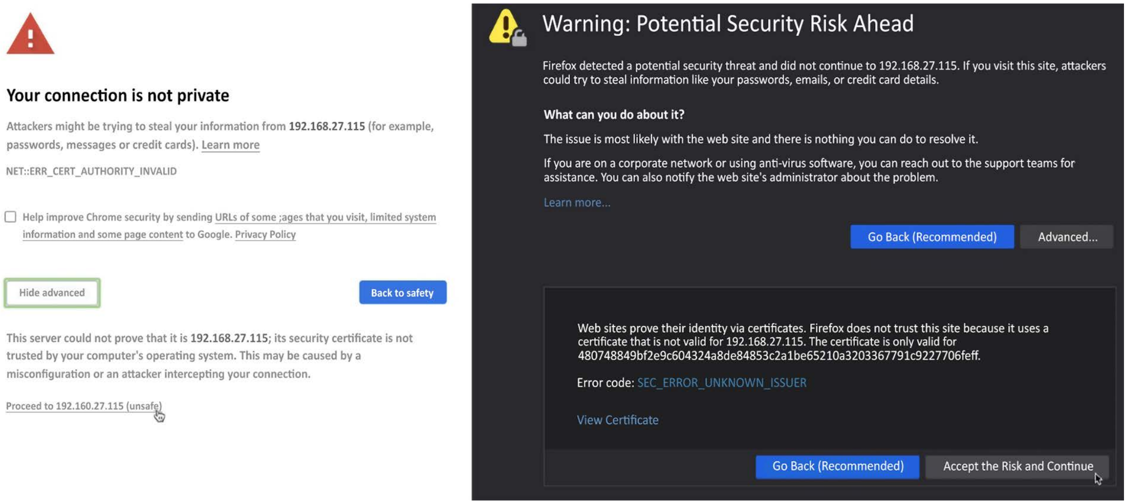


Figure 2.4: Certificate warnings in Chrome and Firefox

An SSH client will provide you with a slightly friendlier question:

```
tom$ ssh -l admin 192.168.27.115
The authenticity of host '192.168.27.115 (192.168.27.115)' can't be
established.
RSA key fingerprint is SHA256:Qmre8VyePwwGlaDmm6JTYtjou42d1i/Ru6xZmmEk8Yc.
Are you sure you want to continue connecting (yes/no)?
```

The SSH connection will provide you with mostly the same user experience as the console connection, but SSH is more responsive and secure, and you can now access your device from anywhere on the management network.

The web interface provides you with a whole new user experience. When prompted for your username and password, input the username `admin` with default password `admin` or the password you created on the terminal, or the username and password you created on the cloud provider.

Once you are logged in, the first screen you will see is the dashboard, which contains some general information about the health of your system, config changes, and which admins are logged on. The dashboard can be customized and additional widgets can be added from a list of prepared widgets, or widgets can be removed if they are not relevant.

For now, the **General Information** widget contains the most important information as you will need the **serial number** of the physical device, or the **CPU ID** and **UUID** on a virtual device, as shown in the following screenshot. The CPU ID and UUID will be needed to register and activate the VM while a physical device can be activated by its serial number:

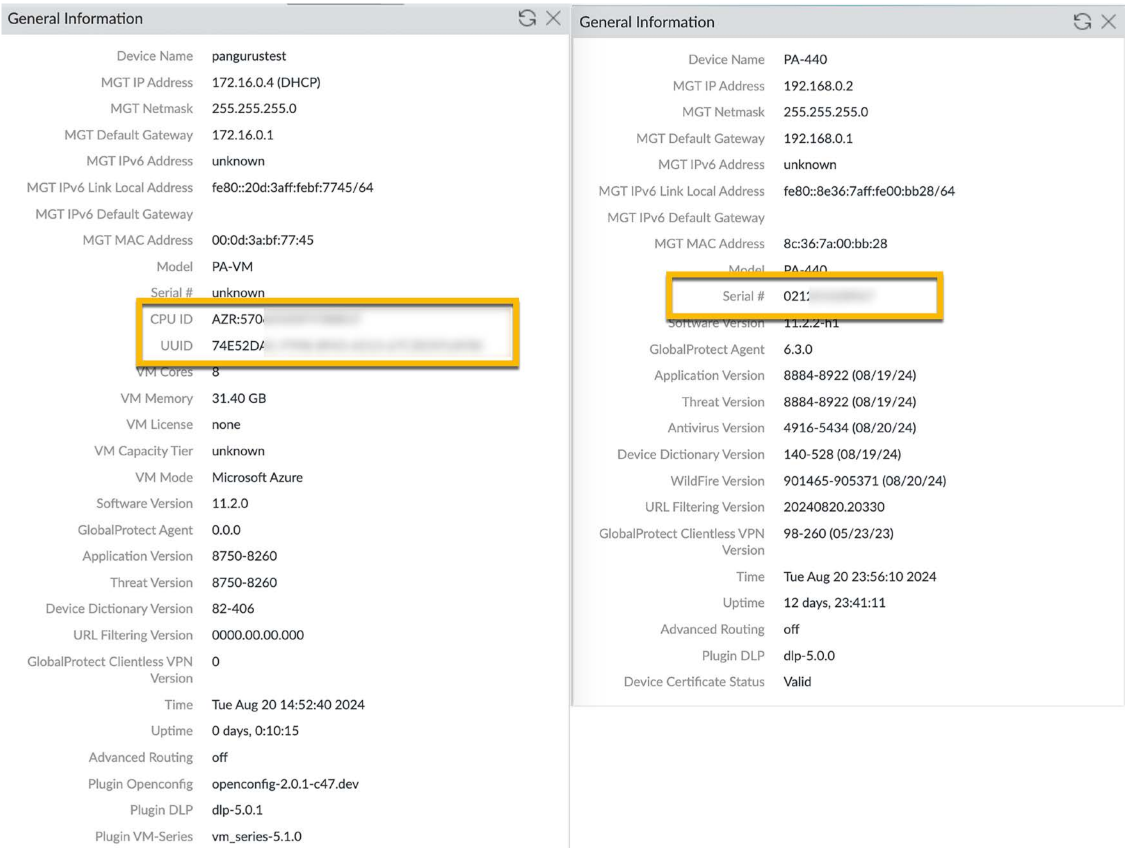


Figure 2.5: On the left is a PA-VM device, and on the right is a PA-440 device

Now that we have access to the web interface and are able to collect the system's base information, we can go ahead and register the firewall and activate any of the feature licenses that were purchased. We will now have a look at how to perform the registration and licensing procedures.

Adding licenses and setting up dynamic updates

Before we can start adding licenses, the device needs to be registered. You will need to note down the device's serial number or, if you do not have a support portal account, the sales order number to create a new account. The sales order number will typically be sent to you by your distributor or reseller after you've placed your order.

Open a new tab or browser and navigate to `https://support.paloaltonetworks.com`. If you do not have an account yet, you will first need to create a new one so that you can gain access to the portal from where you will be able to manage all your devices, activate your licenses, download software packages and updates, and access support cases. If you already have a **Customer Support Portal** (**CSP**) account, you can skip to *Registering a new device*.

Creating a new account

When creating a new account, you will be asked for an email address and whether you want to register using a serial number or an **authorization** (**auth**) code, as in the following screenshot. The serial number is needed when registering a hardware appliance; the auth code is used when registering a VM device:

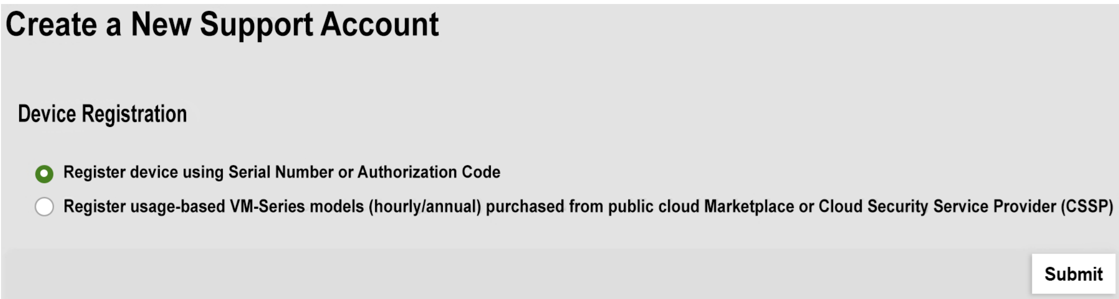


Figure 2.6: Serial or authorization code device registration

Alternatively, if you have set up a virtual appliance on one of the cloud providers, you can pick which provider your device is running on (such as Amazon Web Services, Azure, Google Cloud Platform, and so on).

You then need to provide some basic details, such as the address, the password, the device's serial number, the auth code, and the sales order number or customer ID, if your company already has an account:

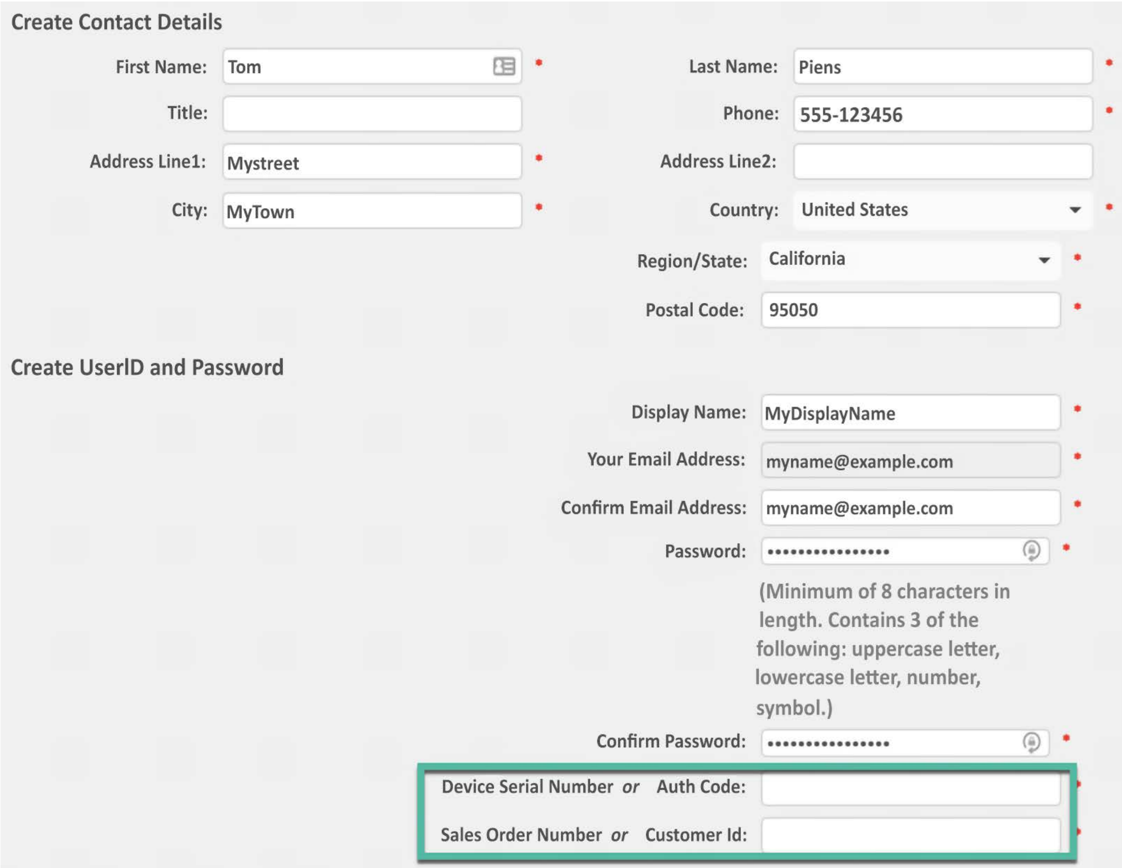


Figure 2.7: General information and device and sales order details

This account creation step will already register your first device; you can go ahead and register more devices in the following section.

Registering a new device

Ensure you are logged in to your account on the support portal and click on **Register a Device** from the home page:

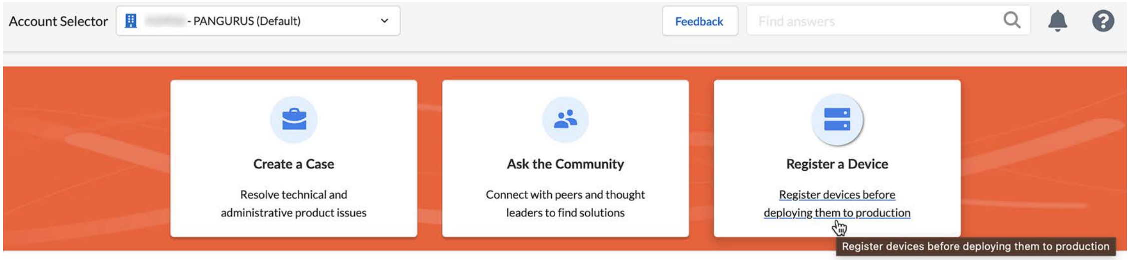


Figure 2.8: Register a Device from the support portal home page

You will be presented with the option to register using a serial number or an auth code. The serial number is needed when registering a hardware appliance and the auth code is used when registering a VM device:

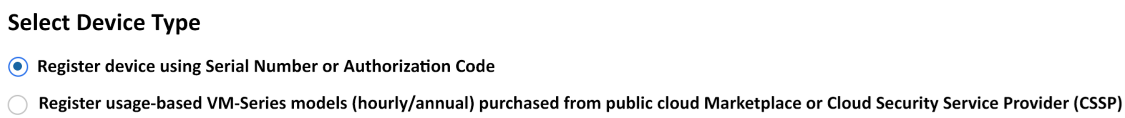


Figure 2.9: Serial or auth code device registration

Register device using Serial Number or Authorization Code will ask you for the serial number, a friendly device name, and a tag if you have several "pools" or groups of devices in your account already. It will also request address details as to where the device will be deployed for **return merchandise authorization** (RMA) purposes.

If you are deploying a cloud instance, you can choose to register usage-based VM-Series models. You'll be asked for the serial number, CPUID, and UUID:

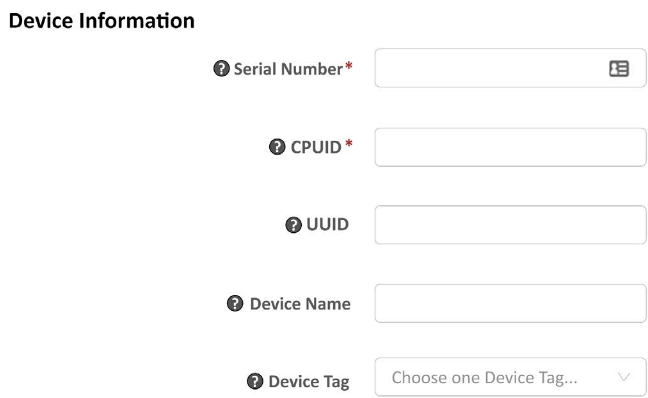


Figure 2.10: Adding a cloud instance to the assets

Now that the devices are registered, it is time to activate the feature and support licenses.

Activating licenses

Once the device is registered, you can add the licenses. You will have received one (a bundle) or several auth codes that you need to enter on the portal or via the **Device** | **Licenses** tab to activate the license and start using the feature on your device. There are multiple ways to activate licenses, which we'll cover in this section, but let's take a look at the different types of licenses first.

Some of the most common licenses include the following:

- **Support**: **Platinum 4h** (**PLAT**), **Premium 24/7** (**PREM**), **Standard 9/5** (**STD**), **Partner-enabled 4h support** (**B4HR**), or **Regular Partner-enabled support** (**BND**). Support licenses grant you access to a support organization, allow you to download software and app ID updates, and entitle you to replacement hardware if your firewall breaks.
- **Threat Prevention** (**TP**): Antivirus, anti-spyware, threat prevention, daily updates.
- **Advanced Threat Prevention** (**A-TP**): Adds inline cloud analysis for SQL and command injection.
- **PAN-DB URL filtering** (**URL4**) is the basic cloud-based URL category lookup license. This license will be phased out in favor of ADVURL.
- **Advanced URL filtering** (**ADVURL**): The **Machine Learning** (**ML**)-enabled URL filtering license that adds an automated mechanism to analyze unknown URLs in real time.
- **GlobalProtect** (**GP**): Enables mobile applications on Android, iOS, Win10 UWP, Chrome OS, and Linux. It enables **Host Information Profile** (**HIP**) checks and agentless VPNs. It also allows split tunneling based on host application or domain names.
- **DNS security** (**DNS**): Dynamic DNS signature lookups and blocking or "sinkholing" of malicious DNS requests.
- **Advanced DNS security** (**A-DNS**): Adds inline cloud analysis for command-and-control lookups.
- **WildFire** (**WF**): Live threat signature feed, real-time ML-enabled analysis, and cloud-based sandbox analysis.
- **Advanced WildFire** (**A-WF**): Live threat signature feed, real-time ML-enabled analysis, and cloud-based sandbox analysis.
- **Decryption port mirroring**: Allows decrypted sessions to be copied to a different device for additional IDS scanning via a dedicated "port mirror" interface.
- **Internet of Things** (**IoT**): Enables detection of IoT devices and generates rulebase adjustments to protect vulnerable IoT devices.
- **Data loss prevention** (**DLP**): ML-powered data loss prevention scanning.

More features are being added as Palo Alto Networks announces new products.

Activating licenses via the customer support portal

In the **CSP**, you can find your registered devices under the **Assets** tab as a device. There's a pencil icon that allows you to activate auth codes:

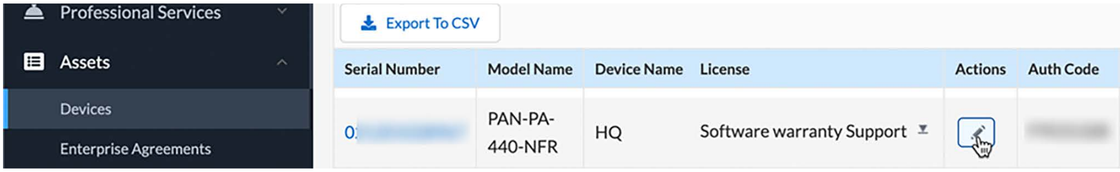


Figure 2.11: The Devices page in the CSP

You will notice there is already a software warranty support license active for a limited amount of time. This is a temporary support license that allows an RMA to be started if your device arrives broken in the box. To add the actual support license and any feature licenses, click on the pencil icon in the **Actions** column you see in *Figure 2.11*. This will call the pop-over feature activation window:

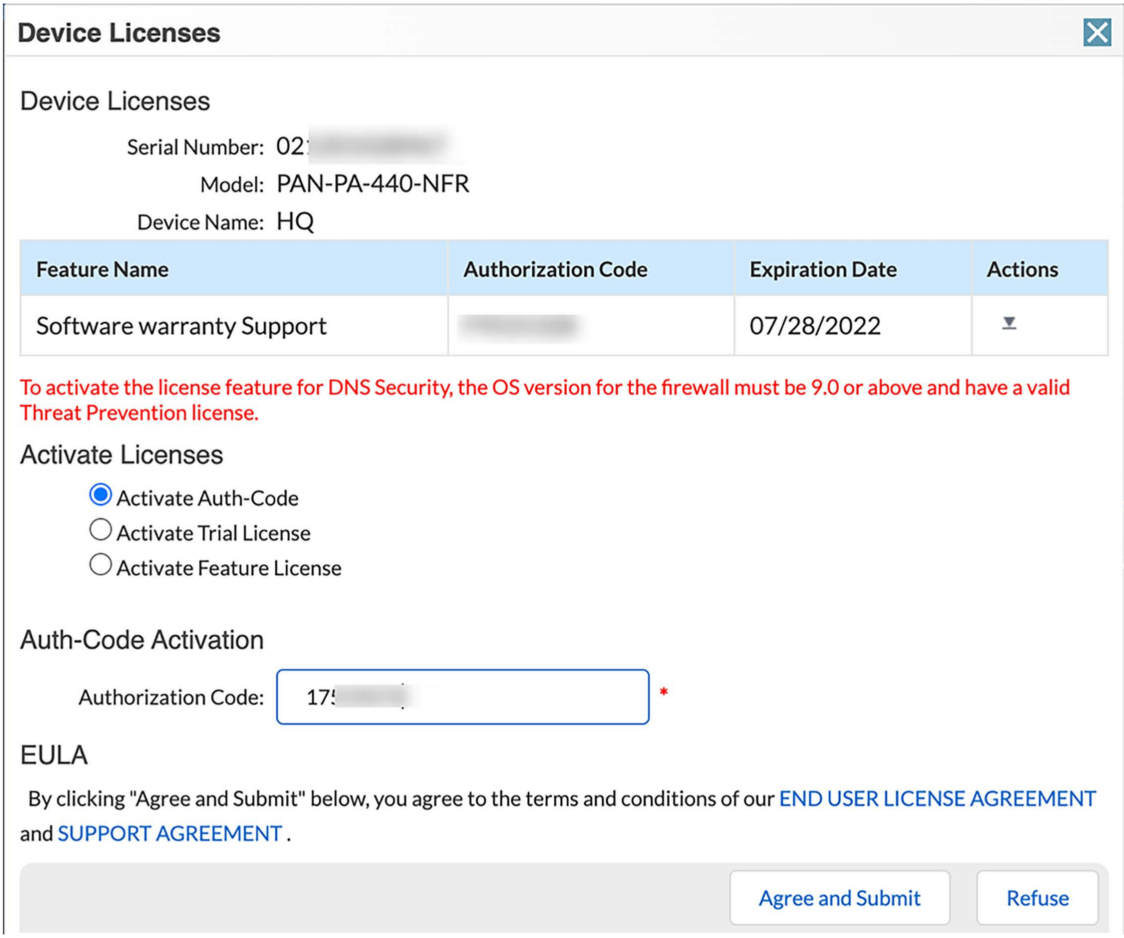
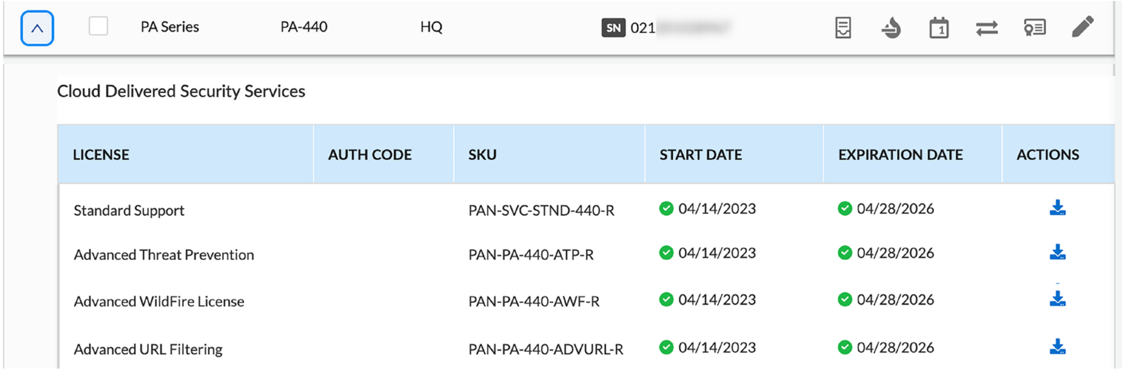


Figure 2.12: Adding auth codes to activate services

Once you've added all your licenses, the device should look something like this:



PA Series	PA-440	HQ	SN 021	

Cloud Delivered Security Services

LICENSE	AUTH CODE	SKU	START DATE	EXPIRATION DATE	ACTIONS
Standard Support		PAN-SVC-STND-440-R	04/14/2023	04/28/2026	
Advanced Threat Prevention		PAN-PA-440-ATP-R	04/14/2023	04/28/2026	
Advanced WildFire License		PAN-PA-440-AWF-R	04/14/2023	04/28/2026	
Advanced URL Filtering		PAN-PA-440-ADVURL-R	04/14/2023	04/28/2026	

Figure 2.13: A fully licensed device

The little download icons next to each license allow you to download the license key file so that you can upload the key onto the firewall. This is required if you intend to run the firewall without an internet connection and want to be able to upload signature files and enforce security profiles.

Besides activating licenses via the support portal, they can also be activated directly from the firewall interface.

Activating licenses via the web interface

> This procedure requires that the management interface has an internet connection and is able to resolve internet domain names via DNS. If an internet connection is not available, see the previous section on how to download the license keys.

To activate licenses via the web interface, navigate to **Device** | **Licenses**. If you activated the licenses in the CSP and then proceeded to download the license key files, you can click on **Manually upload license key**.

If you activated the licenses on the CSP and want to fetch the licenses, click **Retrieve license keys from license server**. Make sure the firewall has been set up with a functional default gateway and DNS servers.

If you want to activate new licenses with an auth code, click on **Activate feature using authorization code** and you will see a popup where you can enter each auth code individually:

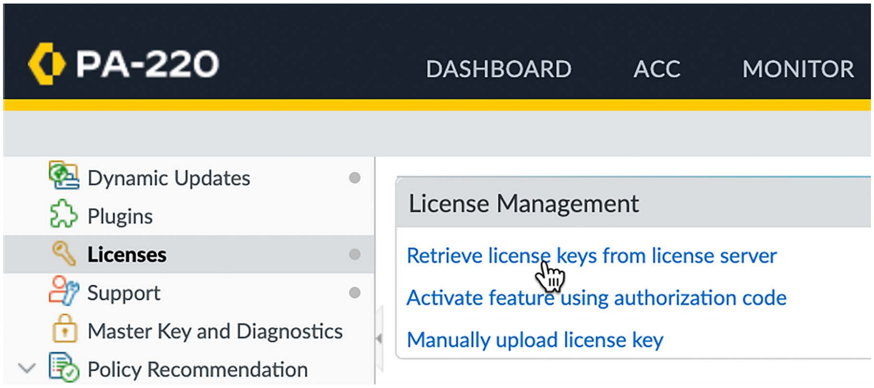


Figure 2.14: Activating a license using an auth code

With each added license, a section will be added containing the license information:

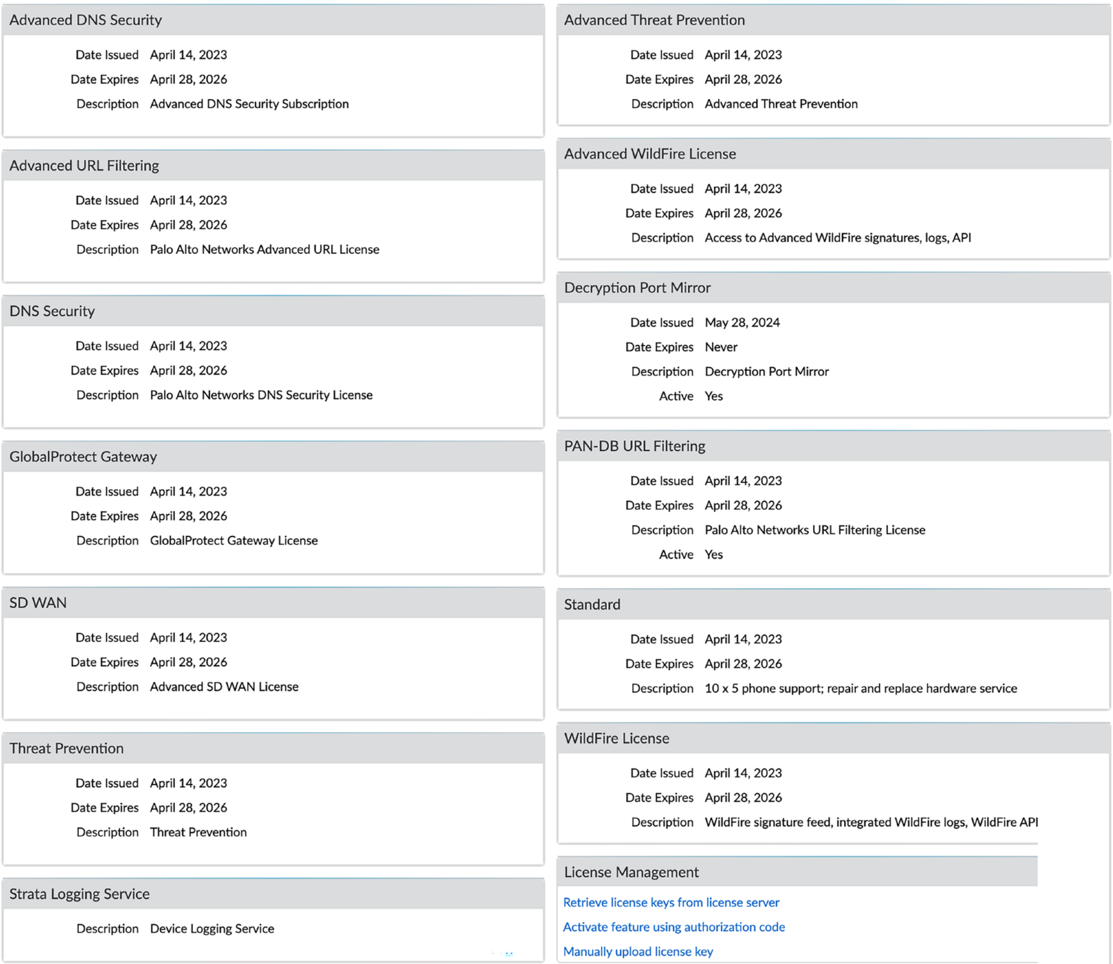


Figure 2.15: Active licenses on the device

To activate the support license, you may need to activate the auth key through the **Support** menu item:

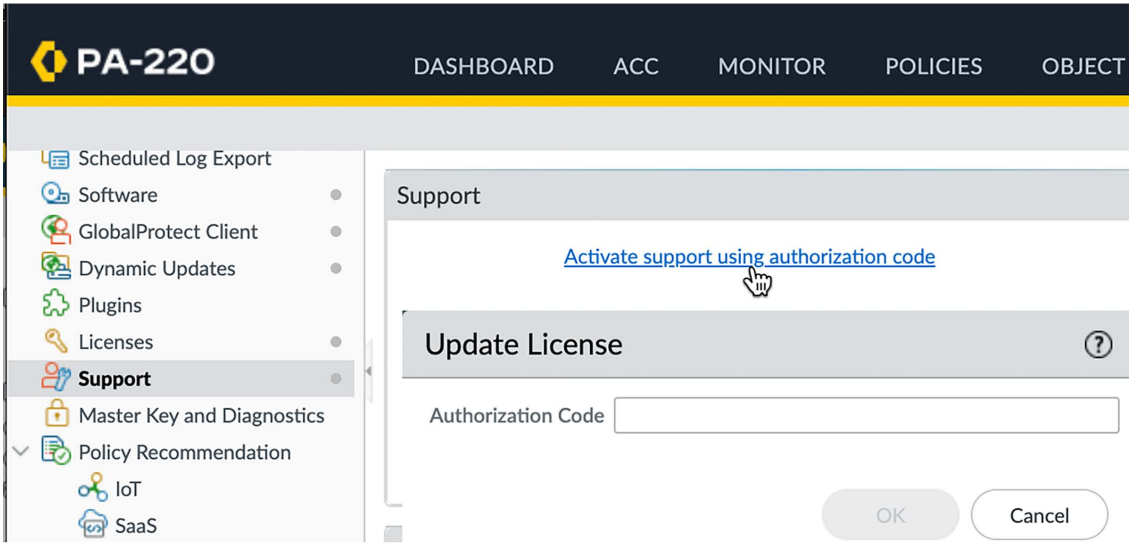


Figure 2.16: Activate support using an authorization code

Important note

On hardware firewalls the support license is more like a contract than a license required for a feature to work; a support person will take your call if something goes wrong, a replacement device will be sent if your unit is broken, and so on. This is the only license that does not need to be on the device necessarily before features start to function (i.e., all base functionalities will work without the support license present).

On VM firewalls, the support license is tied to the VM's activation and the number of CPU cores it supports, so on the VM series, you are required to load the support license.

After all licenses are activated on the device, the next step is to start downloading and scheduling updates to the different databases.

Downloading and scheduling dynamic updates

Now that all the licenses are active, you can set up dynamic updates and start downloading all the content packages.

Navigate to the **Dynamic Updates** menu under the **Device** tab, where you can manually download content packages and set up schedules and installation preferences. The first time you visit this menu, it may look a bit off as the available content has not been loaded onto the device yet. Click the **Check Now** button to connect to the updates server and fetch the available packages for your system, as shown:

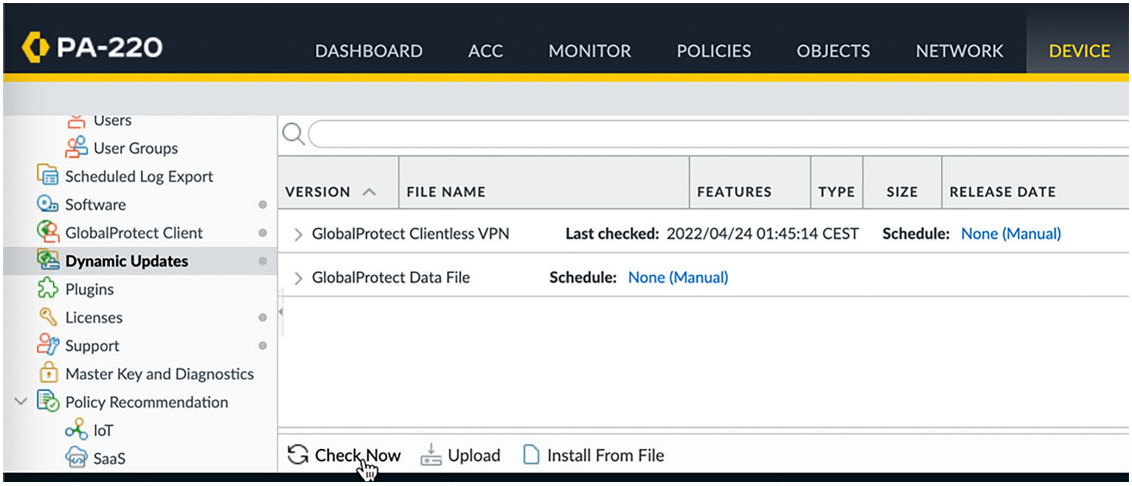


Figure 2.17: The initial Dynamic Updates view

Once the updates have been fetched, you may still notice that some antivirus packages are missing. This is because the device first needs to be brought up to date with all the app ID and content ID application and decoder updates before further packages can be loaded onto the system. Go ahead and download the latest **Applications and Threats** package:

VERSION	FILE NAME	FEATURES	TYPE	SIZE	RELEASE DATE	DOW...	CURRE... INSTA...	ACTION
Applications and Threats	Last checked: 2022/04/28 03:15:14 CEST			Schedule: Every day at 03:15 (Download and Install)				
8548-7321	panupv2-all-contents-8548-7321	Apps, Threats	Full	53 MB	2022/03/31 06:20:36 CEST			Download
8549-7323	panupv2-all-contents-8549-7323	Apps, Threats	Full	53 MB	2022/04/01 03:46:34 CEST			Download
8550-7325	panupv2-all-contents-8550-7325	Apps, Threats	Full	53 MB	2022/04/05 01:37:16 CEST			Download
8551-7330	panupv2-all-contents-8551-7330	Apps, Threats	Full	53 MB	2022/04/05 23:43:41 CEST			Download

Figure 2.18: Downloading the first Applications and Threats package

Important note

If no threat prevention license has been activated, there will only be an **Applications** package available for download. This will be indicated by the package name starting with `panupv2-all-apps` instead of `panupv2-all-content` and the **FEATURES** column only listing **Apps.**

Once the package has been downloaded, click **Install.** Once the installation is complete, click **Check Now** again, and the antivirus will become available. Go ahead and download and install the latest package of antivirus updates.

Important note

URL filtering and DNS security do not have update packages because URLs are looked up against the cloud service and then stored in the local cache.

You can now start building schedules by clicking on the blue **None (Manual)** option after **Schedule:**

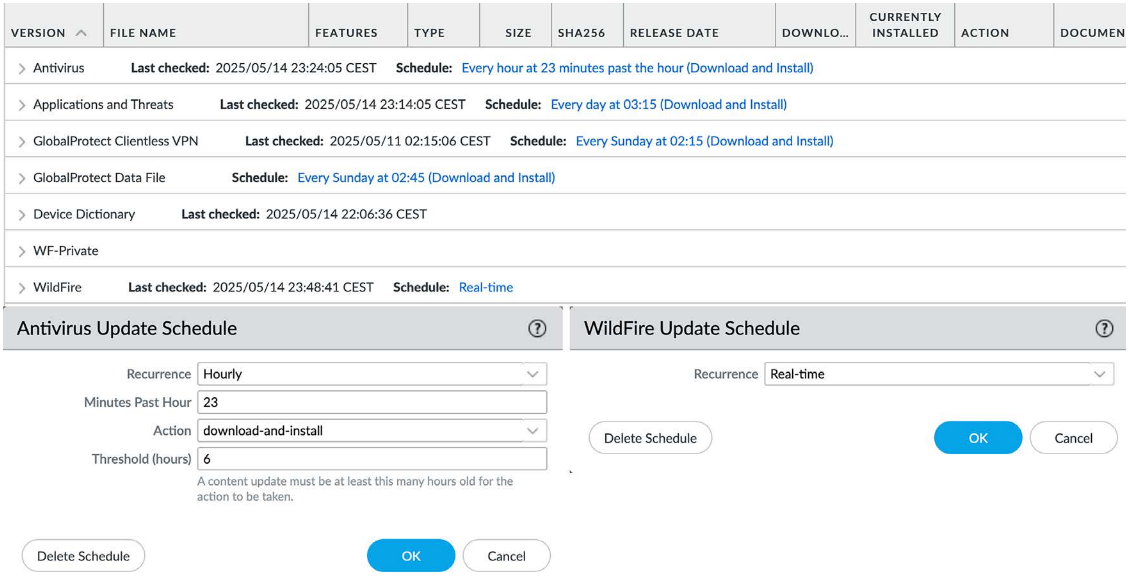


Figure 2.19: The antivirus and WildFire schedules

The antivirus and WildFire schedules look very similar. They both include the **Recurrence** parameter, which tells the firewall how regularly it needs to check for updates. The options differ for both schedules, though:

- The update interval options for **Antivirus** are **Weekly**, **Daily**, **Hourly**, or **Manual**.
- The update interval options for **WildFire** are **Real-time**, **Every minute**, **15 minutes**, **30 minutes**, **1 hour**, or **Never**.

When **Recurrence** is set to any value higher than 1 minute (15 or 30 minutes, or hourly), you can additionally set at which minute within the frame the actual check should take place (e.g., 13 minutes past the hour, 4 minutes into the quarter hour, etc). This helps prevent conflicting update connections to the update server in cases where the outgoing internet bandwidth is restricted.

The **Action** can be set to simply **download** or **download-and-install**. If the action is set to **download**, manual installation is required.

Threshold is a feature that the antivirus update shares with **Applications and Threats:**

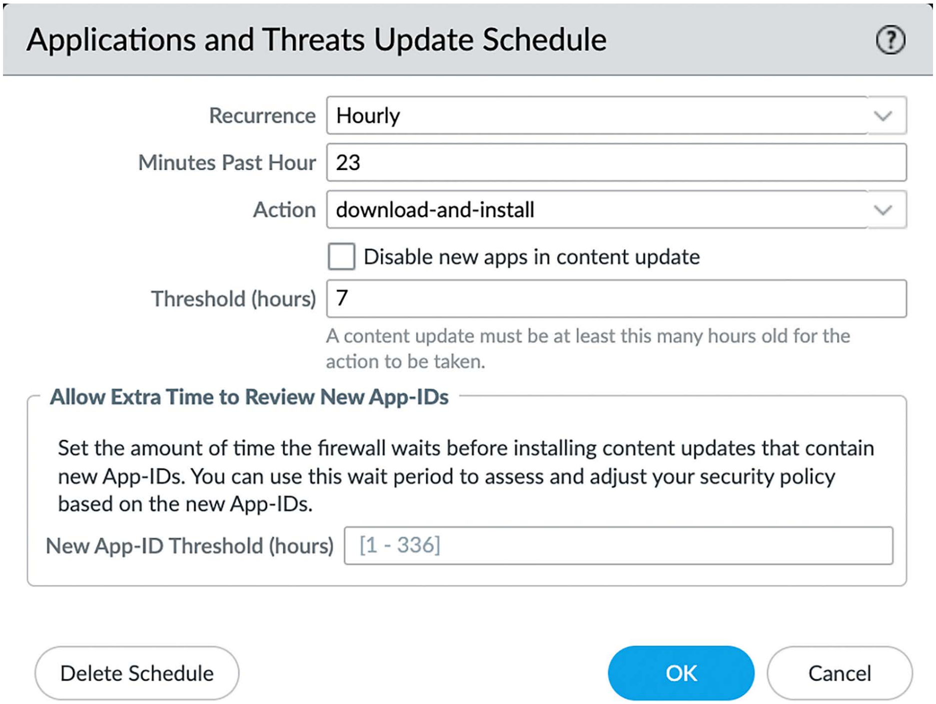


Figure 2.20: Antivirus and WildFire schedules

Threshold is a setting that delays the installation of a content package for a set number of hours. At the time that this threshold expires, the firewall checks for a new update package. If a new package is found, the new package is downloaded and **Threshold** is reset for one more attempt. If yet another update package is found after the first reset, the schedule will reset until the next full occurrence. If no new packages are detected, the package will be installed as defined by **Threshold.**

The threshold delay is a mechanism to prevent installing faulty packages; if the vendor provides poorly crafted content, the delay is set in hours, which should allow other accounts to experience the flaws and report the content issue back to the support teams. So, the content is rolled back via the administrators or the vendor. This thresholding option correlates with a company's tolerance for the risk of vendor errors and the balance of new and emerging threats to the organization.

According to Palo Alto Networks' best practices, a security-first approach is to set the threshold between 6 to 12 hours; however, for a critical environment, the threshold should be 24 hours.

The **Application** content package also has an option to completely **Disable new apps in content update** or enable a separate threshold on the app IDs only **to Allow Extra Time to Review New App-IDs**. The reasoning here is that what is identified as web browsing today may change into a unique application after installing the **Application** content package tomorrow. If the security policy has been set up to only allow previously known applications, this could potentially cause issues with users who suddenly can't access that specific application.

The **New App-ID Threshold** setting allows you to schedule a review period to see whether any applications need to be accounted for in the security policy before they become active. If no action is needed, the applications will become active automatically. The **Disable new apps in content update** option will not activate any new applications until you manually review and activate all new applications.

Important note

At the time of writing, the release schedule for new applications is the third Tuesday of each month. Regular threat package updates happen on Tuesdays, but urgent updates are sent out immediately.

Dynamic updates cheat sheet

Here is a quick set of recommendations for scheduling dynamic updates:

1. Click on **Check Now**.
2. Download and install the latest `panupv2-all-contents` or `panupv2-all-apps` package:
    - `panupv2-all-content` includes all app ID, spyware, and vulnerability updates. This package requires an active Threat Prevention license to be installed successfully.
    - `panupv2-all-apps` only includes app ID updates and is used when a Threat Prevention license is not active on the device.
3. Click **Check Now**, which will make the antivirus packages visible.
4. Download and install the latest `panup-all-antivirus` package.
5. Set an **Antivirus** update schedule:
    - Hourly recurrence
    - 15 minutes after the hour
    - Download and install
    - 6-hour threshold
6. If you have a WildFire subscription license, set a **WildFire** update schedule to **Real-time** (firewalls on high-latency internet links can be set to 1-minute or 15-minute update intervals).

7. Set an **Applications and Threats** update schedule:
    - Every 30 minutes
    - 7 minutes past the half-hour
    - Download and install
    - 6-hour threshold
    - Leave the **Allow Extra Time to Review New App-IDs** field blank if new app IDs can be added immediately; set to a 48-hour threshold (or more) if the security team wishes to review new applications before they are activated

The settings in this section are considered best practices as they ensure dynamic updates are scheduled so as not to interfere with other scheduled tasks, like report generation. The threshold ensures an update is not applied until some time has passed by postponing the actual installation and then rechecking the available content package at the threshold time. If a new package is available due to an error or an urgent update, the new package is downloaded and the threshold timer is refreshed. The process repeats once the threshold has been reached again and either the latest package is installed or the update is postponed to the next scheduled occurrence. Not only do content packages need to be updated frequently, but new software versions are made available at regular times to address bugs or introduce new features. Let's now have a look at the steps needed to upgrade your firewall.

Upgrading the firewall

In this section, you will learn how to upgrade your firewall and what steps need to be taken to ensure a smooth process. We will review important information to keep in mind when preparing your maintenance window and providing a contingency plan. In-depth upgrade procedures are provided in *Chapter 9*, *Upgrading Firewalls and Panorama*.

Understanding the partitions

Before we start the upgrade procedure, there's an important bit of information you need to know. Like most Linux systems, the hard disk has been partitioned into specific segments. These segments serve a specific purpose.

A few important ones are as follows:

- `/` is the root partition, which is where the operating system is installed
- `/opt/pancfg` is where the configuration files and dynamic update files are kept
- `/opt/panrepo` is the repository for downloaded operating system (PAN-OS) images
- `/opt/panlogs` is the partition where `logdatabase` is stored

There is a hidden `maint` partition that can only be accessed during bootup. This partition is used for disaster recovery, factory resetting the device, and disk issues.

The disk space usage can be viewed with the following command:

```
admin@PA-440> show system disk-space
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p2   21G  5.9G   14G  31% /
none            7.8G   64K  7.8G   1% /dev
/dev/mmcblk0p5   32G   13G   18G  42% /opt/pancfg
/dev/mmcblk0p6   18G  9.6G  7.0G  58% /opt/panrepo
tmpfs           7.8G  3.8G  4.1G  48% /dev/shm
cgroup_root     7.8G     0  7.8G   0% /cgroup
/dev/mmcblk0p8   22G   11G  9.6G  53% /opt/panlogs
None
```

The cool thing about the / root partition is that it is actually one of two `sysroot` partitions. The system has been partitioned with two operating system-specific partitions, of which just one is mounted at a time. The upgrade procedure installs the new PAN-OS onto the inactive partition.

This allows inline upgrades without interrupting the active partition. Once the new operating system has been installed, the GRUB bootloader is configured to load the other `sysroot` partition at the next boot, causing the new PAN-OS to become active:

```
admin@PA-440> debug swm status
Partition         State             Version
------------------------------------------------------------------------------
--
sysroot0          RUNNING-ACTIVE    11.2.2-h1
sysroot1          REVERTABLE        11.2.0
maint             READY             11.2.2-h1
```

This mechanism also allows a smooth rollback in case an upgrade fails and it is decided you need to go back to the previous situation. You can trigger the `> debug swm revert` debug command to tell the bootloader to switch the toggle again to the previous `sysroot` partition and reboot the system via `> request restart system`. After the device has rebooted, you are back on the previous PAN-OS with the pre-upgrade configuration loaded.

Upgrade considerations

When upgrading, you will need to map out where you are, where you need to go, and how you need to get there. You can find where you are by looking at the dashboard's **General Information** section and looking for the software version. Deciding where you need to go may require some research and consideration:

- Which features are required? Are there new features in a release you need, or are you running smoothly with the features you have?
- Is the code train "mature"? Is a new major release brand new, and do the new features weigh up against the risk of being an early adopter?

- Are there outstanding advisories that trump the required features? Was a critical vulnerability found that has not been addressed in the version you plan to go to?
- When is an upgrade required and when is it optional? Is there a direct need to upgrade due to a vulnerability or a bug, or are there features in a newer release that will be helpful?
- Will your current OS code train reach its end of life soon? (You can review the end-of-life dates here: `https://www.paloaltonetworks.com/services/support/end-of-life-announcements/end-of-life-summary`).

These questions will help you determine which upgrades need to take place immediately and which ones can be scheduled ahead of time, possibly with a more relaxed testing process before deploying to the production environment or even postponed until a more mature maintenance release version is available. Let's now try to answer some of these questions.

Which features are required?

Determining which features are contained in each PAN-OS version requires the most research. You can open `https://docs.paloaltonetworks.com` and search for `Feature Guide`, which will return all the new feature guides for the major PAN-OS versions.

Is the code train "mature"?

Maturity can be estimated by looking at the maintenance release version. All PAN-OS versions are made up of three numbers – PAN-OS X.Y.Z (for example, 10.1.5):

- X is the number of the major software release
- Y is the number of the feature version release
- Z is the number of the maintenance release

X will change when a new major software version is released containing new functionality and usually containing some changes in its expected behavior and possibly a new look and feel.

Each new software release is usually followed by a new feature version around 6 to 9 months after its release, mostly containing some new features. Maintenance release versions are released for all code trains anywhere between 5 and 9 weeks and mostly contain bug fixes.

There will occasionally be PAN-OS version names that end in `-hx`, which denotes a hotfix. This is a maintenance release that was published ahead of schedule and usually only contains one or a few critical hotfixes (for example, `11.2.2-h1`).

A code train will usually reach a reliable maturity around the `x.y.4` or `x.y.5` maintenance release version when it is somewhat safe to assume most critical bugs have been found and addressed.

Check the release notes for any known issues so that you can appropriately prepare if there are any caveats: `https://docs.paloaltonetworks.com/pan-os/11-2/pan-os-release-notes.html`.

Are there outstanding advisories that trump the required features?

Advisories regarding which maintenance release versions to choose or avoid can be found at `https://securityadvisories.paloaltonetworks.com/` and `https://live.paloaltonetworks.com/t5/Customer-Resources/`.

When is an upgrade required and when is it optional?

Each major version has a base image, usually the `x.y.0` version, which contains all the vital parts of the PAN-OS image.

This allows the following maintenance versions to be smaller in size, containing only critical updates. The base image needs to be downloaded onto the system before a maintenance version can be installed. It is not required for the base image to be installed in order to be able to install the maintenance version when upgrading from a lower major version. It is also not required to install any intermediate maintenance versions unless the release notes explicitly mention that there is an issue that requires a step in between.

Say, for example, that your firewall is currently on PAN-OS 11.0.4 and you need to get to PAN-OS 11.1.5. You can download a PAN-OS 11.1.0 base image, followed by a PAN-OS 11.1.5 maintenance image, and then directly install and reboot PAN-OS 11.1.5. Your system will be directly upgraded from 11.0.4 to 11.1.5.

If, for example, your firewall is currently on PAN-OS 10.0.10 and you want to go to PAN-OS 11.1.5, you do need to download a PAN-OS 10.1.0 base image and download, install, and reboot the latest maintenance release (e.g., 10.1.14) before you can install PAN-OS 11.1.5.

Important note

In PAN-OS 10.1, a new feature was introduced that allows you to skip up to three PAN-OS versions when upgrading.

When upgrading PAN-OS 9.1 to 10.1, you are required to install and reboot into PAN-OS 10.0. When upgrading PAN-OS 10.1 to 11.1, you can immediately go to the latest 11.1 maintenance version without needing to install 10.2 or 11.0.

You will now have a good understanding of when and why you would need to upgrade and how to decide which version you need to upgrade to. In the following section, we'll briefly go over the upgrade process via different methods; see *Chapter 9*, *Upgrading Firewalls and Panorama*, for a more thorough upgrade process.

Upgrading via the CLI

Using the CLI, commands can be quickly executed to perform tasks. To upgrade via the CLI, you can follow these steps:

1. You first need to retrieve the available software images that can be installed on your system. You won't be able to download any images before the list is retrieved:

```
admin@PA-440> request system software check
Version                          Size    Released on         Downloaded
-------------------------------------------------------------------------
11.2.2-h1                        749MB   2024/08/02 15:58:14  no
11.2.1                           502MB   2024/07/08 12:04:42  no
11.2.0-b3                        1228MB   2024/03/14 12:31:10  no
11.2.0-b1                        1258MB   2024/02/06 14:20:41  no
11.2.0                           1239MB   2024/05/02 10:15:34  no
11.1.4-h1                        1015MB   2024/08/08 02:18:35  yes
11.1.4                           946MB   2024/06/27 08:17:35  no
11.1.3                           893MB   2024/05/14 15:14:14  no
```

2. Next, you can download the desired PAN-OS version. Remember to first download the base image if you're upgrading to a new major release:

```
admin@PA-440> request system software download version 11.2.0
Download job enqueued with jobid 31
```

3. You can track the download status with the following command:

```
admin@PA-440> show jobs id 31
Enqueued            Dequeued ID Type   Status Result Completed
--------------------------------------------------------------
2024/08/22 23:24:15 23:24:15 31 Downld FIN    OK     23:25:31
Warnings:
Details:Successfully downloaded
Preloading into software manager
Successfully loaded into software manager
```

4. Follow up with the maintenance release:

```
admin@PA-440> request system software download version 11.2.2-h1
Download job enqueued with jobid 32
```

5. When the software is successfully downloaded, you can commence installing it onto the system. You will be prompted that a reboot is required to complete the installation and to confirm whether you are sure that you want to continue. Type `Y` to proceed with the installation:

```
admin@PA-440> request system software install version 12.2.2-h1
Executing this command will install a new version of software. It will
not take effect until system is restarted. Do you want to continue? (y or
n)
Software install job enqueued with jobid 32. Run 'show jobs id 32' to
monitor its status. Please reboot the device after the installation is
done.
```

6. You can track the installation progress through the `show jobs` command:

```
admin@PA-440> show jobs id 32
Enqueued            Dequeued ID Type   Status Result Completed
--------------------------------------------------------------
2024/08/22 23:35:28 23:35:28 32 SWInstall FIN OK     23:38:59
Warnings:
Details:Software installation successfully completed. Please reboot to
switch to the new version.
```

7. To complete the installation, reboot the firewall. Type `Y` into the dialog if you are certain that you want to go ahead with the reboot. Rebooting will cause all sessions to be interrupted and no new sessions to be accepted until the firewall has completed the `autocommit` job:

```
admin@PA-440> request restart system
Executing this command will disconnect the current session. Do you want
to continue? (y or n)
```

8. The `autocommit` job runs right after a reboot and serves to load the configuration onto the data plane. After a software upgrade, this process can take a while:

```
admin@PA-440> show jobs all
Enqueued            Dequeued ID Type   Status Result Completed
--------------------------------------------------------------
2024/08/22 23:44:27 23:44:27 1  AutoCom FIN   OK      23:44:36
```

If you prefer to upgrade the firewall via the web interface, follow the procedure outlined in the next section.

Upgrading via the web interface

Software images can be downloaded and installed from the **Device** | **Software** menu. The first time you access this page, you will be presented with an error message because no repository has been loaded yet:

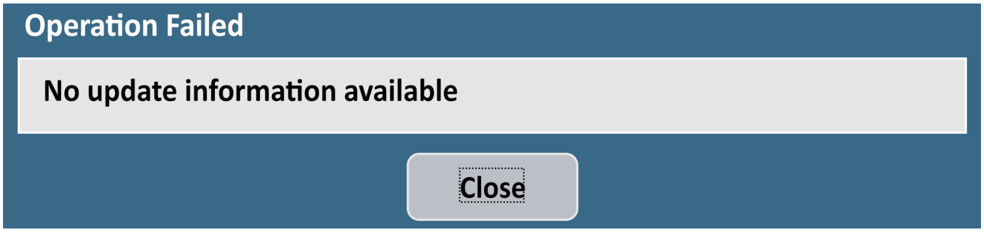


Figure 2.21: Error message on the first visit to the software page

You can ignore this warning; click **Close** and then click **Check Now**. Once the repository has loaded, you will see all the available software images. If you want to install a certain maintenance release version and you want to make sure you meet all the requirements, starting from PAN-OS 10.2, you can click the **Validate** button and the system will check which requirements there may be for this version and whether you have already downloaded them. This will show you the following:

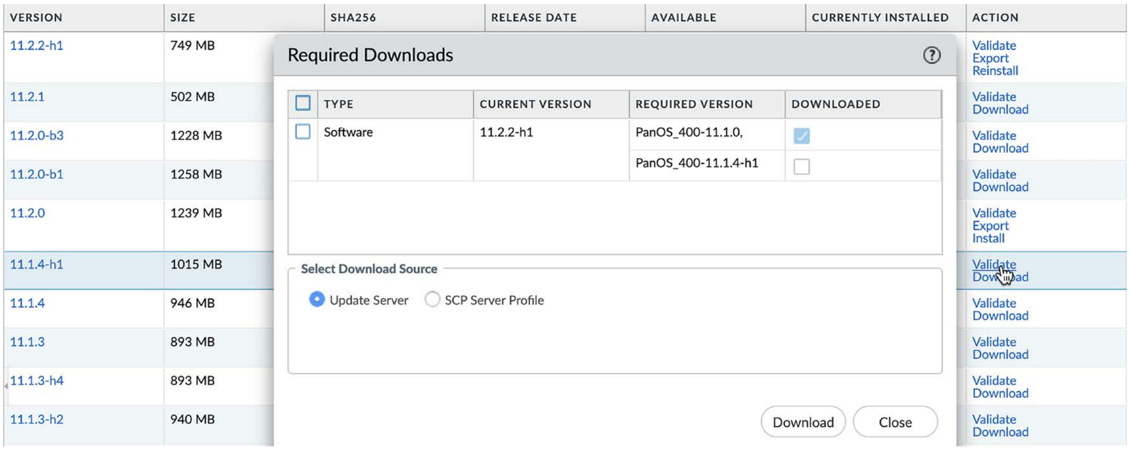


Figure 2.22: Validate action

Click the **Download** link next to the PAN-OS version you want to upgrade to and wait for the download dialog to complete:

VERSION	SIZE	SHA256	RELEASE DATE	AVAILABLE	CURRENTLY INSTALLED	ACTION	DOCUMENTATION	RELEASE TYPE
11.2.2-h1	749 MB	dc6bb08c42def20e5425...	2024/08/02 15:58:14			Validate Download	Release Notes	
11.2.1	502 MB	e01057f9aa1a80cf3075...	2024/07/08 12:04:42			Validate Download	Release Notes	
11.2.0-b3	1228 MB	6486b919373239de66e...	2024/03/14 12:31:10			Validate Download	Release Notes	
11.2.0-b1	1258 MB	5641eb4d7bbc1c647b1...	2024/02/06 14:20:41			Validate Download	Release Notes	
11.2.0	1239 MB	7771ec3da10f5d9a2024...	2024/05/02 10:15:34	Downloaded	✓	Validate Export Reinstall	Release Notes	Base
11.1.4-h1	1015 MB	7a22cc2799d5b8b9aa85...	2024/08/08 02:18:35			Validate Download	Release Notes	
11.1.4	946 MB	1be2daf822958208b56...	2024/06/27 08:17:35			Validate Download	Release Notes	
11.1.3	893 MB	15e1058633db29a434e...	2024/05/14 15:14:14			Validate Download	Release Notes	
11.1.3-h4	893 MB	e32959deb63d688b8da...	2024/08/08 13:01:43			Validate Download	Release Notes	
11.1.3-h2	940 MB	af059832438b5d01f021...	2024/07/18 08:43:21			Validate	Release Notes	

Figure 2.23: Software management page

Once the new PAN-OS package is downloaded, it will be listed as such on the **Software** page, as shown. Click the **Install** link next to the image to start the installation:

VERSION	SIZE	SHA256	RELEASE DATE	AVAILABLE	CURRENTLY INSTALLED	ACTION	DOCUMENTATION	RELEASE TYPE
11.2.2-h1	749 MB	dc6bb08c42def20e5425...	2024/08/02 15:58:14	Downloaded ☒		Validate Export Install	Release Notes	
11.2.1	502 MB	e01057f9aa1a80cf3075...	2024/07/08 12:04:42			Validate Download	Release Notes	
11.2.0-b3	1228 MB	6486b919373239de66e...	2024/03/14 12:31:10			Validate Download	Release Notes	
11.2.0-b1	1258 MB	5641eb4d7bbc1c647b1...	2024/02/06 14:20:41			Validate Download	Release Notes	
11.2.0	1239 MB	7771ec3da10f5d9a2024...	2024/05/02 10:15:34	Downloaded	✓	Validate Export Install	Release Notes	Base
11.1.4-h1	1015 MB	7a22cc2799d5b8b9aa85...	2024/08/08 02:18:35			Validate Download	Release Notes	
11.1.4	946 MB	1be2daf822958208b56...	2024/06/27 08:17:35			Validate Download	Release Notes	
11.1.3	893 MB	15e1058633db29a434e...	2024/05/14 15:14:14			Validate Download	Release Notes	
11.1.3-h4	893 MB	e32959deb63d688b8da...	2024/08/08 13:01:43			Validate Download	Release Notes	
11.1.3-h2	940 MB	af059832438b5d01f021...	2024/07/18 08:43:21			Validate Download	Release Notes	

Figure 2.24: Image downloaded and ready to install

At the end of the installation, you will be prompted to reboot. You can skip the reboot if you want to postpone the actual upgrade to a later time. Otherwise, click **Yes**, as shown:

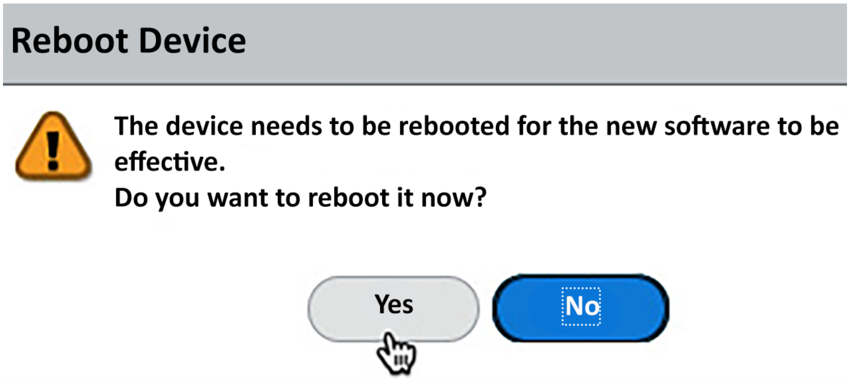
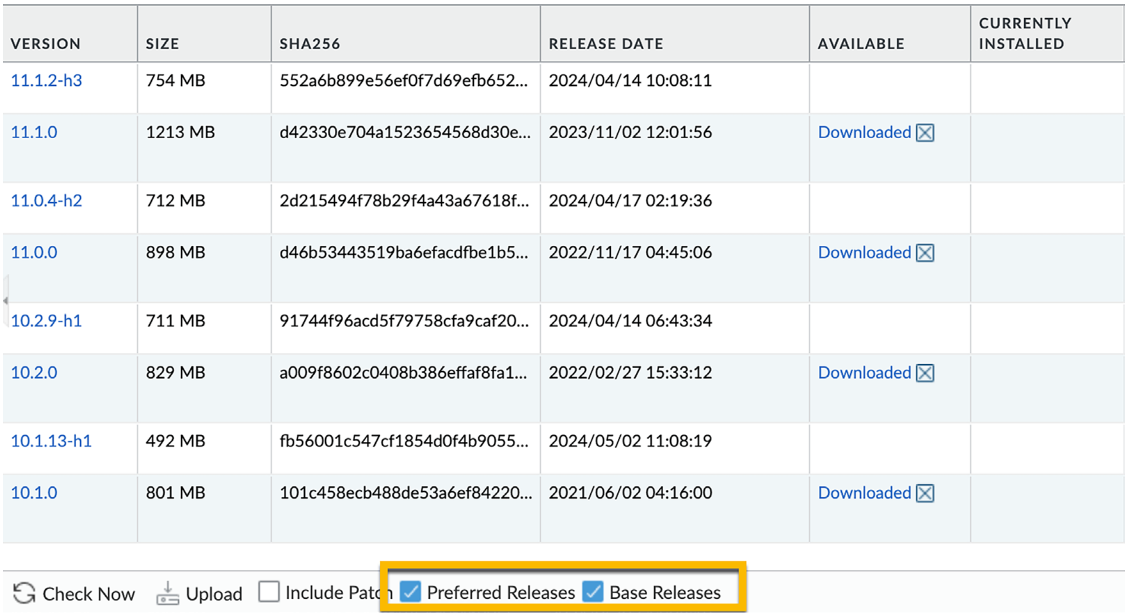


Figure 2.25: Post-installation reboot dialog

If you return to the **Software** page after the upgrade, you may notice that not all of the PAN-OS options are listed anymore; this is due to a new feature that can filter all the **Preferred Releases** and **Base Releases**, which are enabled by default. These reflect Palo Alto's recommended software versions or base images for easier access. Disabling both checkboxes will show all available OS releases again.



VERSION	SIZE	SHA256	RELEASE DATE	AVAILABLE	CURRENTLY INSTALLED
11.1.2-h3	754 MB	552a6b899e56ef0f7d69efb652...	2024/04/14 10:08:11		
11.1.0	1213 MB	d42330e704a1523654568d30e...	2023/11/02 12:01:56	Downloaded	
11.0.4-h2	712 MB	2d215494f78b29f4a43a67618f...	2024/04/17 02:19:36		
11.0.0	898 MB	d46b53443519ba6efacdfbe1b5...	2022/11/17 04:45:06	Downloaded	
10.2.9-h1	711 MB	91744f96acd5f79758cfa9caf20...	2024/04/14 06:43:34		
10.2.0	829 MB	a009f8602c0408b386effaf8fa1...	2022/02/27 15:33:12	Downloaded	
10.1.13-h1	492 MB	fb56001c547cf1854d0f4b9055...	2024/05/02 11:08:19		
10.1.0	801 MB	101c458ecb488de53a6ef84220...	2021/06/02 04:16:00	Downloaded	

Figure 2.26: Preferred Releases and Base Releases filter

Upgrade cheat sheet

This upgrade cheat sheet will help you prepare and plan your upgrade. These steps outline a solid methodology to get to a stable PAN-OS version before placing the firewall in production:

1. Go to `https://live.paloaltonetworks.com/t5/Customer-Resources/` for release recommendations.
2. In **Device** | **Software**, click on **Check Now** to load the latest list of available PAN-OS images.
3. Download and install the recommended image of your current release.
4. When the installation is completed, a dialog window will ask if you want to reboot the device. Click **Yes**.
5. Wait for the unit to boot up again and download the base image for the next major version.
6. Download and install the recommended maintenance release for the next major version.
7. When the dialog asks you to reboot the device, click **Yes**.
8. Repeat steps 5 through 7 until you're on the version you need to reach or until you reach PAN-OS 10.1, at which time, you can skip ahead up to three OS versions.

Remember that for an HA cluster or Panorama environment, you need to do the following:

- Disable preemption in the HA configuration before you start, and re-enable it after the upgrade is completed on both members of the HA cluster.
- Check both members for functionality before you start. The upgraded device will become non-functional until the lowest member has caught up (the cluster favors the lowest software member).
- Upgrade the Panorama centralized management first before upgrading any of your devices.

You have now made sure the firewall is fully set up for success by ensuring the content packages are automatically downloaded and installed, and the appropriate PAN-OS firmware has been installed. Next, we will take a look at ensuring the management interface configuration is set up securely.

Hardening the management interface

It is paramount that the management interface is kept secure, and access is limited to only those administrators that need access. It is recommended to place the physical management interface in an **Out-of-Band** (**OoB**) network, which limits exposure to the broader network. If access to the management server is needed from a different network, it is best to set up a dual-homes bastion host that mediates the connection, either by only allowing admins to log in to it and use services from there, or having it set as a (transparent) proxy with a log of all sessions and limiting the source users and IP subnets as much as possible. Admin accounts also need to be set so they only have access to the sections of the configuration they need to access and use external authentication. Let's look at these different scenarios in detail.

Limiting access via an access list

The management interface local access list can be edited by navigating to **Device** | **Setup** | **Interfaces** and clicking on the **Management Interface**:

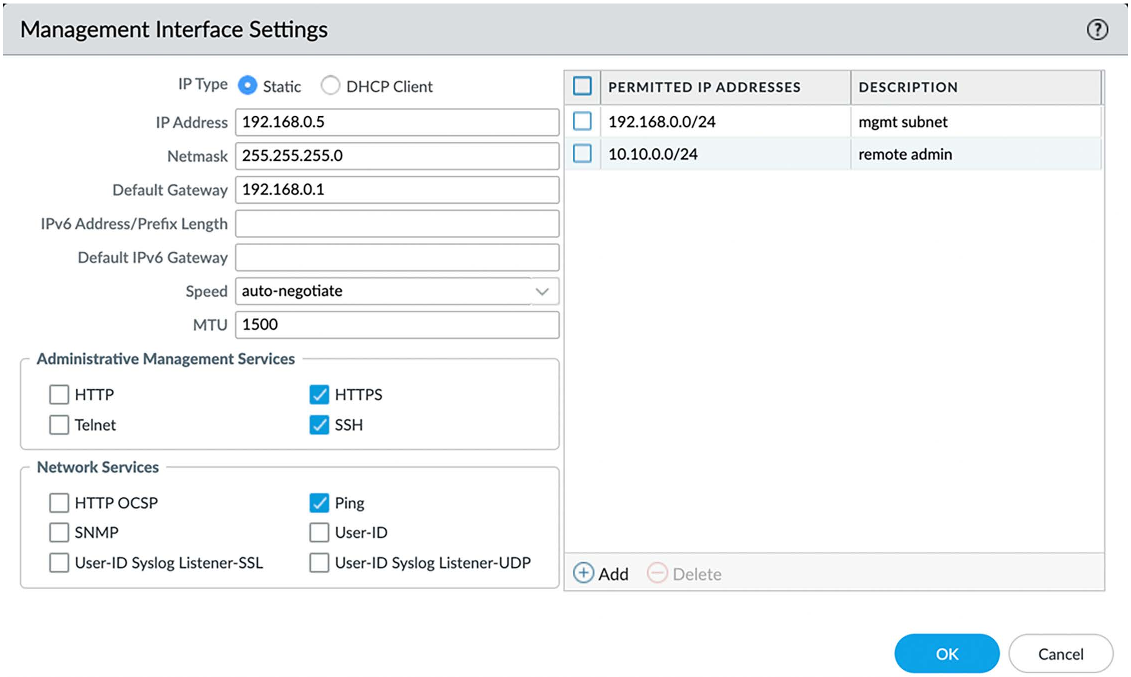


Figure 2.27: Management Interface access list

The associated CLI `configure` mode command is as follows:

```
admin@PA-440>configure
admin@PA-440# set deviceconfig system permitted-ip 192.168.0.0/24 description
"mgmt subnet"
```

You can also attach an interface management profile (shown in the following screenshot) to an interface, which enables the selected services (SSH and HTTPS, usually) on the IP address of the assigned data plane interface. This is not recommended as it introduces significant risk if not implemented properly:

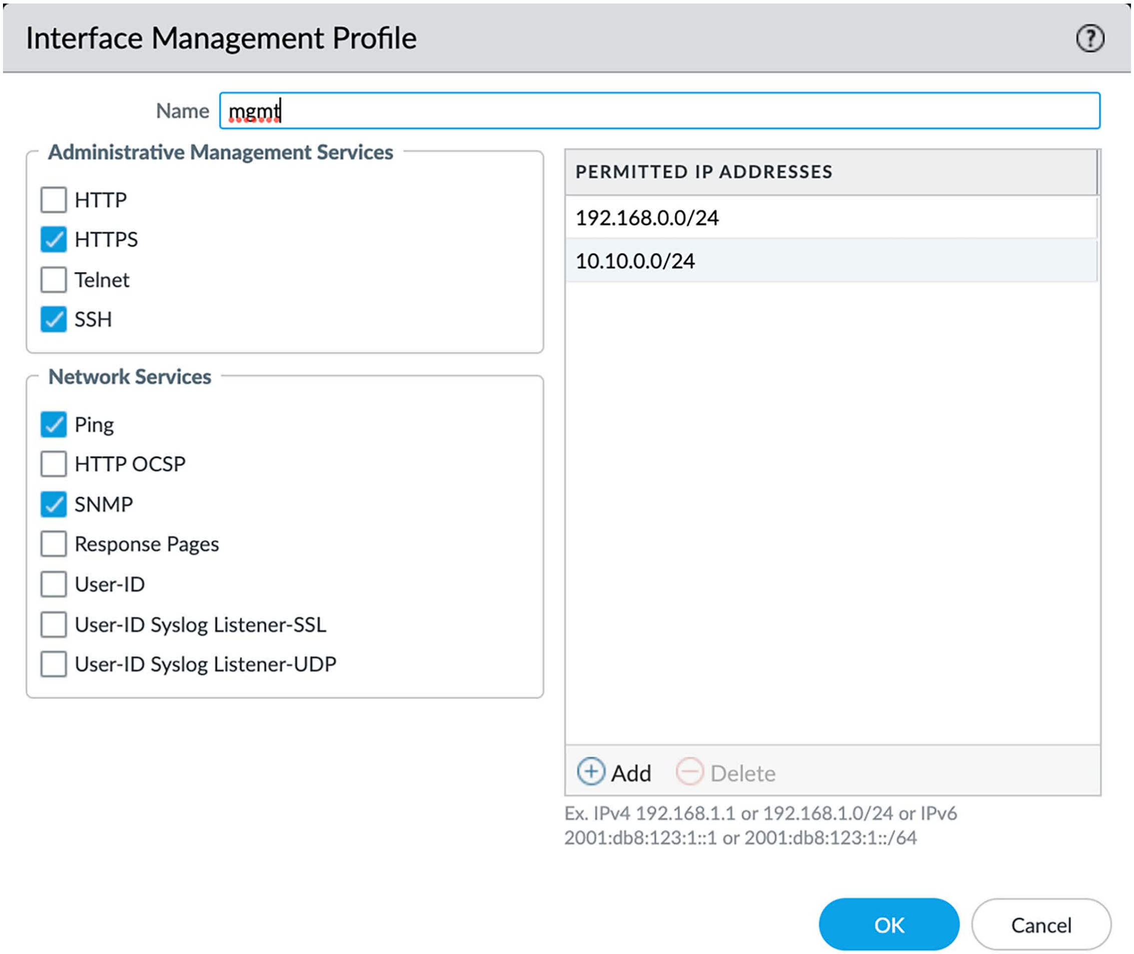


Figure 2.28: Interface Management Profile

If you must enable this profile, make sure it is on a sufficiently shielded interface, preferably a loopback interface, that has security policies associated with gaining access to the management services. As a secondary measure, also enable an access list on the profile.

These profiles can be configured in **Network** | **Network Profiles** | **Interface Management Profile** and then attached to an interface in **Network** | **Interfaces** on the **Advanced** tab of the selected interface:

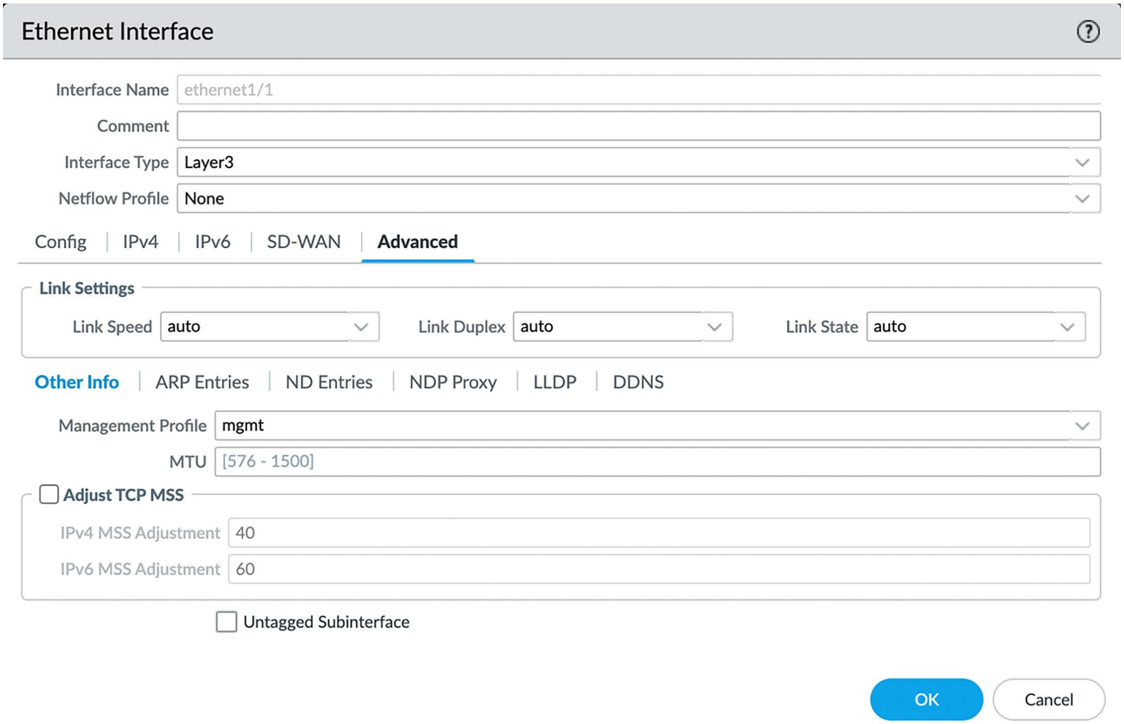


Figure 2.29: An interface management profile attached to an interface

The CLI command to create an interface management profile, set its services to HTTPS and SSH, and add an **Access Control List** (**ACL**) is as follows:

```
# set network profiles interface-management-profile mgmt https yes ssh yes
permitted-ip 10.15.15.37
```

The subsequent ACL items can be set via the following command:

```
# set network profiles interface-management-profile mgmt permitted-ip
192.168.0.5
```

As a best practice, never configure a management profile on an internet-facing interface.

Now that access to the management interface has been set, let's look at access from the management interface.

Accessing internet resources from offline management

If the management interface does not have access to the internet, this can create interesting challenges as it will not be able to retrieve updates or perform cloud lookups. A workaround is to enable service routes that route specific applications, services, or protocols via the backplane onto a designated data plane interface, allowing the management plane to reach out to the internet while its physical interface does not have access outside of its management LAN.

Service routes can be configured from the **Device** | **Setup** | **Services** menu, where you can click on **Service Route Configuration** to get the following dialog:

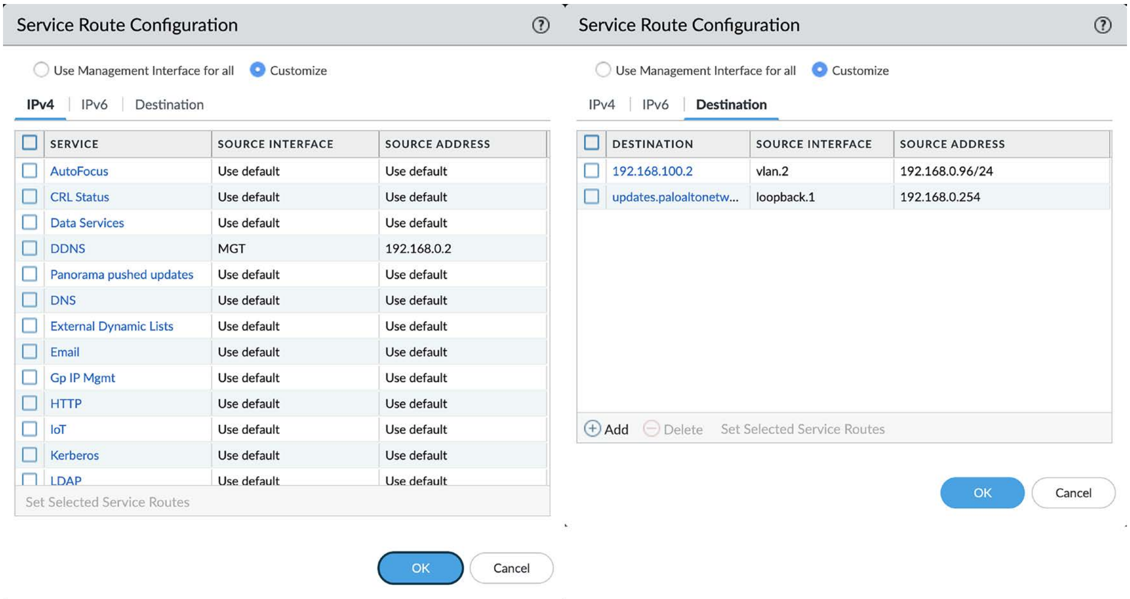


Figure 2.30: Service Route Configuration

Here's a quick overview of what the predefined service routes are used for:

- **AutoFocus:** A direct connection to the Autofocus subscription service so you can consume it directly from the firewall web interface
- **CRL Status:** Fetching **Certificate Revocation List** (**CRL**) information when processing certificates (e.g., during TLS handshakes)
- **Data Services:** Used for data uplinks from the data plane to Palo Alto cloud services, used for IoT, DLP, and SaaS subscription services
- **DDNS:** Updates the interface IP with the **Dynamic Domain Name System** (**DDNS**)
- **Panorama pushed updates:** Content and software updates deployed from Panorama (these are triggered from Panorama but initiated from the firewall)
- **DNS:** System DNS lookups
- **External Dynamic Lists:** Fetching updates for configured **external dynamic lists** (**EDLs**)
- **Email:** Email sent out by the system
- **Gp IP Mgmt:** External DHCP for GlobalProtect IP assignment

- **HSM: Hardware Security Module** is a remote system that maintains the private key for the certificate used for SSL/TLS decryption
- **HTTP:** System HTTP forwarding
- **IoT:** Live uplink to the cloud IoT service
- **Kerberos:** Kerberos authentication
- **LDAP: Lightweight Directory Access Protocol** connections
- **MDM: Mobile Device Manager** connections
- **Multi-Factor Authentication: MFA** authentication server
- **NetFlow:** NetFlow collector for network statistics
- **NTP: Network Time Protocol** updates
- **Palo Alto Networks Services:** Updates from Palo Alto Networks and WildFire sandbox connections
- **Panorama:** Connection to the Panorama management server
- **Panorama Log Forwarding:** Logs sent to a log collector (used by the PA_5200 chassis only)
- **Proxy:** Connections to the system proxy
- **RADIUS: Remote Authentication Dial-in User Service** connections
- **SCEP: Simple Certificate Enrollment Protocol** connections for requesting and distributing client certificates
- **SNMP Trap:** Sending SNMP statistics to an SNMP server
- **Syslog:** Syslog messaging server
- **TACACS+: Terminal Access Controller Access Control System Plus** authentication
- **UID Agent:** User ID agent connections
- **URL Updates:** URL filtering live lookup service
- **VM Monitor:** VM monitoring
- **WildFire Private:** Connection to the private WildFire server
- **Ztp:** Connection to the ZTP service

Once you set the radio button from **Use Management Interface for all** to **Customize**, you will be able to select which source interface will be used for each service. From the **Destination** tab, you can also add specific IP addresses or entire subnets that need to be routed out of a specific interface. The routing table used by the target interface will determine how the session is routed to the destination, so service routes can only be attached to OSI Layer 3 interfaces.

The associated CLI configuration command to set a service route is as follows:

```
# set deviceconfig system route service dns source interface ethernet1/8
address 192.168.0.15
```

If you want to see a full list of all the available services, hit the *Tab* key after typing `service`:

```
# set deviceconfig system route service <Tab>
  autofocus                    AutoFocus Cloud
```

```
crl-status                  CRL servers
ddns                        DDNS server(s)
...
```

This will enable access to resources that are normally not accessible through the management network. In the next section, we'll prepare administrator accounts and provide access as needed.

Admin accounts

The "admin" account is probably one of the most abused accounts in internet history, so your next task is to get rid of it and replace it with named accounts. Instead of the default "admin" account, it is best practice to use named accounts so changes can be tracked by the user and personalized access can be granted easily.

Navigate to the **Device | Administrators** tab to view local administrators. When creating new administrator accounts there are two types of accounts available, **Dynamic** and **Role Based**, which you can select by setting the **Administrator Type** toggle:

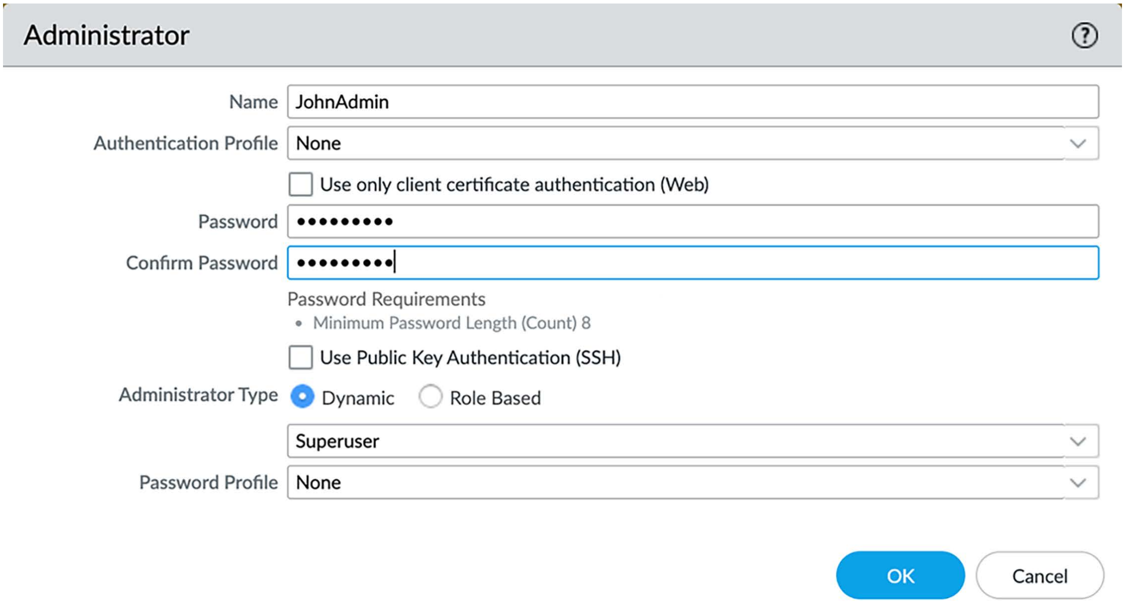


Figure 2.31: Creating a new admin account

First, we'll take a look at dynamic account profiles and their benefits.

Dynamic accounts

Dynamic accounts may be of the following types:

- **Superusers** can do everything. They are the only users that are allowed to perform certain tasks like creating other administrator accounts or creating a tech support file.
- **Device administrators** can do everything besides create new users or virtual systems.

- Virtual system-capable devices also have **virtual system administrators**, who are also device administrators and are restricted to one or several specific virtual systems.
- There are also read-only flavors that can view everything but not make changes.

Your first account will need to be a new superuser to replace the default admin account.

Role-based administrators

Once all the required superusers and device administrators are created, additional role-based administrators can be added for teams that only require limited functionality. Role-based administrators can be customized down to individual menu items so that they can do anything or have read-only or no access.

The roles can be configured through the **Device** | **Admin Roles** menu:

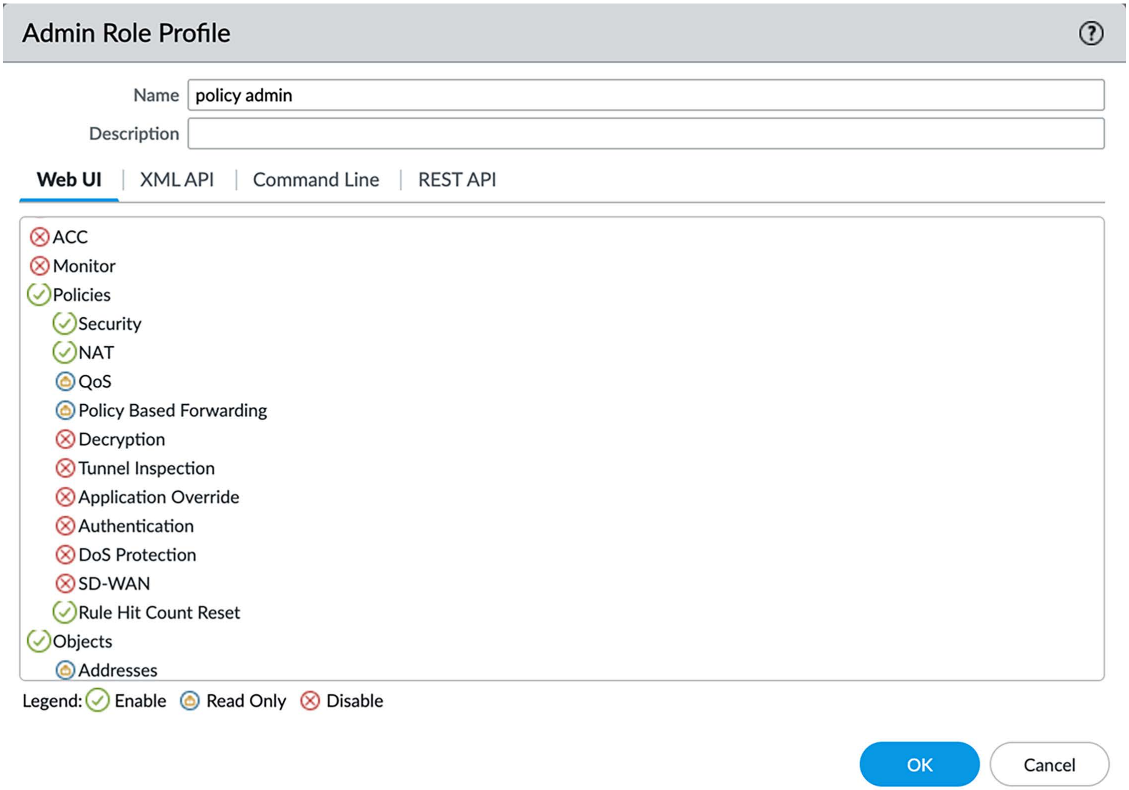


Figure 2.32: Admin Role Profile

Set each topic to one of the following options by clicking the icon to cycle to the option you need:

- A red cross indicates that these administrators will not see the menu item
- A lock indicates that the admin will be able to see objects or menu items, but not make any changes

- A green checkmark indicates that the admin has full access to this menu item and can make changes to objects or configurations within it

In the **XML API/REST API** tabs, each role can be granted or denied access to certain API calls:

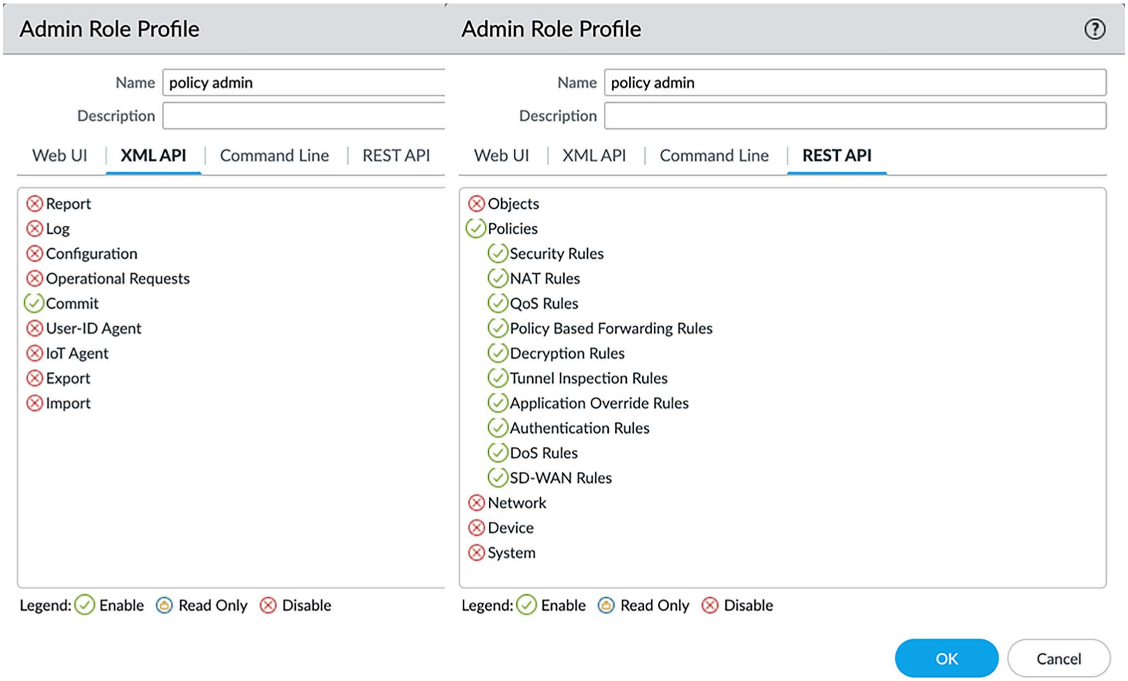


Figure 2.33: XML API/REST API

In the **Command Line** tab, each role can be granted a certain level of access or denied access altogether:

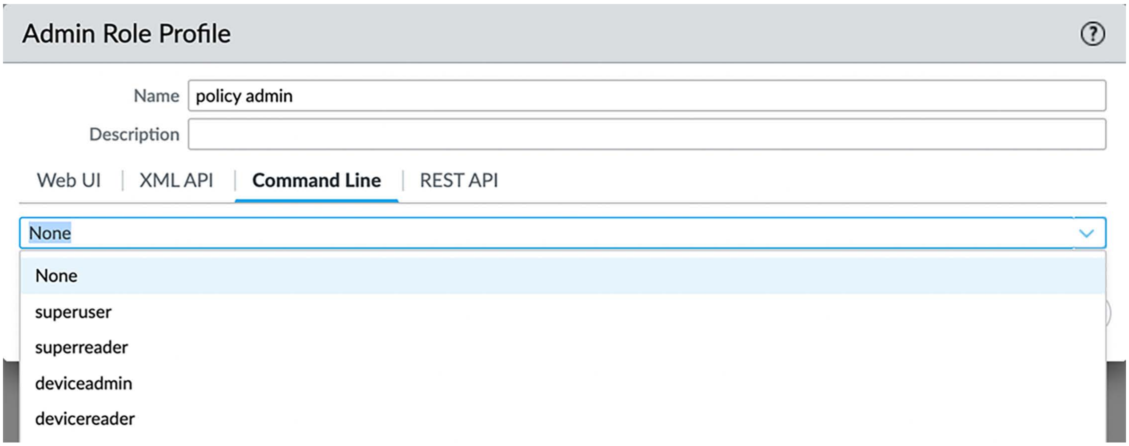


Figure 2.34: The Command Line permissions

Now that we can set up administrator accounts, we should also create a password security profile to prevent weak password discipline.

Password security

You will need to add a password profile by going to **Device | Password Profiles** to ensure that the password is changed on a regular basis:

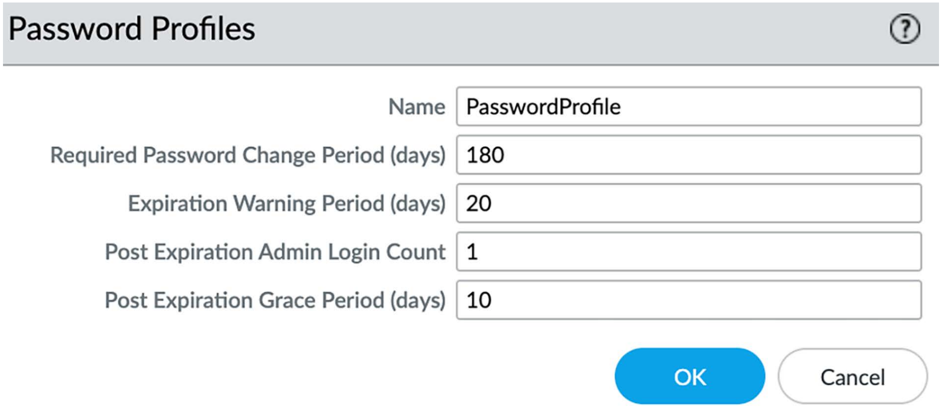


Figure 2.35: Password Profiles

These are the configurable settings in **Password Profiles:**

- The change period indicates how long a password is valid
- The expiration warning pops up a warning when an admin logs on if their password is about to expire
- The post-expiration login feature allows the admin to log on a certain number of times, even after their password has expired
- The post-expiration grace period indicates how long an admin will be able to log on after their account has expired before it is locked permanently and will require intervention from a different admin

Additionally, you should enforce a minimum password complexity for local accounts to ensure no weak passwords are used by administrators via **Device** | **Setup** | **Management** | **Minimum Password Complexity:**

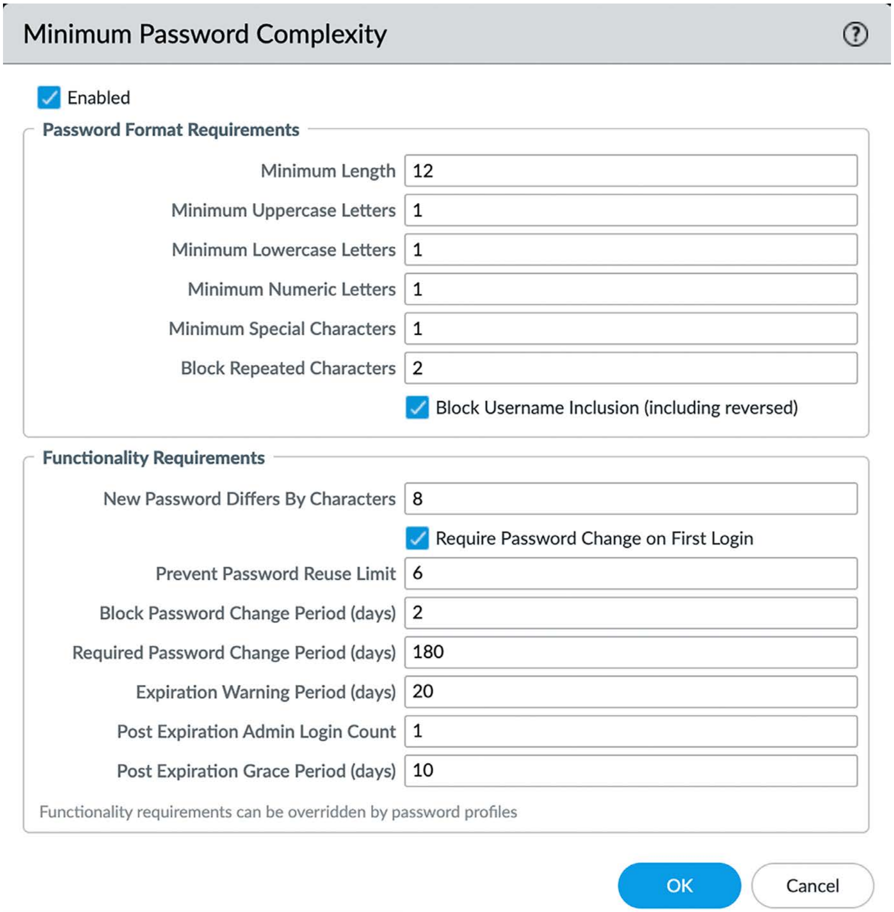


Figure 2.36: Minimum Password Complexity

> NIST has an extensive guideline on authentication and life cycle management that can be found at `https://pages.nist.gov/800-63-3/sp800-63b.html`.

Let's now look at the external authentication factors.

External authentication

It is best to use external authentication factors, such as Kerberos, LDAP, RADIUS, or SAML, to keep control over credentials in a centralized system, which enables admins to only change passwords once for multiple devices or to be locked out of all critical infrastructure at once if they leave the organization.

It is a good safeguard to keep one "break-glass" local account in case the management interface loses all access to the external authentication servers. This account should only be used in case of emergency and an alert should be sent when the account is used to log on. See *Chapter 10, Logging and Reporting*, on how to set up a profile that can send out an alert if the account is used.

You first need to create a server profile from the **Device | Server Profiles** menu. Each server type has its own configuration parameters. The following profiles are available:

- **TACACS+** (**Terminal Access Controller Access Control System Plus**) is an authentication protocol developed by Cisco that is still used in many environments for terminal access authentication.
- **LDAP** (**Lightweight Directory Access Protocol**) is probably one of the most commonly available protocols to authenticate against directory services and will work with Microsoft Active Directory, eDirectory, and custom LDAP servers.
- **RADIUS** (**Remote Authentication Dial-in User Service**) is an open standard authentication protocol that is widely used in remote access authentication.
- **Kerberos** is an authentication protocol that is mostly used for single sign-on and relies on a negotiation that does not require the exchange of passwords.
- **SAML** (**Security Assertion Markup Language**) uses an XML framework to exchange security information and is mostly used with cloud-based **identity providers** (**IdPs**). It integrates with **multi-factor authentication** (**MFA**) very easily.
- Multi-factor profiles allow for several built-in MFA providers to be configured and added separately to an authentication profile.

Let's take a look at each of these profiles and how to set their basic configuration.

The TACACS+ server profile

TACACS+ requires you to choose between **Password Authentication Protocol** (**PAP**) and **Challenge-Handshake Authentication Protocol** (**CHAP**) and set the secret associated with connecting to the TACACS+ authentication server.

Optionally, you can set the profile so that it can only be used for administrator authentication, and set the profile to use a single session for all authentication events, rather than a new session per authentication event:

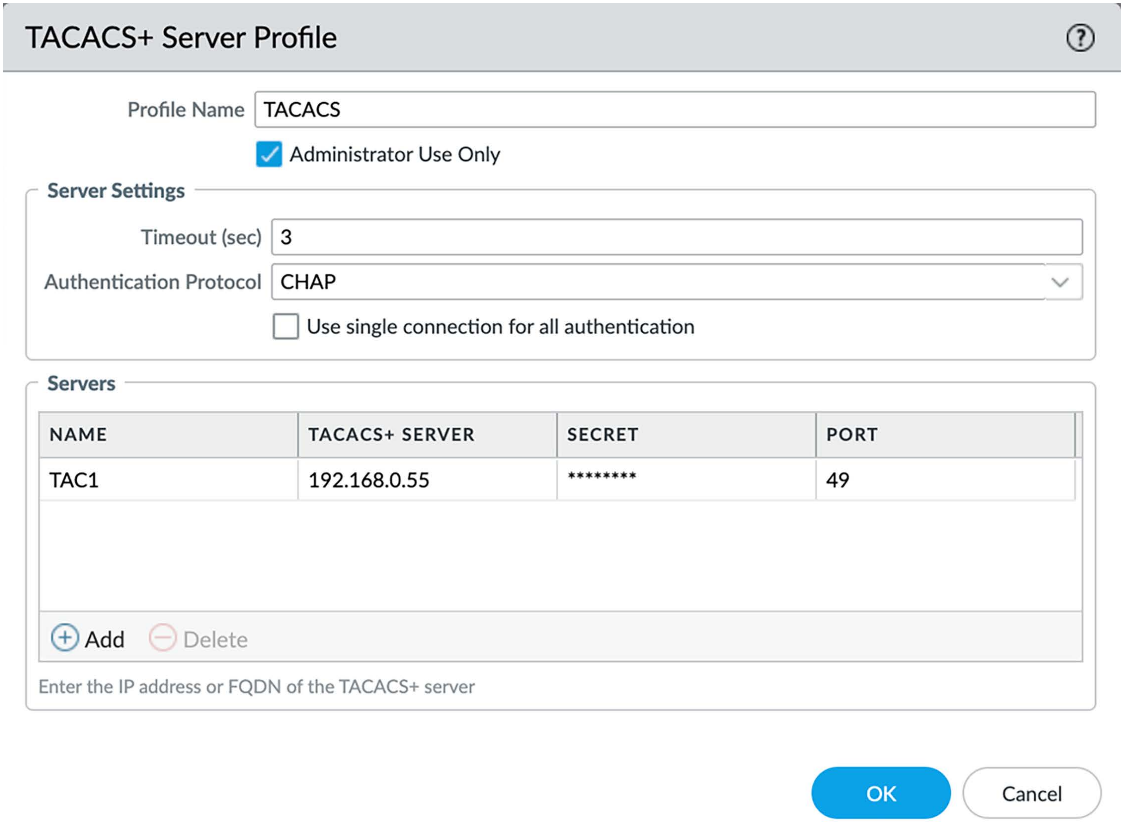


Figure 2.37: TACACS+ Server Profile

While TACACS+ is somewhat rare, as it is mostly used in industrial environments or to manage infrastructure equipment, LDAP authentication is very common as it's mostly applied in user environments.

The LDAP server profile

For an LDAP profile, you need to provide the type of the LDAP server, which can be **active-directory**, **E-directory**, **sun**, or **other**.

You need to provide a **Base DN** value, which is the domain name (or the distinguished name) of the LDAP tree. The **Bind DN** field is for the user account that will be used to connect to the LDAP server and perform the request and its password. **Bind DN** can be fully qualified, as shown in the following screenshot, or be a **user principal name** (**UPN**) formatted as `user@domain`:

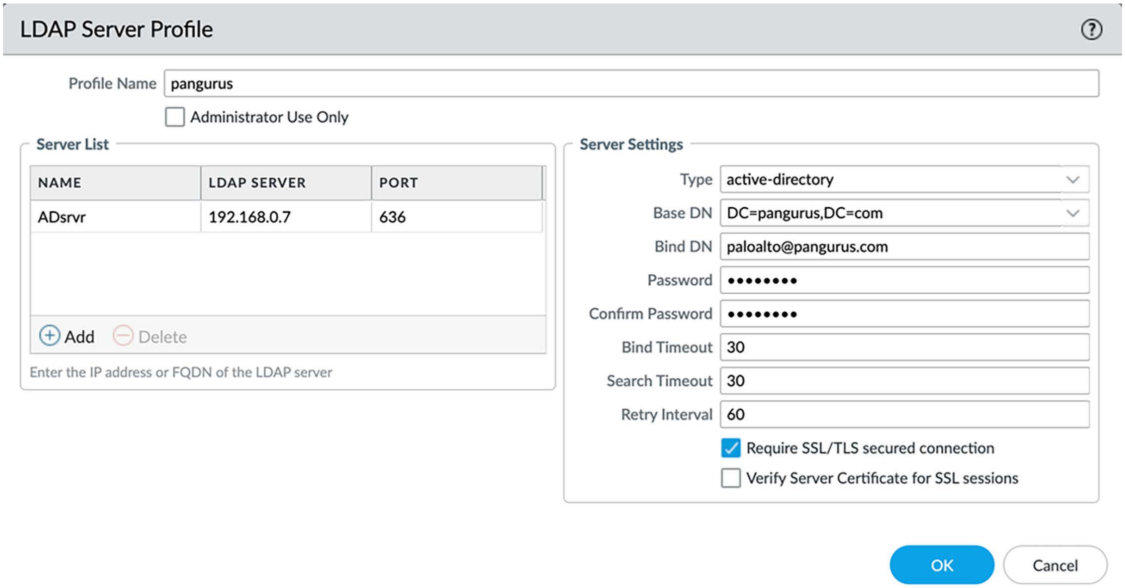


Figure 2.38: LDAP Server Profile

One thing to remember is that when you configure the server IPs and you have **Require SSL/TLS secured connection** enabled, the default port for LDAPS is `636`, rather than `389`.

If your LDAP server has an externally signed certificate, enable **Verify Server Certificate for SSL sessions** to ensure the authenticity of your server. For the certificate check to work, the LDAP server root and intermediary certificates need to be in the device certificate store in **Device | Certificate Management | Certificates | Device Certificates**. The server name in the profile must match the **fully qualified domain name** (**FQDN**) certificate and `Subject AltName` attribute for this check to pass.

The RADIUS server profile

RADIUS is one of the most popular authentication methods and supports the following authentication protocols:

- PEAP-MSCHAPv2: **Protected Extensible Authentication Protocol** (**PEAP**) with Microsoft CHAP v2 provides improved security over PAP or CHAP by transmitting both the username and password in an encrypted tunnel.
- PEAP with **Generic Token Card** (**GTC**): PEAP with GTC enables the use of one-time tokens in an encrypted tunnel.
- EAP-TTLS with PAP: **Extensible Authentication Protocol** (**EAP**) with **Tunneled Transport Layer Security** (**TTLS**) and PAP is used to transport plain text credentials for PAP in an encrypted tunnel. EAP-TTLS uses certificates to secure the connection and should be the preferred protocol as it is the most secure.
- **CHAP:** Used if the RADIUS server does not support EAP or PAP or is not configured for it.
- **PAP:** Used if the RADIUS server does not support EAP or CHAP or is not configured for it.

Palo Alto Networks uses vendor code 25461.

Like the other profiles, RADIUS can be set so that it is only used for administrator authentication. The **Allow users to change passwords after expiry** option is limited to GlobalProtect users if the profile is also used to authenticate GlobalProtect inbound connections.

The **Make Outer Identity Anonymous** option ensures the admin username is not visible for anyone sniffing the authentication sessions if PEAP-MSCHAPv2, PEAP with GTC, or EAP-TTLS are used and the server supports this:

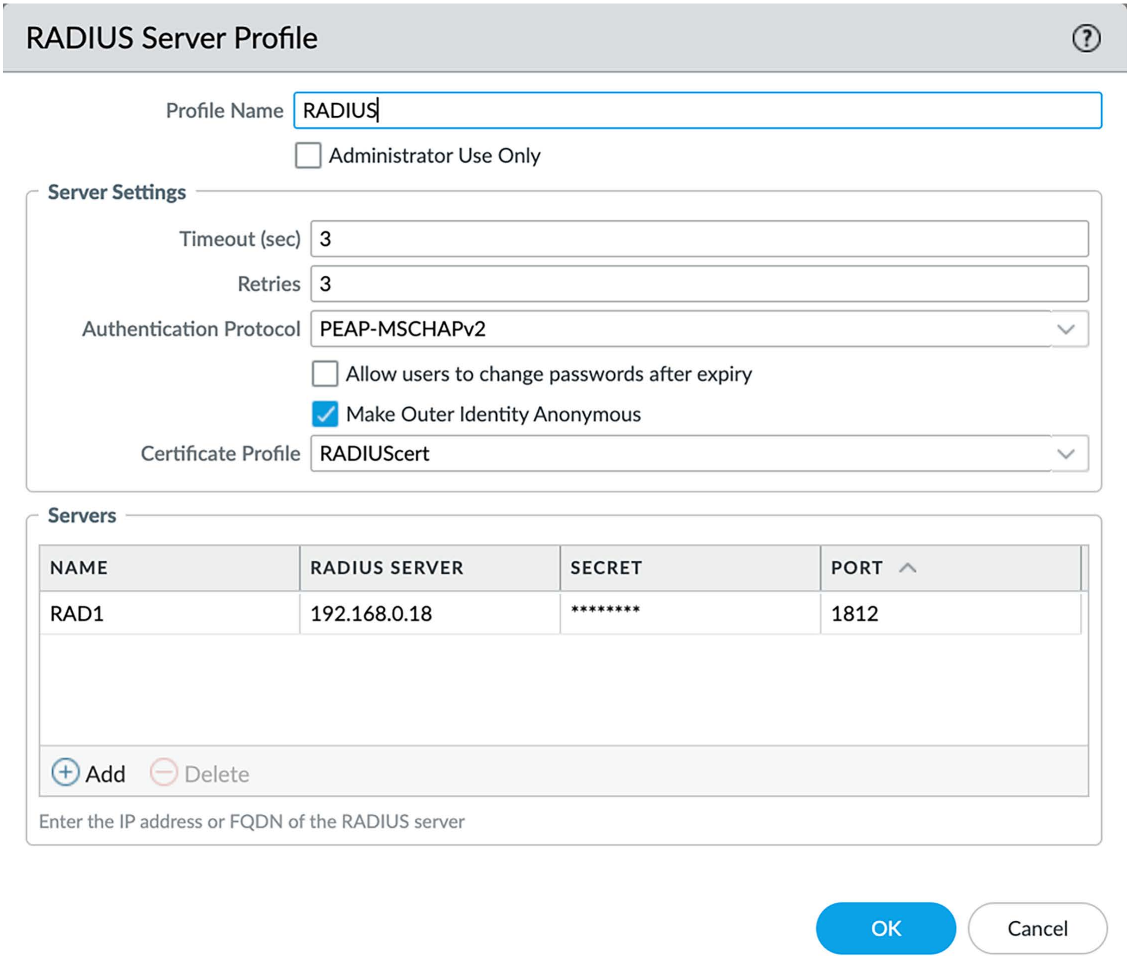


Figure 2.39: RADIUS Server Profile

The certificate verification for RADIUS server profiles requires a certificate profile that allows more checks to be performed than just having the root certificate in the trusted certificate store compared to TACACS+. Several mechanisms can be used to verify server validity and actions can be taken if particular conditions are met with the certificate check, such as opting to allow or block a session if the certificate is valid but has expired:

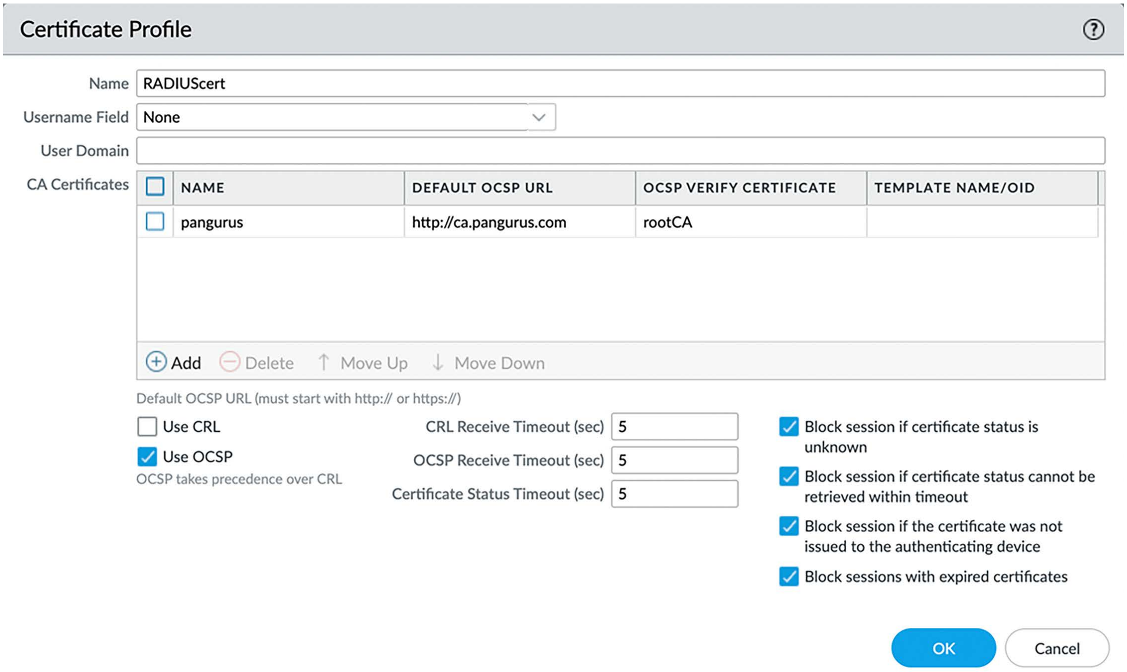


Figure 2.40: Certificate Profile

The Kerberos server profile

The Kerberos server profile is very simple to configure as it only requires an IP or FQDN and a port number, but it does require a few specific configuration settings:

- The firewall needs to have the domain set in **Device** | **Setup** | **Management** | **General Settings**
- The firewall is synced to a **Network Time Protocol** (**NTP**) server from **Device** | **Setup** | **Services** so that its clock is in sync with the local `ActiveDirectory` server
- Its DNS servers need to be set to internal DNS servers that are joined to the domain, rather than external DNS servers

This is shown in the following screenshot:

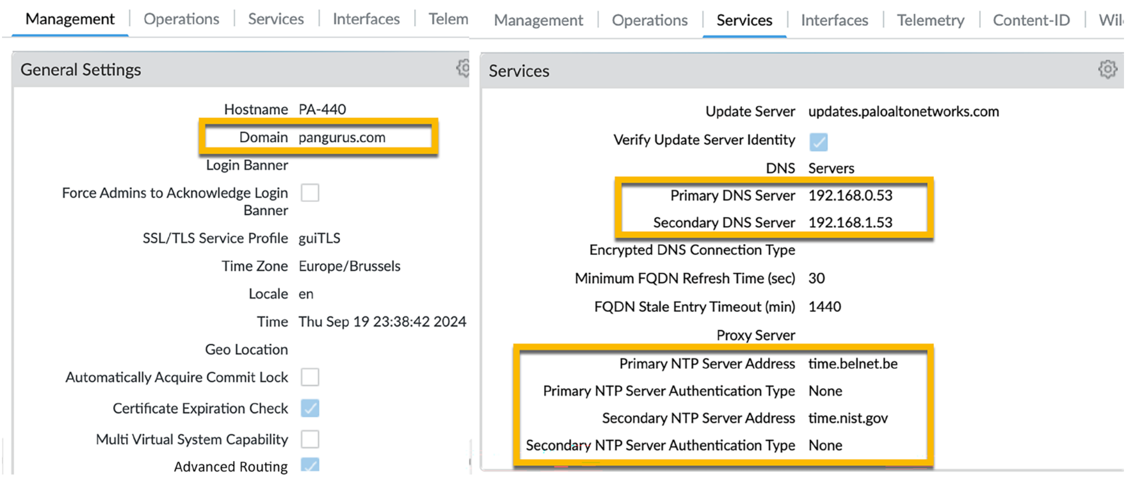


Figure 2.41: Kerberos requirements

With single sign-on adoption on the rise and many authentication services making a move to the cloud, the popularity of SAML authentication is also increasing.

The SAML server profile

The SAML profile enables authentication against an external **single sign-on** (**SSO**) provider (such as PingID and Okta).

Your IdP should provide you with the following:

- An identifier so that it can certify whether the authentication session originates from you
- The root and intermediary certificates, which you can load to **Device** | **Certificate Management** | **Device Certificates** to verify the identity of the SSO and **single log-out** (**SLO**) sites
- An SSO URL
- An SLO URL

SAML HTTP binding provides an option to **post**, which sends a Base64-encoded HTML form to the IdP, or **redirect**, in which case the firewall will send Base64-encoded and URL-encoded SSO messages within the URL parameters.

You can sign SAML messages to the IdP with a certificate for signing requests, which can be configured in the SAML Identity Provider Server Profile:

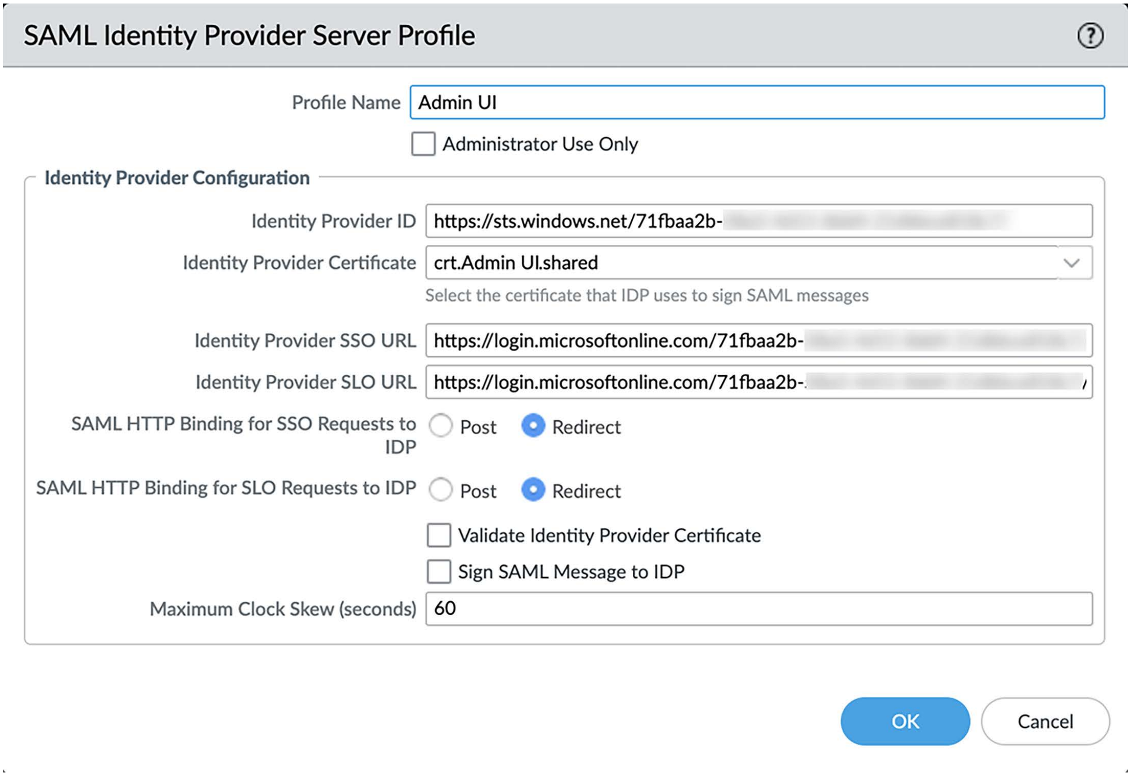


Figure 2.42: SAML Identity Provider Server Profile

The MFA profile

It is highly recommended to add MFA as regular passwords require good hygiene from each administrator and could be inadvertently shared, stolen, or guessed. If your regular authentication does not support integrating an additional factor like SMS, push messages, tokens, etc., you can add an MFA profile and add the extra profile to the authentication profile.

Currently, four MFA providers are available as standalone profiles: Duo, Okta, RSA, and PingID. To configure an MFA profile, you will need some parameters from the provider.

Settings such as the API host may depend on your geolocation, and keys and secrets will be unique identifiers to your account:

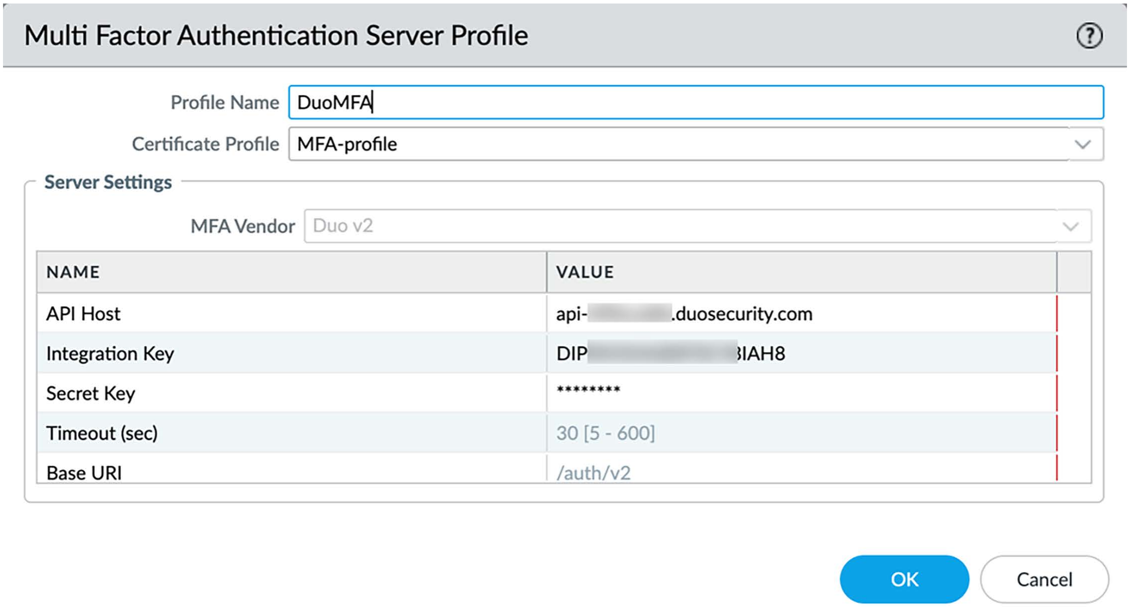


Figure 2.43: Multi Factor Authentication Server Profile

Once the appropriate server profiles have been set up, they need to be added to an authentication profile.

Setting up the authentication profile

Now that the appropriate server profile has been configured for your environment, you can go ahead and set up an authentication profile, which will set the stage for the administrators to sign in. Go to **Device** | **Authentication Profile** and create a new authentication profile.

The **Authentication** tab lets you choose the type of authentication you want to use for this profile; this will match the server profile you configured in one of the previous steps. You can then add additional parameters, such as setting `sAMAccountName` or `userPrincipalName` for the LDAP login attribute. **Username Modifier** lets you control how the username that the end user enters is translated and sent to the authentication server. This allows you to simply forward what the user inputs or add the user domain in UPN format (`user@domain`) or traditional `domain\user` backslash format:

- `%USERINPUT%`
- `%USERDOMAIN%\%USERINPUT%`
- `%USERINPUT%@%USERDOMAIN%`

This may be necessary in a multi-domain forest environment (e.g., `acme.com`, `acme.local` and `mydomain.net` in the same directory):

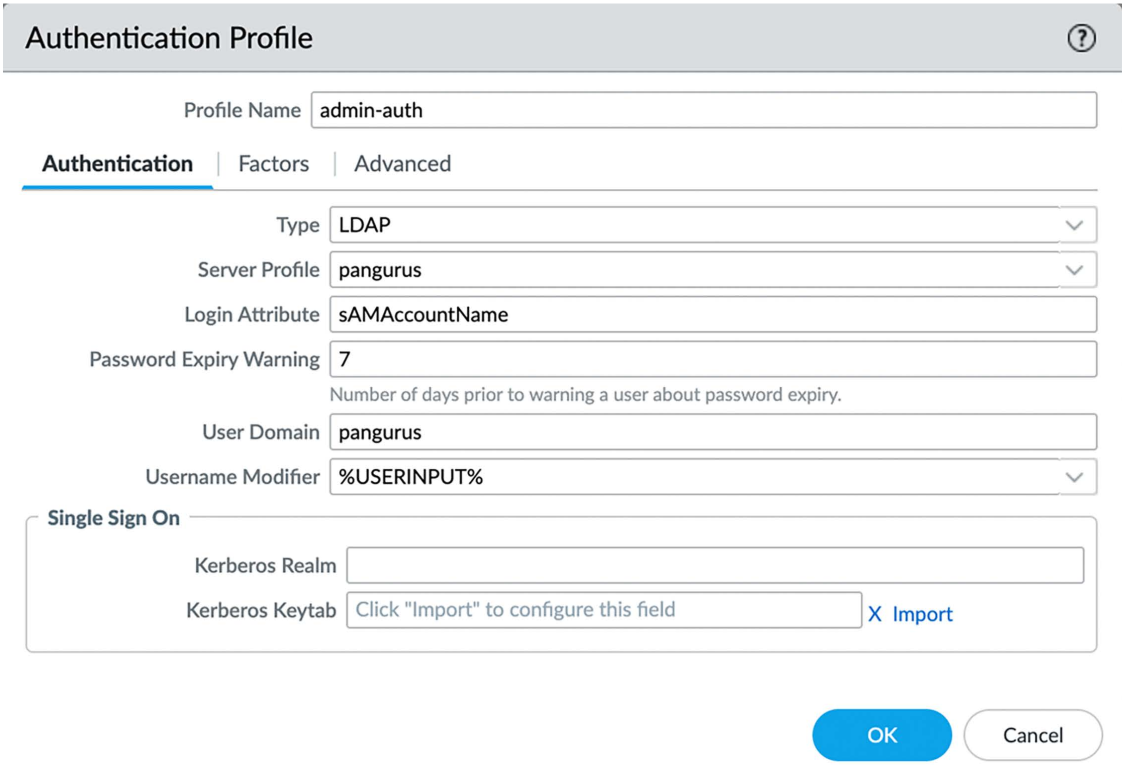


Figure 2.44: Authentication Profile LDAP example

In the **Factors** tab, you can add a profile for an MFA policy that will trigger the secondary authentication once a user logs in:

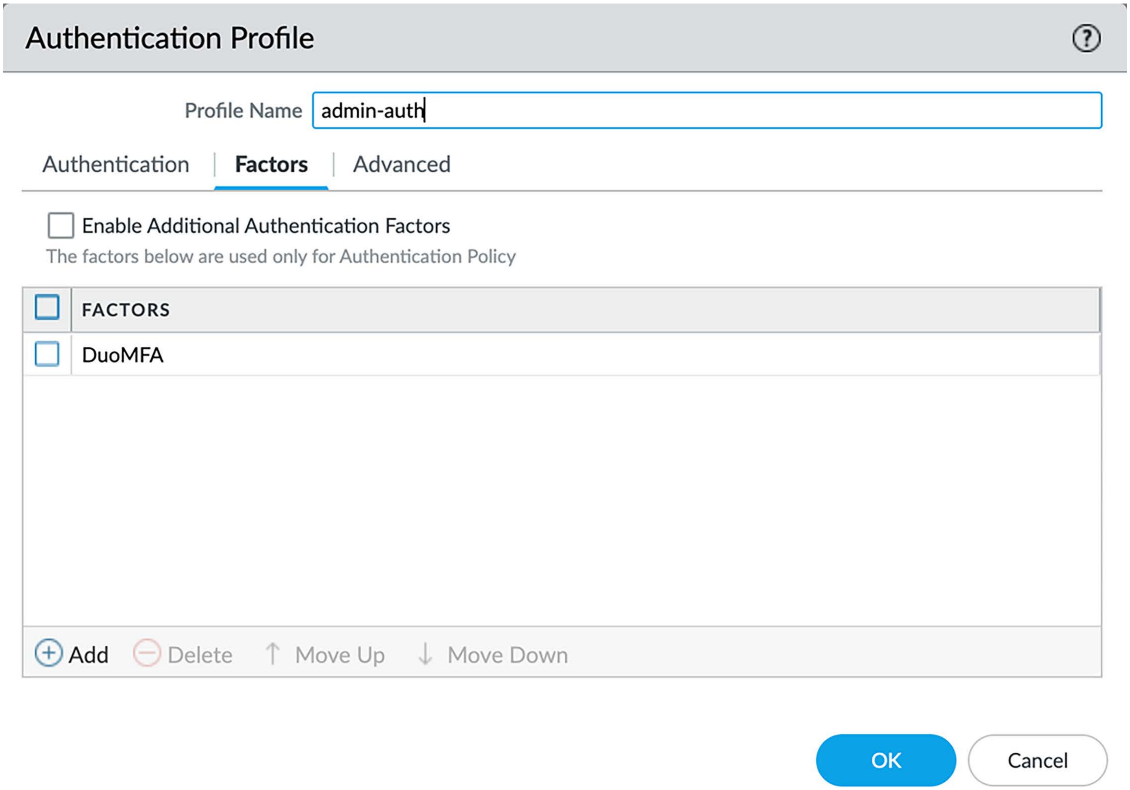


Figure 2.45: Authentication Profile MFA

The **Advanced** tab creates a bit of a chicken-and-egg situation as it requires you to tell the firewall which usernames or user groups are allowed to attempt authentication, but the list of users is only populated after you have properly set up the user ID. If you have not set up a user ID group mapping yet, set the user to **all** until you can return and narrow down the list to the actual admin user groups or usernames.

For security purposes, you should configure a lockout policy that prevents logins for an amount of time after several failed attempts to log in:

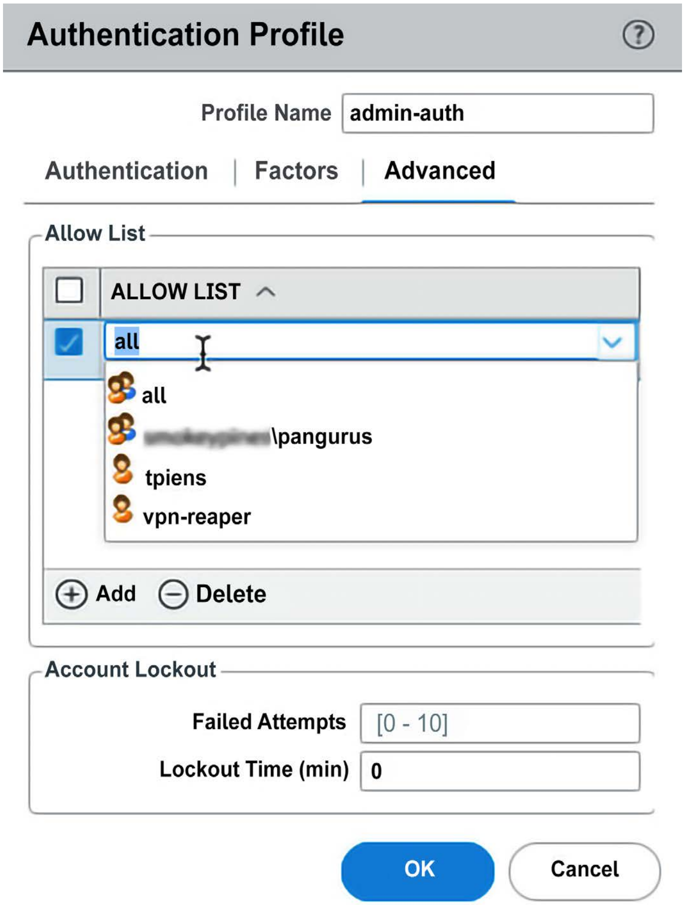


Figure 2.46: Authentication Profile allowed users

When the profile has been created, you can use it instead of a static password when creating administrator accounts. The Authentication Profile can be set in the administrator account to replace the static password:

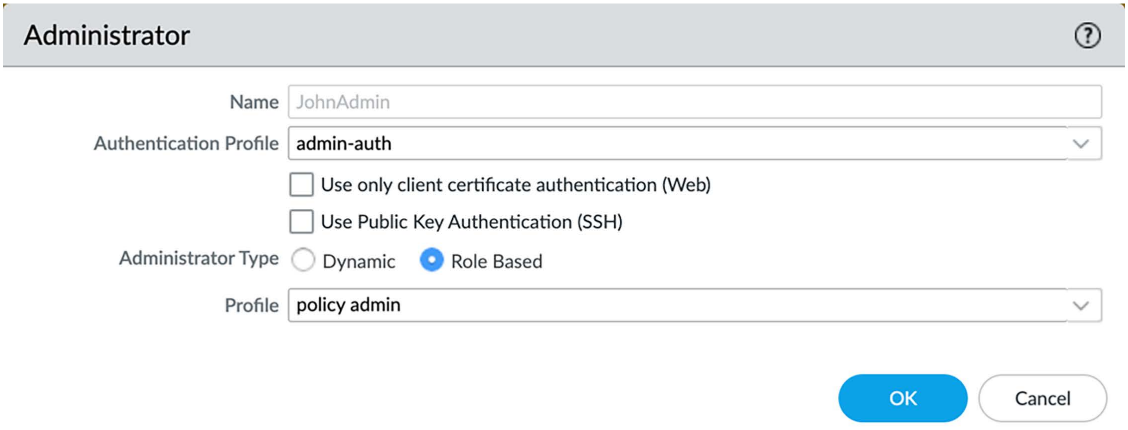


Figure 2.47: Admin account with an authentication profile

With the topics we covered in this last section, you are now able to set up admin accounts that are not only restricted to the access they need (complying with RBAC requirements) but can also leverage external authentication mechanisms and add MFA to strengthen administrator access and prevent unauthorized access. In the next section, we will learn about the different types of interfaces.

Understanding the interface types

When you open the **Network** | **Interfaces** menu, you will see an assortment of physical interfaces. There are several different interface types that will cause an interface to behave in a specific way. We will first cover the four basic interface types and continue with the more specialist ones after:

- **Virtual Wire** (**VWire**) is a "Layer 1" passthrough set of two interfaces
- Layer 3 is routed interfaces
- Layer 2 is "switched" interfaces that rely on VLAN tags rather than routing tables
- Tap is a "sniffing" interface

Let's discuss them in more detail.

VWire

Just as the name suggests, VWire is intended to be a "bump in the wire." VWire always consists of two physical interfaces—no more and no less. There is no low-level interference with VLAN tags and there are no routing options; packets are inspected in flow.

Using a VWire interface can be an easy way to "drop in a firewall" without needing to interfere with an existing routing or switching environment. It easily plugs in in front of an ISP router or can be placed in between a honeypot and the network to add a layer of detection.

Out of the box, a firewall will have a default VWire set up on `ethernet1/1` as the outside interface and `ethernet1/2` as the inside interface with a security rule to allow all outbound traffic.

Before you can create a VWire interface, you first need to set two interfaces to the **Virtual Wire** type and assign each of them to a different zone:

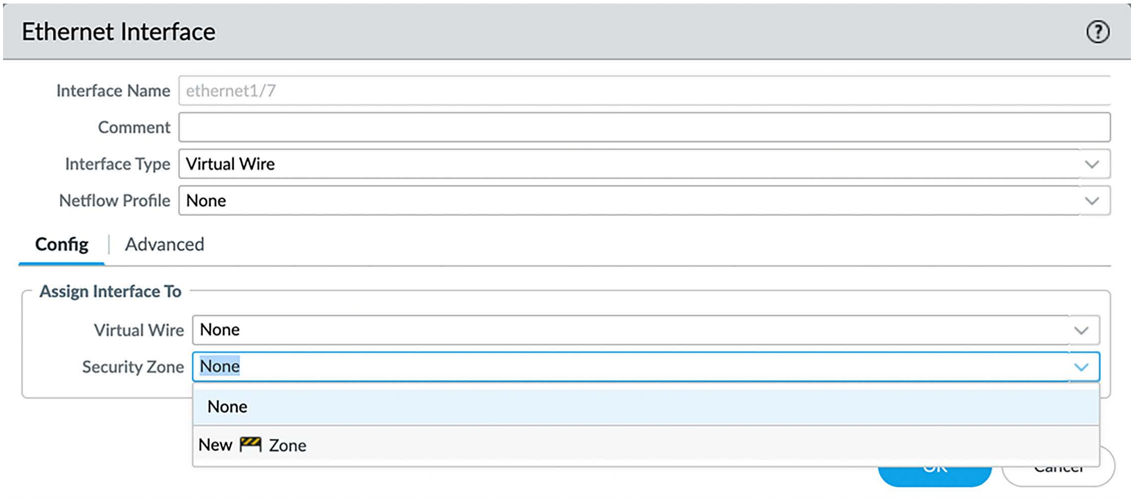


Figure 2.48: VWire interface

You can now connect both interfaces to a VWire profile by going to **Network | Virtual Wires** and creating a new VWire profile.

As illustrated in the following screenshot, you will need to select the two interfaces with which you will form a VWire connection. If the VWire interface is placed over a trunked link (one that contains the `VLAN/802.1Q` tags), you need to indicate which ones are allowed. If you want to allow all tags, set `0-4094`. If you want to add single tags or ranges, you can add integers or ranges, separated by commas (for example, `5,15,30-70,100-110,4000`). **Multicast firewalling** needs to be checked if you want to be able to block or otherwise apply security policies to multicast traffic. If unchecked, multicast is forwarded across VWire without checking the security rules. Enabling **Link State Pass Through** brings the opposite interface down if one side loses its connection.

This ensures that both the client and server sides see the link go down and respond accordingly:

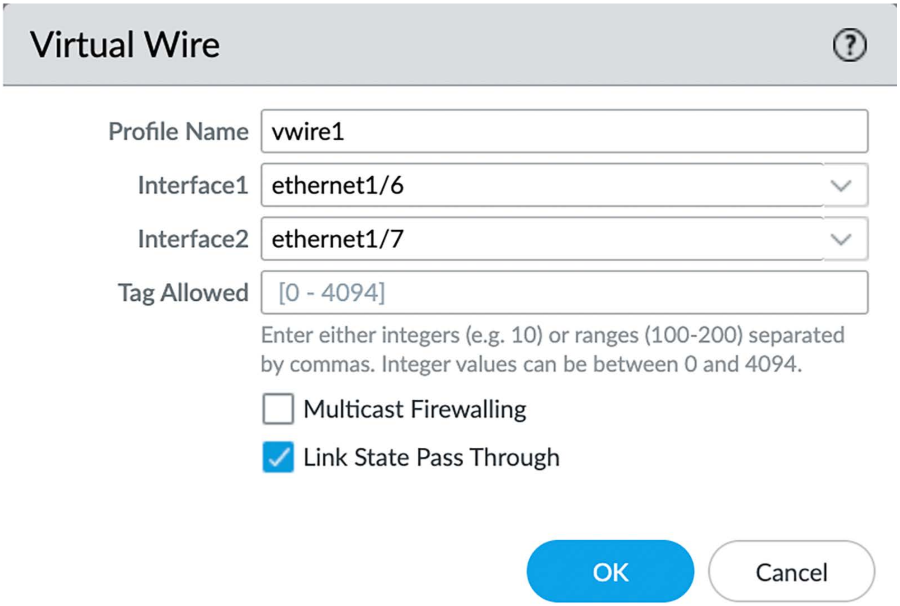


Figure 2.49: VWire configuration

The Layer 3 interface

A Layer 3 interface is a routed interface. This means it has an IP address and can be used as a default gateway for clients on the inside connected to it or a next hop for a routing device. On the outside, it can communicate with ISP routers and forward traffic out to the internet. We'll first look at the interface.

Exploring the interface

In the **Config** tab of the interface, you need to assign a **Virtual Router** (**VR**) and a security zone (we'll learn more about the VR later in this chapter, so don't worry about it just yet). This zone will represent the subnet(s) connected to it when traffic needs to flow from one interface to another:

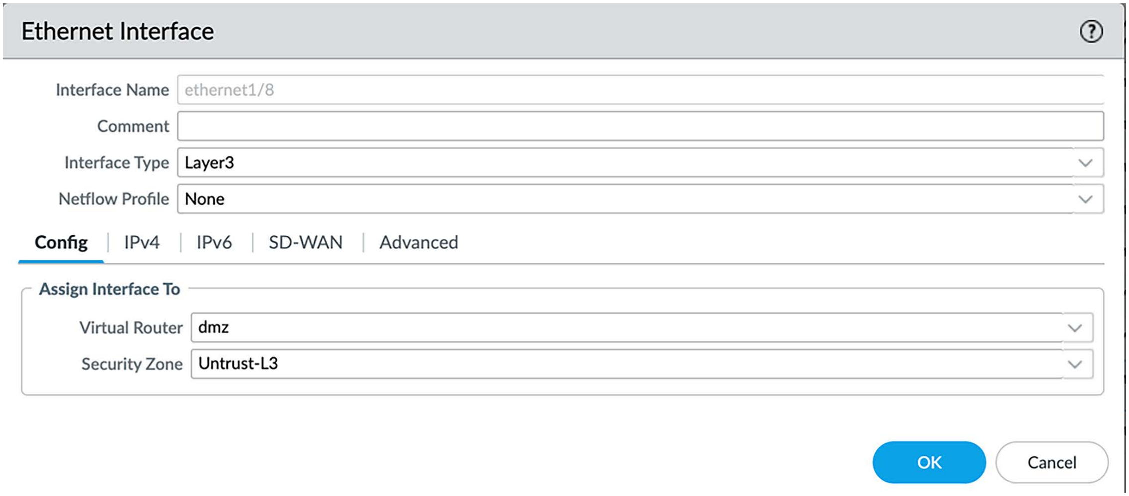


Figure 2.50: Layer 3 interface configuration

The IP configuration can be statically configured as an IP/subnet. If needed, multiple IP/subnets can be added to represent additional networks that are directly connected to the interface:

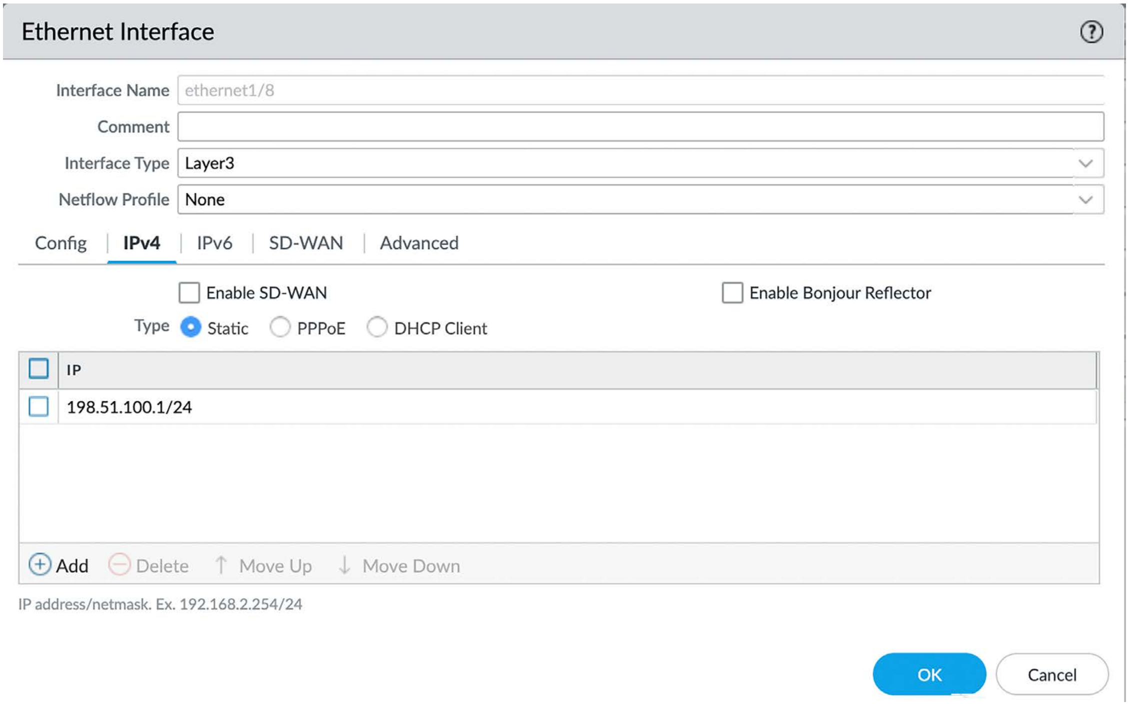


Figure 2.51: Layer 3 interface IP

A Layer 3 interface can also be set as a **Point-to-Point Protocol over Ethernet** (**PPPoE**) client if the upstream connection is provided by a broadband ISP over cable or DSL. In this case, in the **General** tab, the ISP authentication username and password can be configured:

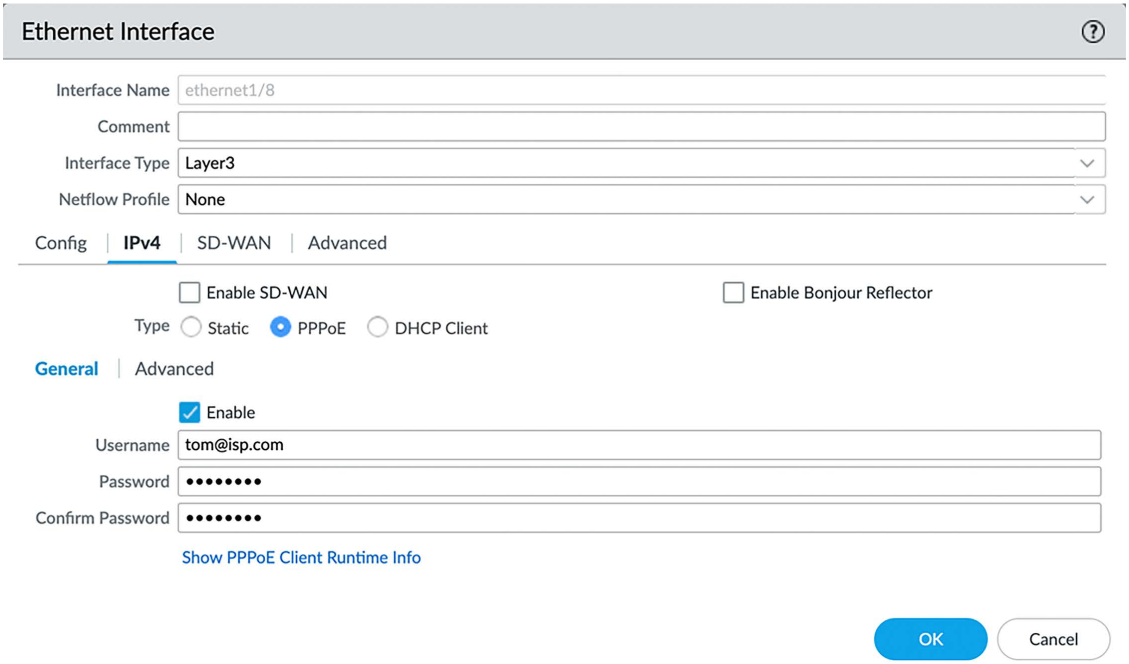


Figure 2.52: Layer 3 PPPoE

In the **PPPoE** | **Advanced** tab, you set the authentication protocol to **PAP**, **CHAP**, **auto**, or **none**. If the ISP has provided you with a static IP, you can configure it here and you can add an access concentrator and service string if the ISP requires them to be able to connect. If required, you can disable adding the default route received by the ISP to the routing table. Some ISPs require PPPoE clients to be in a passive state as they initiate the connection. You can enable this here:

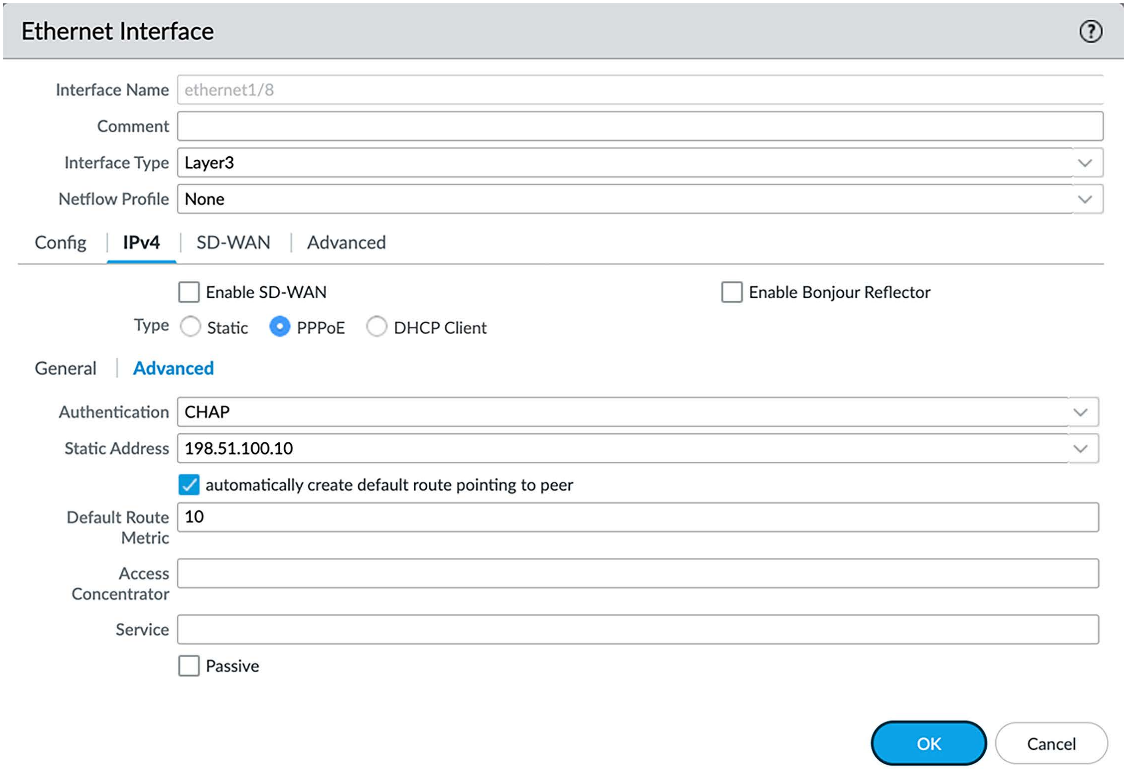


Figure 2.53: Layer 3 PPPoE Advanced options

Once you've configured the interface, click **OK** and commit the change. Once the commit has been completed, you can click on **Show PPPoE Client Runtime Info** from the **Network** | **Interfaces** | **Ethernet** page to receive information on the connection. From the CLI, you can issue the following command to see the same output:

```
admin@PA-440> show pppoe interface <interface>
```

The **DHCP Client** option allows you to configure the interface as a DHCP client, receiving its IP information from an upstream DHCP server:

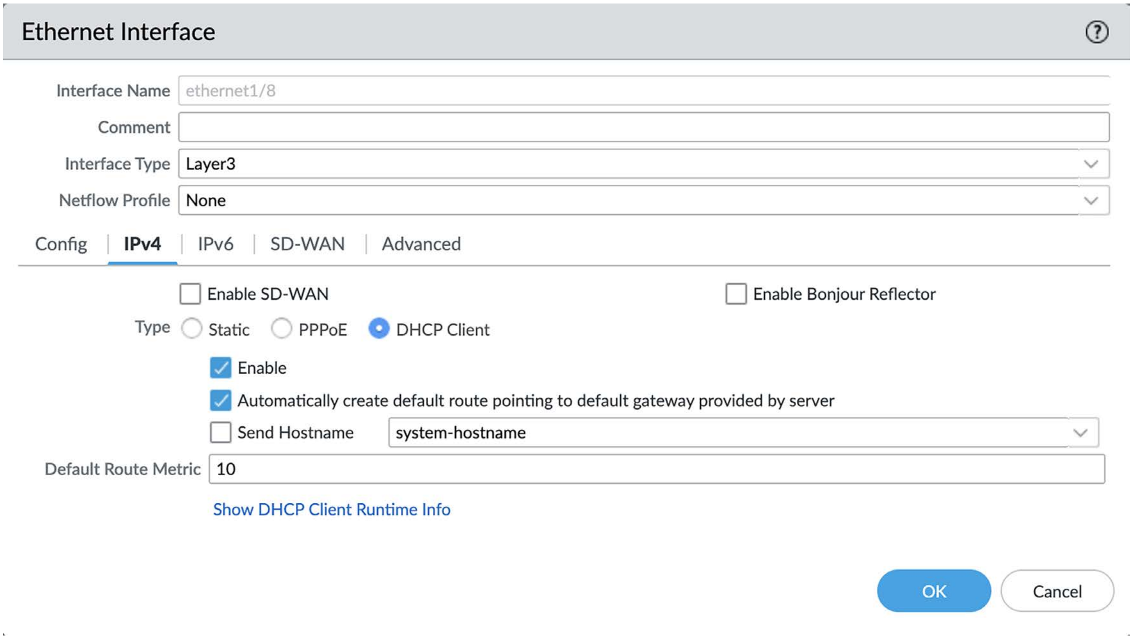


Figure 2.54: Layer 3 DHCP Client advanced options

You can choose to automatically accept the default route or disable this functionality and manually define the default route in the VR. If needed by the ISP, you can send the system or custom hostname, and you can determine the metric for the default route.

In the **Layer3** | **Advanced** tab, we gain access to several advanced features:

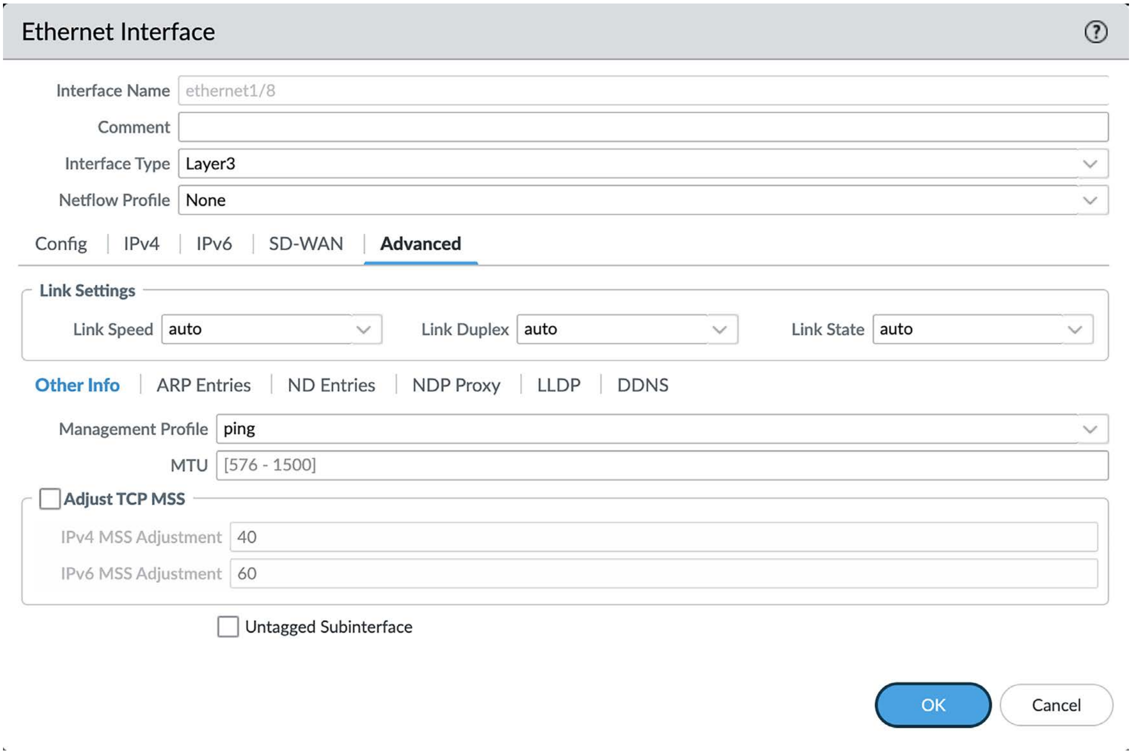


Figure 2.55: Layer 3 advanced options

Here we can set the link speed, duplex, and state of the interface. In the **Other Info** tab, we can add the management profile we saw earlier, change the **maximum transmission units** (**MTUs**), and enable **Adjust TCP MSS** with custom values to change the MSS header value in TCP packets flowing through the firewall.

In the other tabs available, we can also set static IPV4 ARP Entries, which can be useful in case an upstream device requires **a static ARP entry**, IPV6 **Neighbor Discovery** (**ND**) **Entries**, or **Neighbor Discovery Protocol** (**NDP**) **Proxy**. A **Link Layer Discovery Protocol** (**LLDP**) profile can also be added to interface with switches.

Lastly, if the external interface has a dynamic IP assigned to it by the ISP, a DDNS hostname can be set up. Currently, four vendors are supported: DuckDNS, afraid.org, no-IP, and Palo Alto:

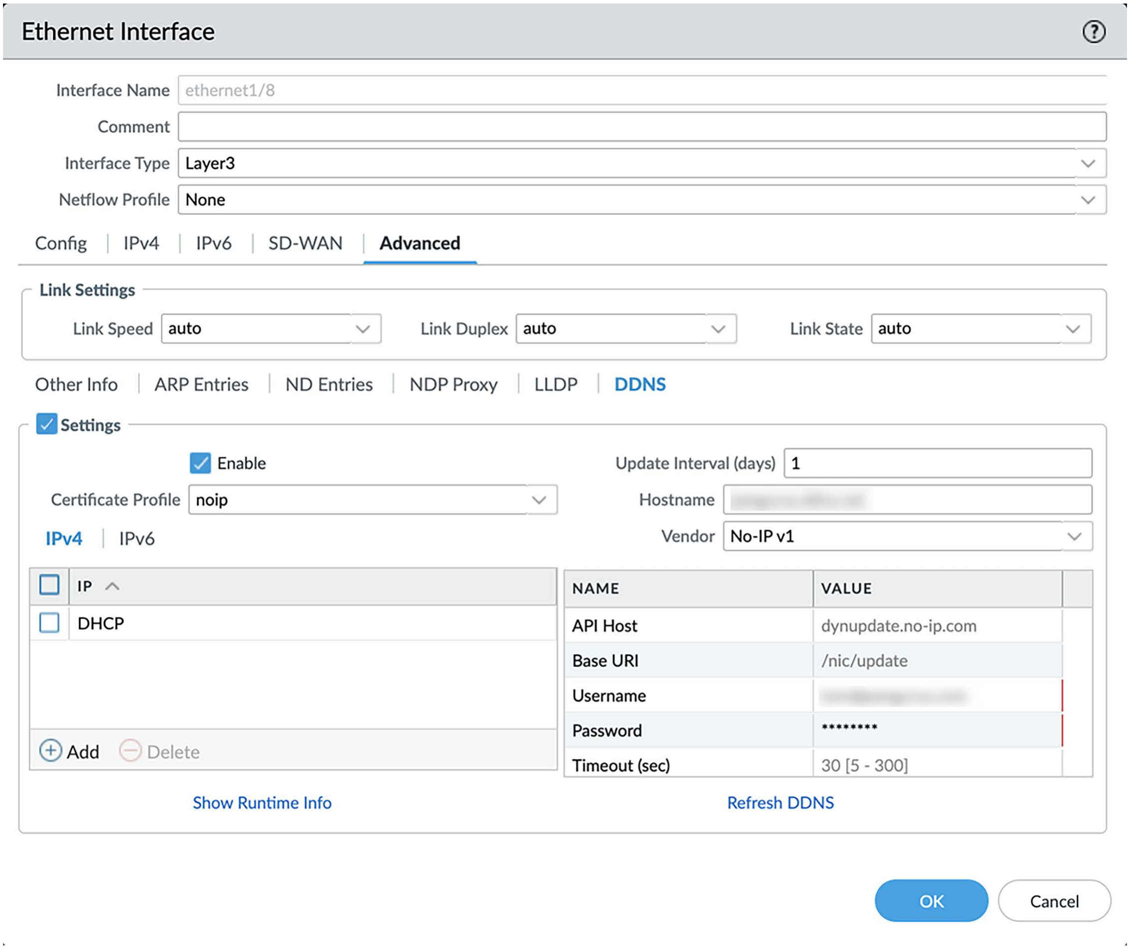


Figure 2.56: DDNS configuration

For the Layer 3 subnets and IP addresses to be reachable across interfaces, they need to be added to a routing table; this is accomplished in the VR.

VR

A VR is the routing element of the firewall, but, as the name suggests, it is not made up of a single engine but rather, a routing set that an interface is subscribed to. Each Layer 3, loopback, and VLAN interface needs to be associated with a VR, but multiple VRs can be used on a system. Not all interfaces need to be associated with the same VR.

You can configure the default VR or add new VRs from the **Network** | **Virtual Routers** menu. In the **Router Settings** tab of a VR, you can see and add interfaces associated with this VR and adjust the administrative distances if needed. An administrative distance associates a priority with a routing protocol. By default, static routes have a higher priority (lower administrative distance) than **Open Shortest Path First** (**OSPF**), but you can change this priority if you want OSPF routes to have priority and only use static routes if OSPF becomes unavailable. Routes within the same routing protocol can be assigned a metric to give them a higher (lower metric) or lower (higher metric) priority. Routes inside the same type (static, OSPF, etc.) are first prioritized based on the size of their subnet. A smaller subnet (for example, `/32`) will have priority over a larger subnet (for example, `/16`). Next, the metric will be used to determine priority for networks with the same subnet mask with the lower metric having priority (for example, `10.0.0.0/24 metric 10` will have priority over `10.0.0.0/24 metric 20`):

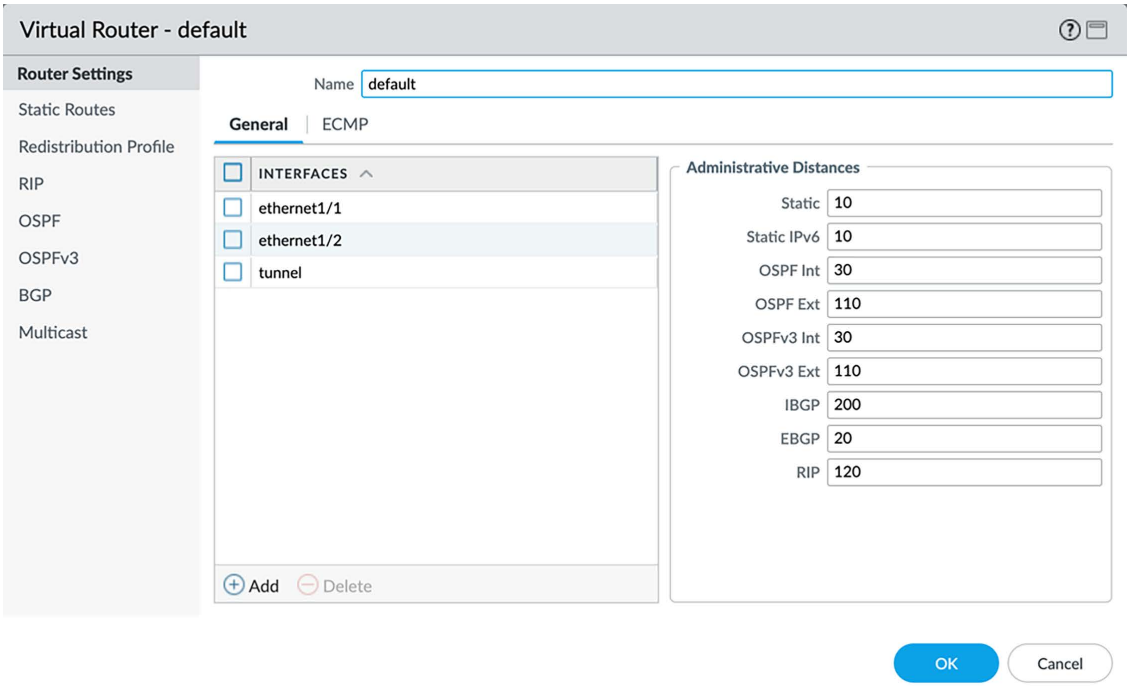


Figure 2.57: VR settings

In the **Static Routes** tab, you can add destination routes as needed. By default, the firewall loads all the connected (configured on a Layer 3, loopback, or VLAN interface) networks in the routing table; adding static routes makes remote networks available from a routing perspective.

One of the first routes you may need to configure is the default route, which allows clients to connect to the internet. If a Layer 3 interface is configured as a PPPoE or DHCP Client, an **Automatically Create Default Route** option can be enabled, which will automatically add the default route based on the upstream ISP.

The **Destination** for the default route is `0.0.0.0/0`. A regular route could have a smaller subnet, such as `172.16.0.0/24`.

The **Interface** option indicates what the egress interface will be. If the route is pointing to the internet, the interface will be the one where the ISP router is connected.

Next Hop has several options:

- **IP Address:** The IP of the upstream router to forward packets to.
- **Next VR:** Whether the packet needs to be handed over to a different VR on the same device.
- **FQDN:** If the upstream router has a dynamic IP, it could be useful to use an FQDN that is dynamically updated by a DNS record.
- **Discard:** Routes can be set to “black hole” certain subnets. This can be used to prevent any packets from reaching a connected out-of-band network, even if a security policy were to allow this.
- **None:** Routes may not have a next hop, such as packets routed into a VPN tunnel.

The **Admin Distance** and **Metric** settings can be changed for each route if necessary.

Route Table is used to add routes to regular unicast routing, multicast routing, or both.

You can, if you have redundancy available, use **Path Monitoring** to send a heartbeat ping over the route. If the ping fails a configured number of times, the route will be disabled. The routing table will be re-evaluated for matching packets and the next best match will be used to route packets (that is, a route with a higher metric or larger subnet):

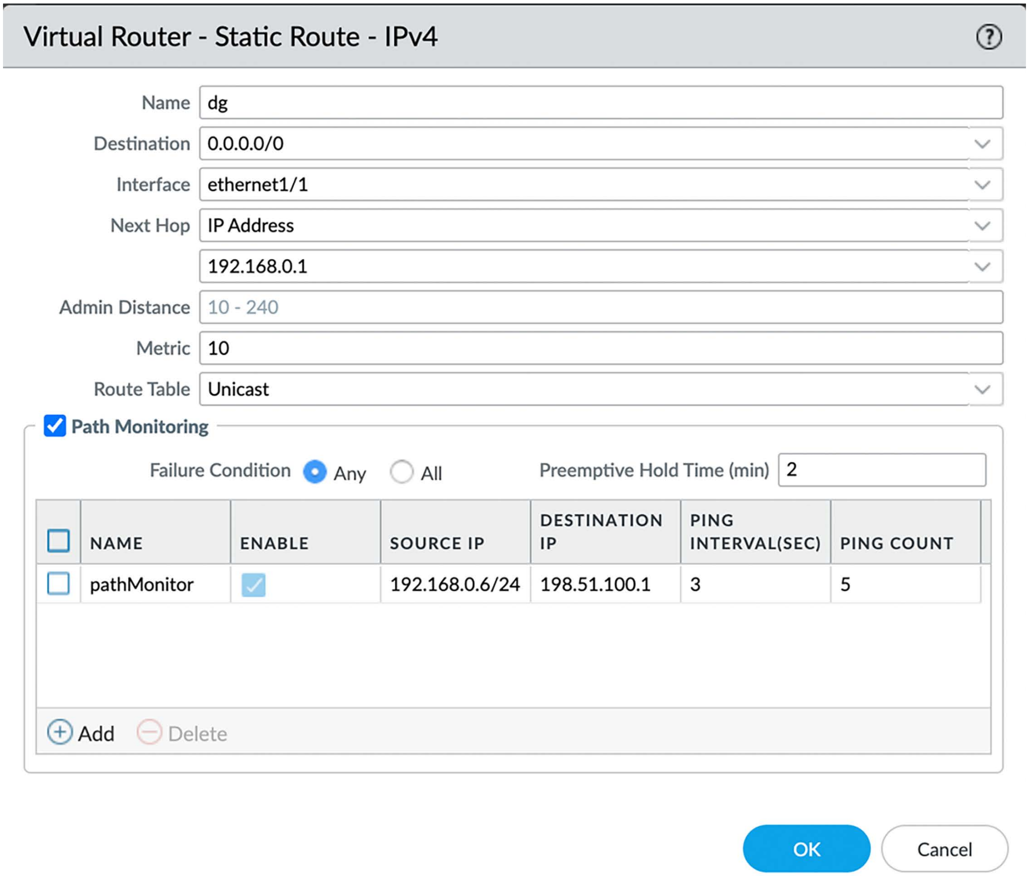


Figure 2.58: VR default route

Any subnets that are configured on a Layer 3 interface are added to the routing table as a connected network and do not need a static route to be added.

The Layer 2 interface and VLANs

Setting interfaces to the Layer 2 type enables the firewall to function in a similar way to placing a switch in the network. Each interface acts as the equivalent of an access port (if you need trunk functionality, refer to the *Subinterfaces* topic) on a switch, and you can add as many interfaces as you need.

Each interface should use a different zone so that a security policy can be leveraged to control traffic between the interfaces. Interfaces set to the same zone will, by default, exchange traffic without inspection and require a catch-all security policy to enable inspection.

To group the interfaces into a logical "switch," you need to create a VLAN object by going to **Networks | VLANs** and adding the interfaces you previously set to Layer 2 and want to be connected:

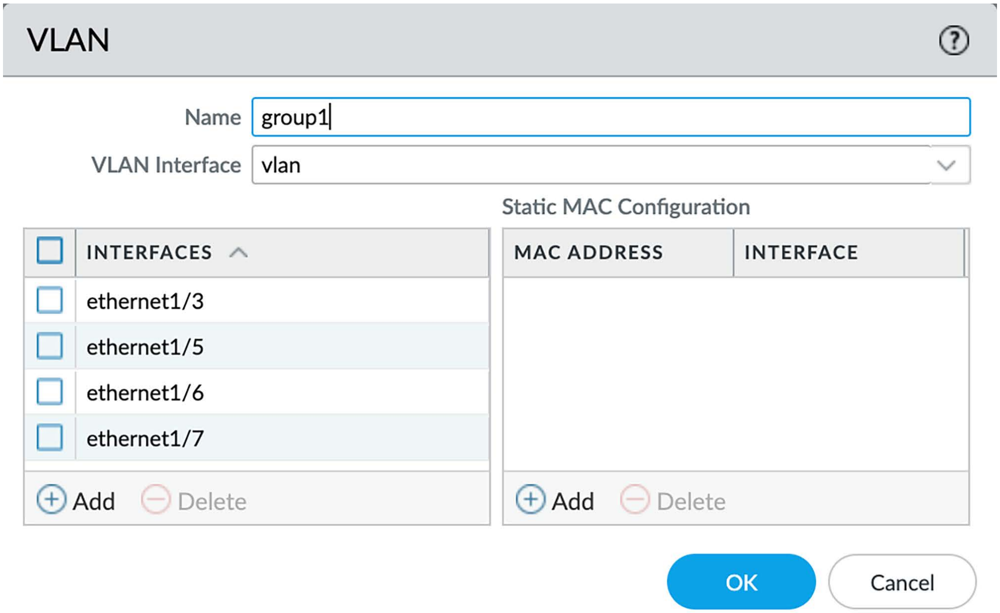


Figure 2.59: VLAN group

The **VLAN Interface** option adds routing functionality to the group as a logical Layer 3 interface. This can be useful if you have an upstream ISP router or a different subnet connected to a Layer 3 interface that you need to interact with.

You can configure the VLAN interface by going to **Network | Interfaces | VLAN**. Assign it to the VLAN group you created, fill in the **Virtual Router** field, and assign it a zone. This zone will represent Layer 2 interfaces when interacting with Layer 3 interfaces for security policies:

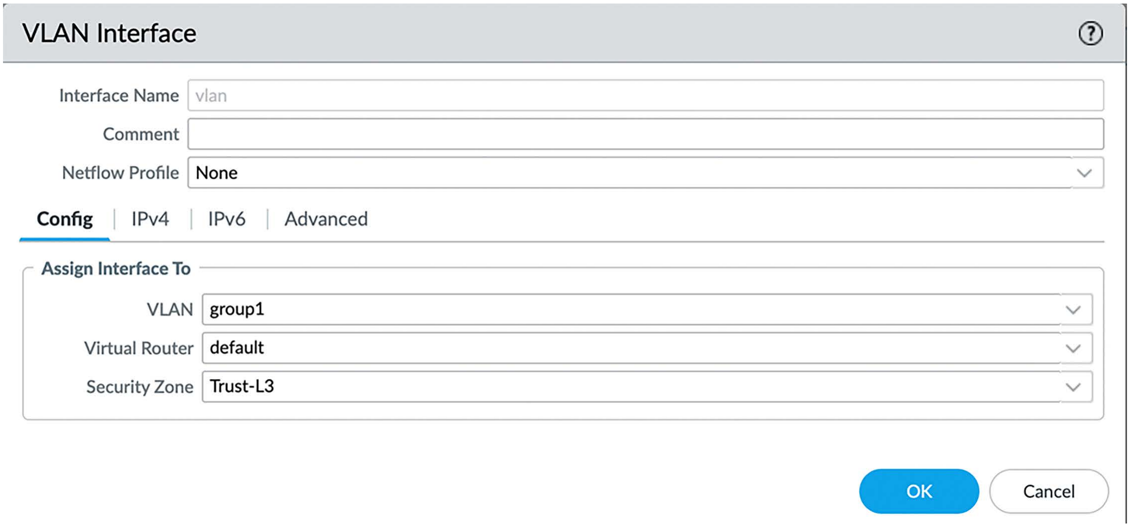


Figure 2.60: VLAN Interface configuration

You will also need to assign the VLAN interface an IP address that the clients on Layer 2 interfaces can use as a default gateway or routing next hop. Make sure it is in the same subnet as your clients on the Layer 2 interfaces:

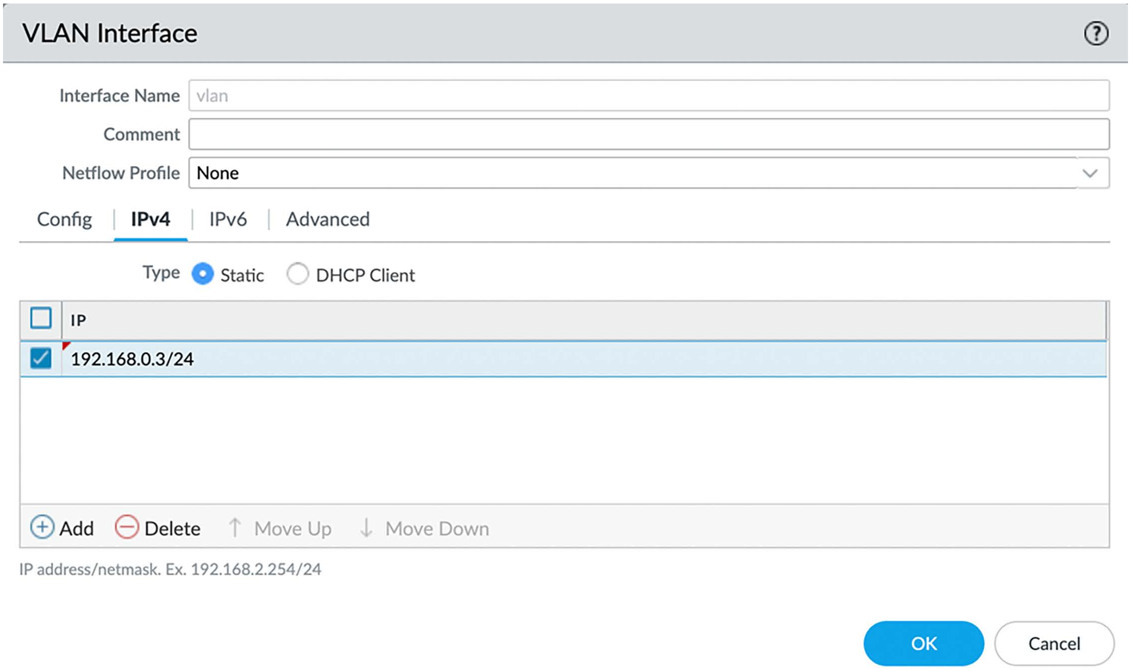


Figure 2.61: VLAN Interface IP address

Tap interfaces

Tap interfaces can be used as a passive sniffing port. If a different network device is set up with port mirroring, its egress port can be connected to the tap interface to intercept all packets and apply the app ID and content ID. As long as the tap interface is sent all packets of a session, it will be able to inspect the traffic as if it is flowing through the firewall. There are, however, a few limitations:

- As the firewall is not actively participating in the processing of packets, it cannot take action if it detects a threat; it can only report it.
- SSL decryption can only be applied to inbound connections if the server certificate can be loaded onto the firewall with its private key.

The tap interface only needs to be configured with a security zone:

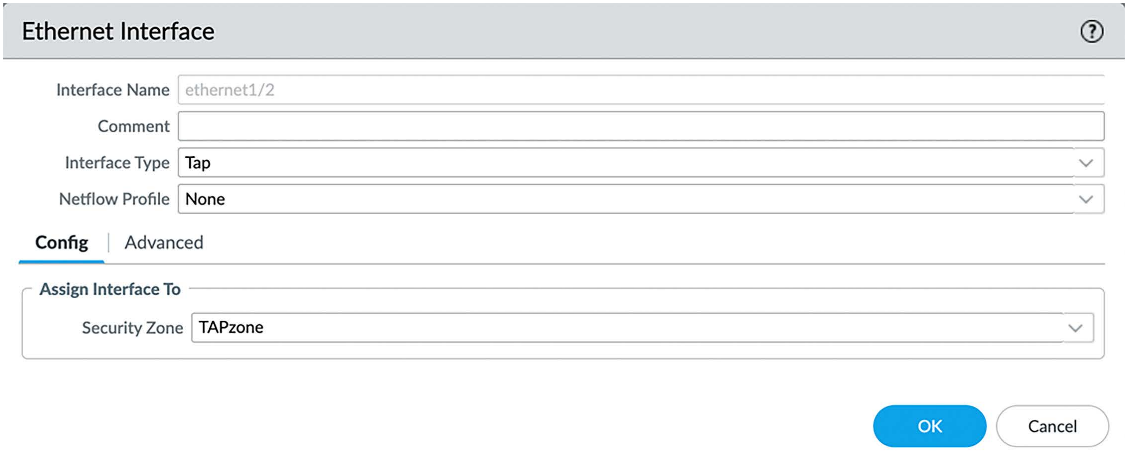


Figure 2.62: The tap interface

To optimally benefit from the tap functionality, a security rule will need to be created that allows all operations, or a specific subset, if you want to limit the scope. The firewall will discard all packets in the background, but setting the security rule to drop would discard the packets before inspection:

			Source		Destination						
	NAME	TYPE	ZONE	ADDRESS	ZONE	ADDRESS	APPLICATION	SERVICE	ACTION	PROFILE	OPTIONS
1	TAP-inspect	universal	TAP...	any	TAPz...	any	any	application-default	Allow		

Figure 2.63: The TAP security rule

Similar to listening in on a port mirror, the firewall can send all unencrypted session data to a third-party **DLP** or threat intelligence device. It can do so via a Decryption Port Mirror interface.

The Decryption Port Mirror interface

The Decryption Port Mirror interface allows the forwarding of decrypted packets to an external device for further inspection: the firewall will receive a TLS packet that is set to be decrypted, so takes all the steps to decrypt the session, and then forwards the clear text payload to a third party device for a "second opinion" using the Decryption Port Mirror interface as an egress interface. This can be useful for DLP, for example. The license can be activated for free via the support portal by browsing `https://support.paloaltonetworks.com` and then going to **Assets** | **Devices**.

There, you can find your firewall and click the **Activate License** button. If you choose to activate a feature license, you will be able to activate **Decryption Port Mirror**:

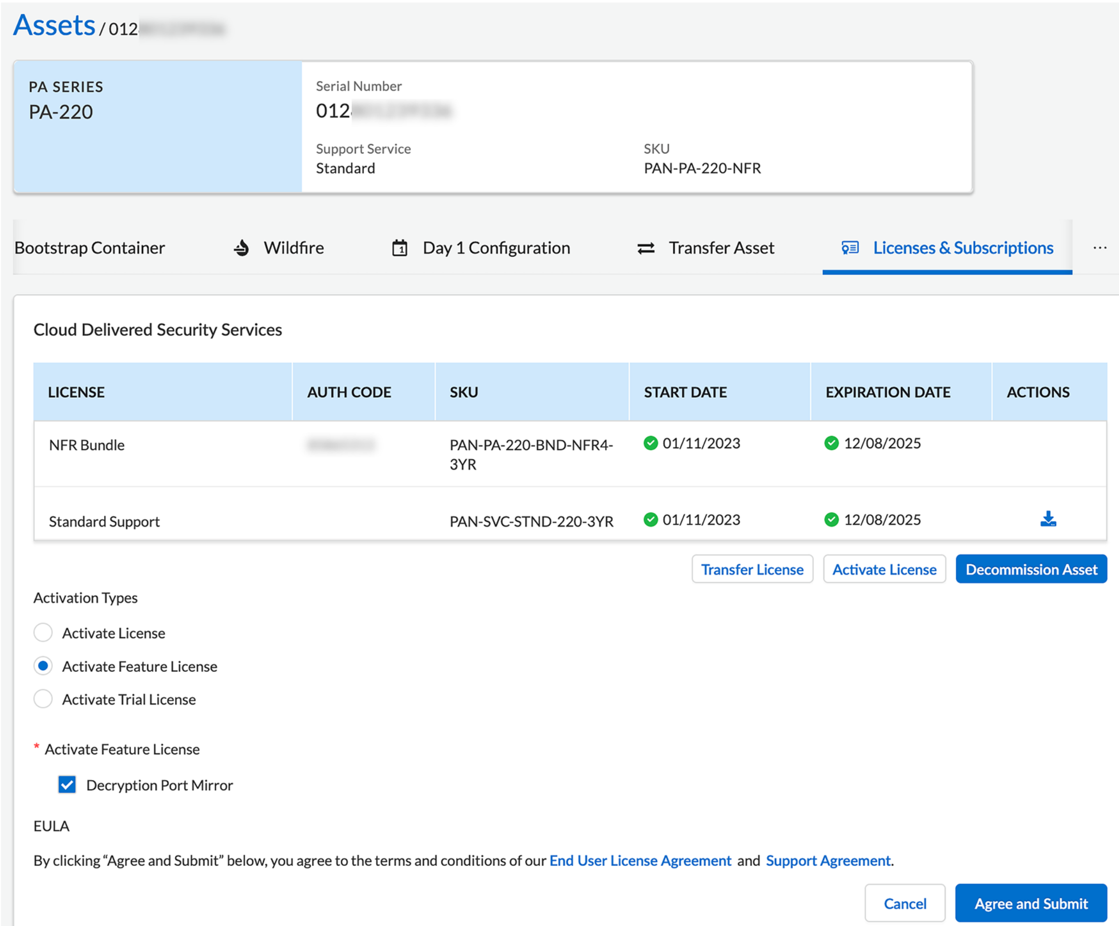


Figure 2.64: Activating a Decryption Port Mirror license

To activate the license on the firewall, follow these steps:

1. From **Device** | **Licenses**, select **Retrieve license keys from license server**.
2. In **Device** | **Setup** | **Content ID** | **Content-ID settings**, enable **Allow forwarding of decrypted content**.

3. In **Network** | **Interfaces** | **Ethernet**, set an interface to the **Decrypt Mirror** type.
4. In **Objects** | **Decryption** | **Decryption Profiles**, open the decryption profile and add the interface to **Decryption Mirroring**.
5. In **Policies** | **Decryption**, create decryption rules that use the decryption profile.
6. Save the changes and connect the Decryption Port Mirror interface.

Besides Ethernet interfaces, there are also three different logical interfaces:

- Loopback
- Tunnel
- VLAN interface

Let's now take a look at the Swiss army knife of interfaces, the loopback.

The loopback interface

A loopback interface is a logical Layer 3 interface that can serve many purposes. One common use case includes adding an additional public IP to its own interface so VPN configuration can be added to it. Another use case is to add a management profile to a loopback, and then leverage security rules to allow administrators to manage the firewall from exotic networks. It needs to be configured with an IP address (only a single IP per loopback interface is supported) and a security zone and it needs to be associated with a VR.

It can be set to a new IP address in the same subnet and zone as one of the Layer 3 interfaces, so services such as **Management Profile**, **Captive Portal**, and **GlobalProtect** can be hosted on a different IP than the main IP of the physical interface.

To add extra security, it can also be set to a different zone so that a matching security rule is needed for clients to be able to connect to the loopback interface:

Figure 2.65: Loopback Interface

The number next to **Interface Name** is an identification number for the logical interface. A number from 1 to 9999 can be assigned to the interface and will only serve as an identification number for the interface.

The tunnel interface

Tunnel interfaces are logical interfaces that serve as the ingress and egress point of tunneled traffic, both site-to-site VPN and GlobalProtect SSL and IPSec. The physical tunnel is terminated on a Layer 3 or loopback interface, but the packets that need to be encrypted should be routed to the tunnel interface:

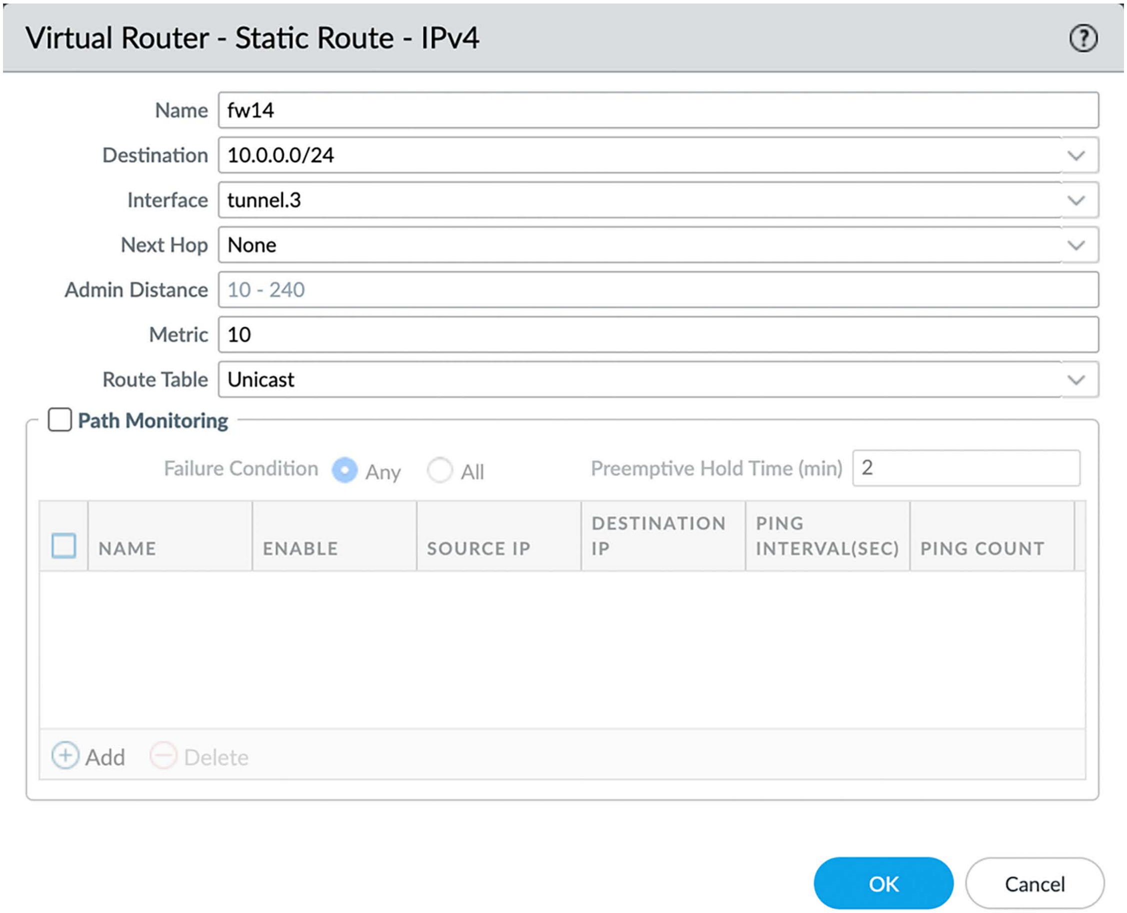


Figure 2.66: Static route for a VPN tunnel

This interface needs to be associated with a VR and a security zone, as you can see in the following screenshot:

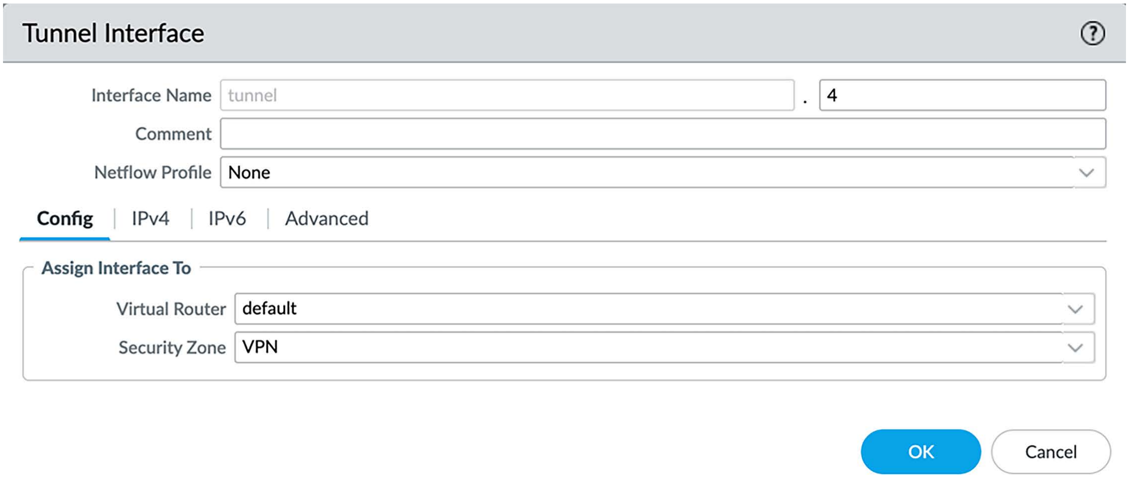


Figure 2.67: Tunnel Interface

The number next to **Interface Name** is an identification number for the logical interface.

Important note

For a strong security posture, set a separate zone for individual VPN connections, even for known locations. Treating each connection and remote network as an individual zone ensures adequate visibility and control. A remote office could be exposed to malware (think WannaCry) and infect other offices if the VPN tunnel is set to the same zone for all remote offices. The default `intrazone` security rule allows all sessions to run and does not apply scanning.

There are also several "special" interface types that provide a specific functionality; we'll cover the special use case interfaces in the following sections.

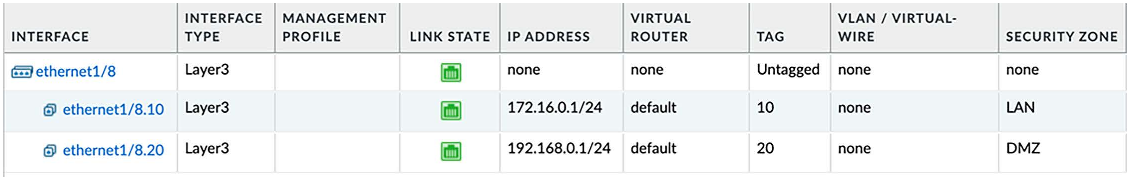
When a switch uplink needs to contain multiple 802.1q VLAN tags, it can be configured as a trunk, and, on the firewall, subinterfaces can be created to correspond to each VLAN tag.

Subinterfaces

All physical (that is, Layer 2, Layer 3, VWire, and Aggregate—we'll cover this type of interface a little further in this chapter) interfaces can have subinterfaces. A subinterface is used when the physical interface is connected to a trunked link containing VLAN (802.1Q) tagged packets. The physical interface is not able to interpret the tags, but subinterfaces are. For each VLAN carried by the trunk, you can create a subinterface to represent the virtual network coming from the switch. The advantage of using subinterfaces is that each VLAN can be associated with its own security zone.

The subinterface will mimic all the configuration specifics of its parent physical interface, but interface types cannot be different from the physical interface type (for example, a Layer 3 physical interface cannot host a Layer 2 subinterface).

You can create these by selecting the desired physical interface and clicking on **Add Subinterface** at the bottom left of **Network | Interfaces:**



INTERFACE	INTERFACE TYPE	MANAGEMENT PROFILE	LINK STATE	IP ADDRESS	VIRTUAL ROUTER	TAG	VLAN / VIRTUAL-WIRE	SECURITY ZONE
ethernet1/8	Layer3			none	none	Untagged	none	none
ethernet1/8.10	Layer3			172.16.0.1/24	default	10	none	LAN
ethernet1/8.20	Layer3			192.168.0.1/24	default	20	none	DMZ

Figure 2.68: Creating a subinterface

HA interfaces

High Availability (**HA**) interfaces are required when setting up a cluster of two firewalls. Some chassis will have built-in dedicated HA interfaces, in which case you may not need to create any HA interfaces yourself. If no onboard HA interfaces are available, or additional interfaces are required to serve as backup HA links, data plane interfaces can be selected to fulfill this role and are connected to the HA peer.

AE interfaces

To increase available bandwidth above the physical limitations of the interfaces, interfaces can be bundled into an **Aggregated Ethernet** (**AE**) group using the 802.1AX protocol. Up to eight interfaces can be combined into a logical bundle.

A new group can be created by clicking on **Add Aggregate Group** under **Network** | **Interfaces** | **Ethernet.**

You first need to set the type to Layer 2, Layer 3, VWire, or HA, which will require the same configuration as the physical interface equivalent (that is, security zone, VR, VLAN, or VWire).

Additionally, you can configure the **Link Aggregation Control Protocol** (**LACP**) to improve interface failure detection. LACP enables link failure detection on the physical and data link layer, while the default protocol only detects physical link failure.

You can set whether the firewall is in **Active** or **Passive** mode. This configuration setting needs to be reviewed with the LACP peer (typically the switch) as only one peer can be configured as **Active**, but LACP will not work if both are set to **Passive.**

The value of **Transmission Rate** will have an impact on the responsiveness of link failure detection, but it will also have an overhead. Slow transmission means every 30 seconds, while fast transmission means every second.

Fast Failover will fail to an operational interface within 1 second when an interface goes down. Traditional failover happens after 3 seconds.

System Priority determines which peer determines port priorities.

Maximum Interfaces determines how many interfaces can be active at the same time within the aggregate group. This number should not exceed the number of physical interfaces you assign to the group, but can be leveraged to limit total available bandwidth while keeping hot interfaces in reserve in case of failure. For example, if a total bandwidth of 4 gigabits is needed for an aggregate group but you also do not want to exceed this bandwidth to preserve system resources, you can assign five or more interfaces to the aggregate group, and set **Maximum Interfaces** to 4. Only when an interface fails will another one be activated to pick up the work. In an HA configuration where two firewalls form a cluster, LACP can be enabled on the passive peer so the link aggregation group is pre-negotiated before the passive peer needs to assume an active role, which cuts down on the time needed to failover.

This is achieved by checking **Enable in HA Passive State**. The same system MAC can be used on both cluster members, but this may not be supported by the connected switches. This is shown in the following screenshot:

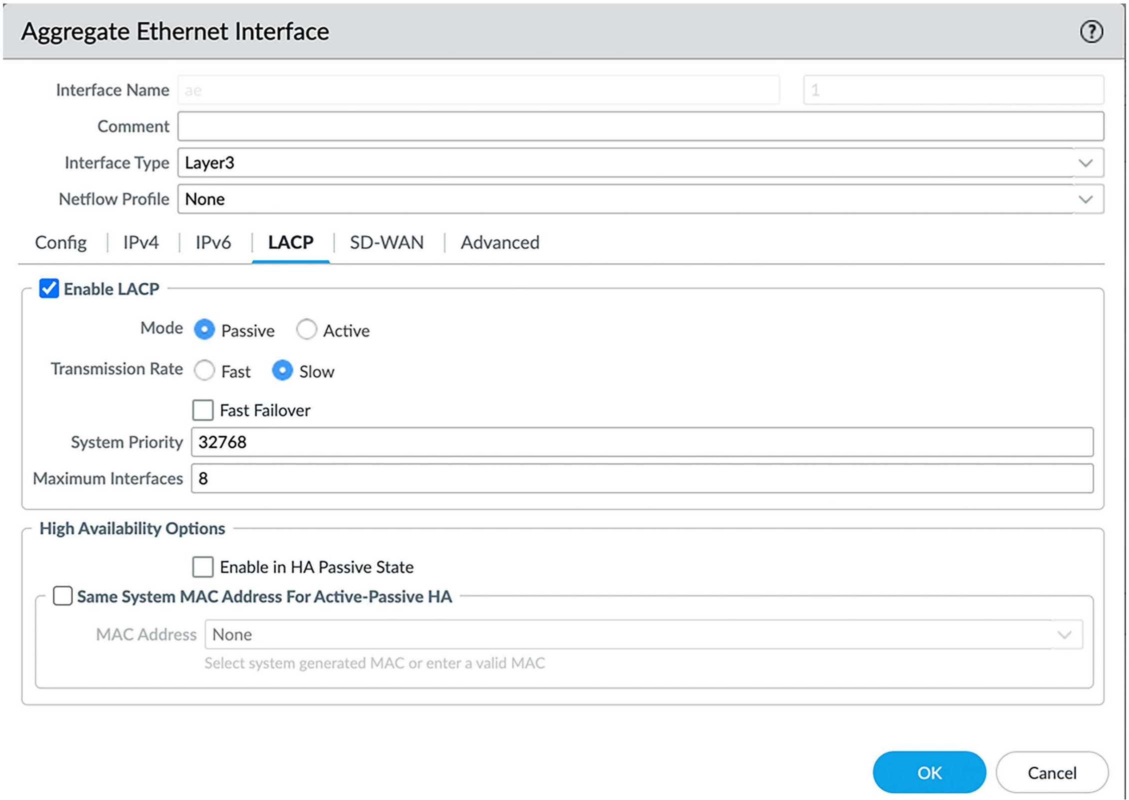


Figure 2.69: LACP

When the aggregate group is created, you can add the interfaces by setting **Interface Type** to **Aggregate Ethernet** and selecting the desired **Aggregate Group**:

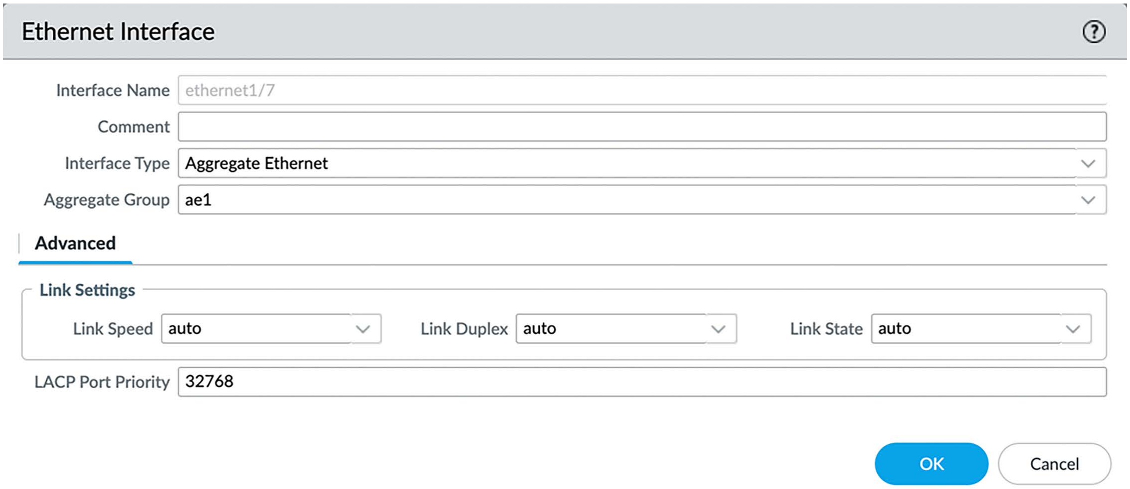


Figure 2.70: A physical interface in an aggregate group

In some cases, you may need to be able to connect to a port mirror on a switch and just listen without participating. For such instances, you can configure a tap interface.

With the information covered in the last sections, you are now able to select the appropriate interface for each network design you may come across. VWire helps you add a firewall in an environment where you can't interfere with existing routing, Layer 3 interfaces put the firewall in the middle of routing decisions, Layer 2 interfaces make the firewall act in a similar way a switch would, and subinterfaces can be added to all of these to account for VLAN tags. You are able to configure link aggregation and can leverage tunnel interfaces to set up IPSec tunnels.

Summary

In this chapter, you learned how to create a support account, register a new device, and add licenses. You are now able to identify all the different support licenses and select the appropriate subscription licenses to address your needs. You can upgrade and update a device so that its firmware is up to date and the latest application and threat signatures are loaded to protect the network. You learned how to protect the management interface so that only legitimate users can connect, and you can assign different accesses and privileges to administrators. You are able to configure all the physical interfaces, like Layer 3 and VWire, and know when each is most appropriate. You can also leverage logical interfaces like tunnel interfaces and loopback interfaces when they are needed.

If you're preparing for the PCNSE, you should take note that upgrading requires the base image to be downloaded before you can move forward to a maintenance release. The recommended threshold for dynamic updates is 6 to 12 hours (unless the device is located in a critical environment, where the threshold should be 24 hours) and you should be able to identify the difference between all the interface types.

In the next chapter, we will start building robust security policies and learn how to set a strong security posture for network traffic.

3

Building Strong Policies

In this chapter, you will get comfortable with configuring security profiles, building rule bases for security, and **network address translation** (**NAT**). We will learn what each setting does, what its expected behavior is, and how it can be leveraged to lead to the desired outcome. Taking full control over all of the features available in the different rule bases will enable you to adopt a strong security stance.

In this chapter, we're going to cover the following main topics:

- Understanding and preparing security profiles
- Understanding and building security rules
- Setting up NAT in all possible directions

By the end of this chapter, you will be able to set up a complete ruleset that will ensure your users are able to reach the applications and resources they need, internally hosted servers can be reached from the internet, and any threats trying to slip through can be stopped in their tracks.

Technical requirements

Before you get started, your firewall must have connectivity between at least two networks, with one preferably being your **internet service provider** (**ISP**), to fully benefit from the information provided in this chapter.

Understanding and preparing security profiles

There are several types of security profiles, like **Antivirus**, **Vulnerability Protection**, **URL Filtering**, and **Anti-Spyware** (we'll cover all of these different profiles in the following sections). Before you can start building a solid security rule base, you need to create at least one custom security profile of each type to use in all of your security rules. There are pre-configured profiles available, but they are set to read-only, which means that if you start using them in security rules, you'll need to replace them one by one if you create custom ones later.

Pro Tip

As a best practice, add security profiles to all of the rules, including drop rules. On drop rules, security profiles do not consume any resources but they will help form a habit of adding profiles to all rules, and if the rule ever changes to **Allow**, the profiles will already be assigned.

Security profiles are evaluated by the first security rule that a session is matched against. If a six-tuple is matched against a security rule with no or limited security profiles, no scanning can take place until there is an application shift and the security policy is re-evaluated. *All* security rules need to have security profiles.

The Antivirus profile

The Antivirus profile has three sections that depend on different licenses and dynamic update settings:

- The actions under **ACTION** rely on the threat prevention license and antivirus updates
- **Wildfire Action** relies on the WildFire license and the WildFire updates that are set to periodical updates (1-minute or longer intervals)
- **Wildfire Inline Ml Action** relies on WildFire set to real time

If any of these licenses are missing from your system, the actions listed in their columns will not be applied.

Application Exception allows you to change the action associated with a decoder for individual applications as needed. The actions that can be set for both threat prevention and WildFire antivirus actions are as follows:

- **allow**: Allows matching signatures *without* logging.
- **drop**: Drops matching signatures and writes an entry in the threat log.
- **alert**: Allows matching signatures to pass but writes an entry in the threat log.
- **reset-client**: Drops matching packets, sends a TCP RST to the client (session initiator), and writes an entry in the threat log. The server (session responder) is not notified.
- **reset-server**: Drops matching packets, sends a TCP RST to the server (session responder), and writes an entry in the threat log. The client (session initiator) is not notified.
- **reset-both**: Drops matching packets, sends a TCP RST to both the client and the server, and writes an entry in the threat log.

Packet captures can be enabled for further analysis by the security team or as forensic evidence. They are attached to the threat log and are limited to packets containing matched signatures.

You can create a new Antivirus profile by going to **Objects | Security Profiles | Antivirus**. As the following screenshot shows, we will use Palo Alto Networks' recommended best practices settings:

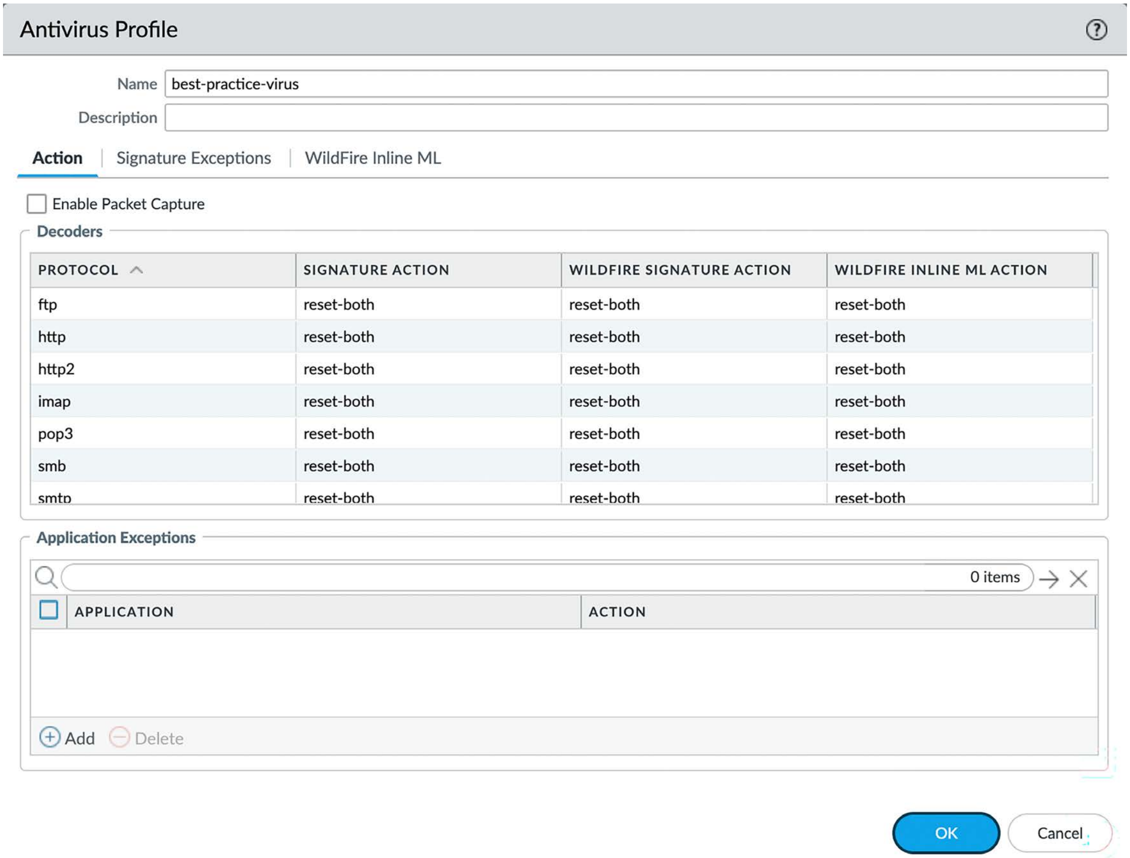


Figure 3.1: Creating an Antivirus profile

For the Antivirus profile, the setting in the default profile will have **imap** and **pop3** set to **alert** only. Since these protocols do not respond too well to reset, this may be a good setting in cases where they are actively being used in the organization. In most cases, email has been replaced by a web-based (TLS/SSL) alternative and it would actually be more secure to reset legacy protocols.

Exceptions can be added in the **Signature Exceptions** tab for false positives or known true positives. Exceptions are added by typing (or pasting) the **Threat ID** as found in the threat log and clicking **Add**:

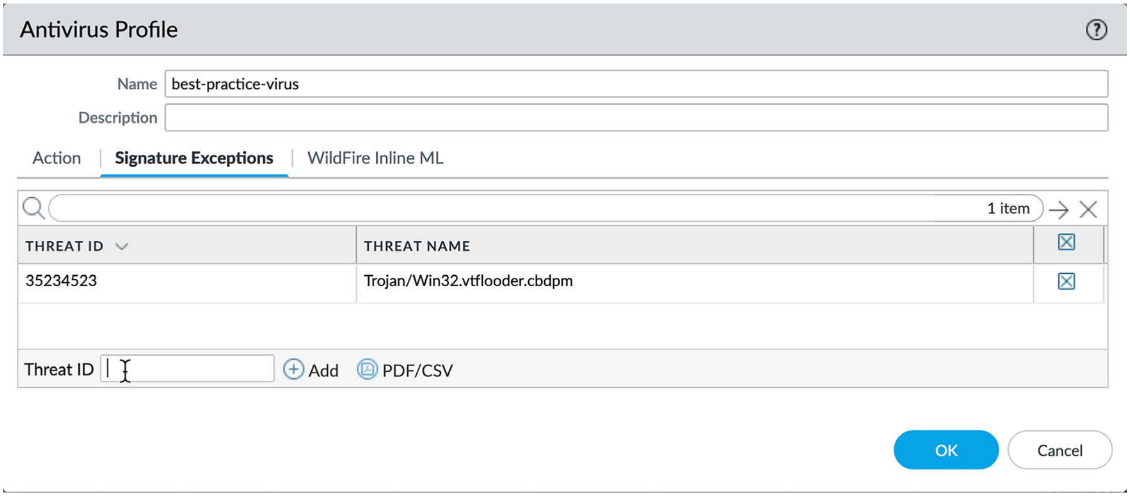


Figure 3.2: Antivirus exceptions

Additional machine learning actions can be set on the **WildFire Inline ML** tab for **Windows Executables**, **PowerShell Script 1** and **PowerShell Script 2**, **Executable Linked Format, MSOffice, Shell, OOXML,** and **MachO.** Each model can be set to one of three actions:

- **enable (inherit per-protocol actions)**, which enables additional machine learning scanning, and if a virus is detected, the matching protocol (**smtp**, **http**, and so on) action that was set in the main **Action** tab is applied
- **alert-only (override more strict actions to alert)** will enable additional machine learning scanning but positive matches will only be reported in logging
- **disable (for all protocols)**, which is the default setting and does not apply machine learning to the selected model

The following screenshot illustrates the available actions per model:

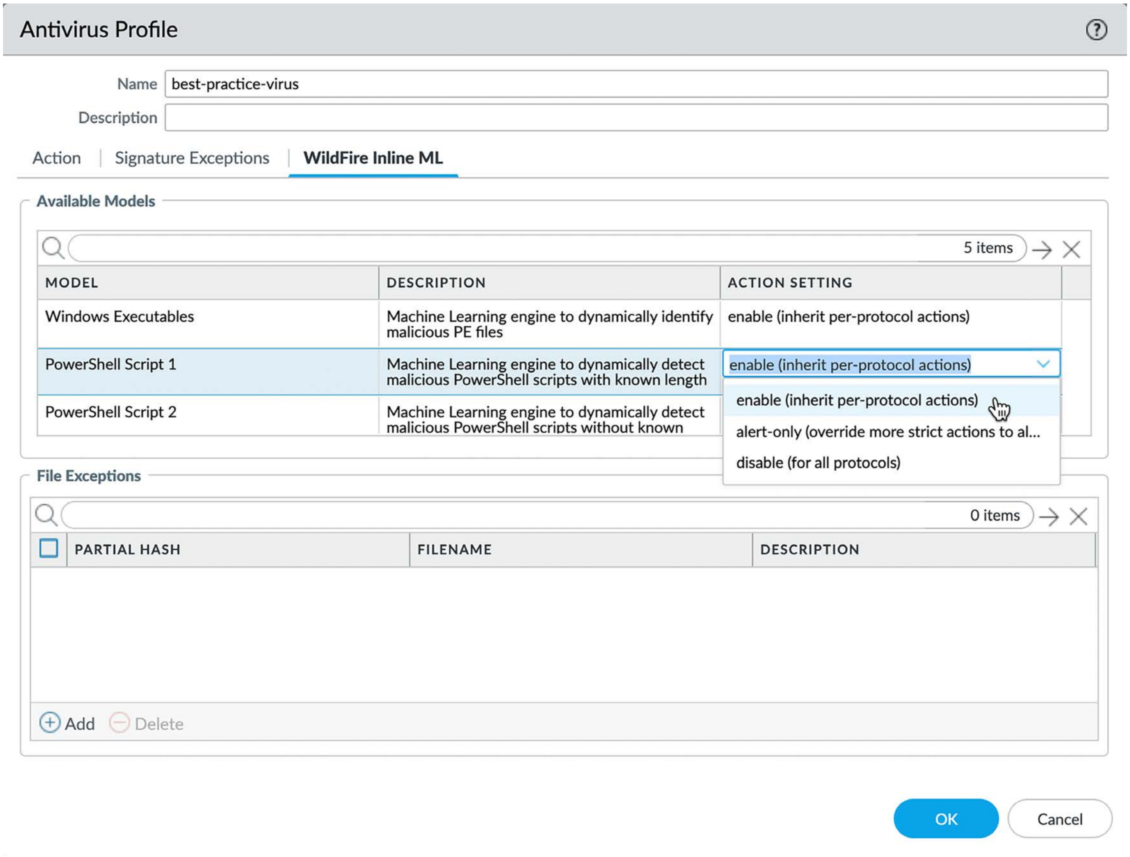


Figure 3.3: WildFire Inline ML

WildFire Inline ML can dynamically analyze certain file types for various file details and PowerShell scripts for malicious characteristics.

The Anti-Spyware profile

The Anti-Spyware profile is extremely customizable and is built by a set of rules within the profile. These rules serve to change the default actions associated with each threat; so, if no rules are created at all, the profile will simply apply the default action for a specific signature when it is detected.

Anti-Spyware supports the same actions as Antivirus (**allow**, **drop**, **alert**, **reset-client**, **reset-server**, and **reset-both**), with the addition of **block-ip**, which can track by source or source-destination pair and will block the offending IP for a duration of `1-3600` seconds. Tracking by source will block all connections from the client for the duration of the block, while tracking by source-destination pair will only block connections from the client to the target destination and will not block the same client from connecting to other destinations.

The **Packet Capture** options include:

- **none**.
- **single-packet**, which only captures the packet containing the payload matching a signature.
- **extended-capture**, which enables the capture of multiple packets to help analyze a threat. The number of packets captured by **extended-capture** can be configured via **Device** | **Setup** | **Content-ID**. The default is `5`.

> **Important note**
>
> Enabling packet capture on all threats does require some CPU cycles. The impact will not be very large, but if the system is already very taxed, some caution is advised.

Severity indicates the severity level of the threat that applies to this rule.

Create a new Anti-Spyware profile and add the following rules:

- **Policy Name**: `Block-Critical-High-Medium`
    - **Severity**: `critical`, `high`, `medium`
    - **Action**: `reset-both`
    - **Packet Capture**: `single-packet`
- **Policy Name**: `Default-Low-Info`
    - **Severity**: `low`, `informational`
    - **Action**: `default`
    - **Packet Capture**: `disable`

Important note

The official best practice profiles may look a little different than they have been laid out here. This is because Palo Alto splits up every severity (but with exactly the same settings). So, the preceding example is a little more efficient!

Your profile will now look like this:

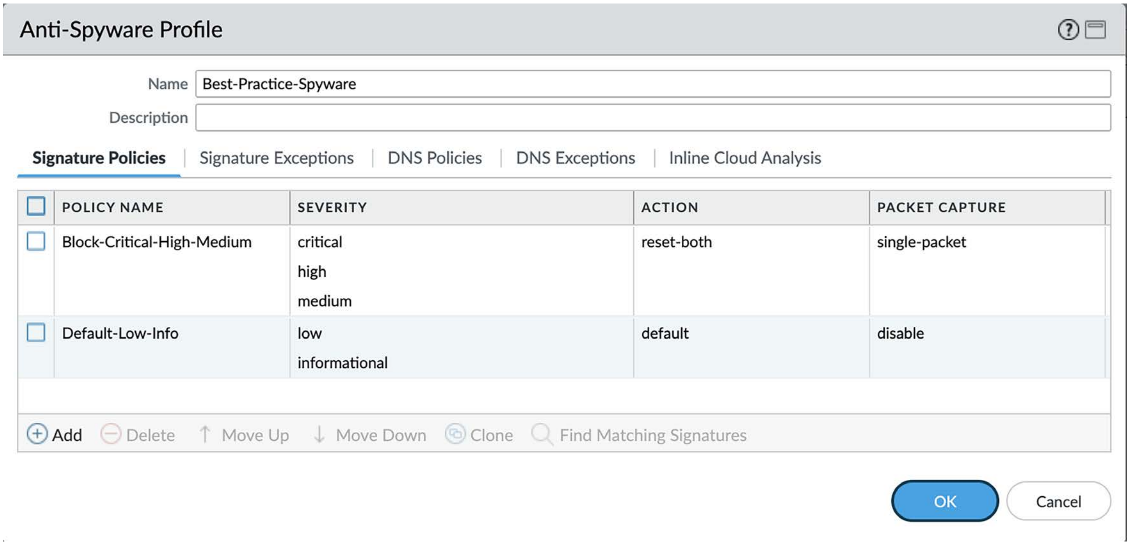


Figure 3.4: Anti-Spyware Profile

The **critical**, **high**, and **medium** severity threats have an overall high ratio of being genuinely malicious, so resetting the connection is considered best practice. Collecting a packet capture for these threats can help us find out whether something that was blocked is false. Low and informational severity threats are likely purely informational and can be left as the default action.

As you can see in the following screenshot, we need to make sure we review **Category** as this allows a fine-grained approach to each type of threat if granularity and individualized actions are needed at a later stage:

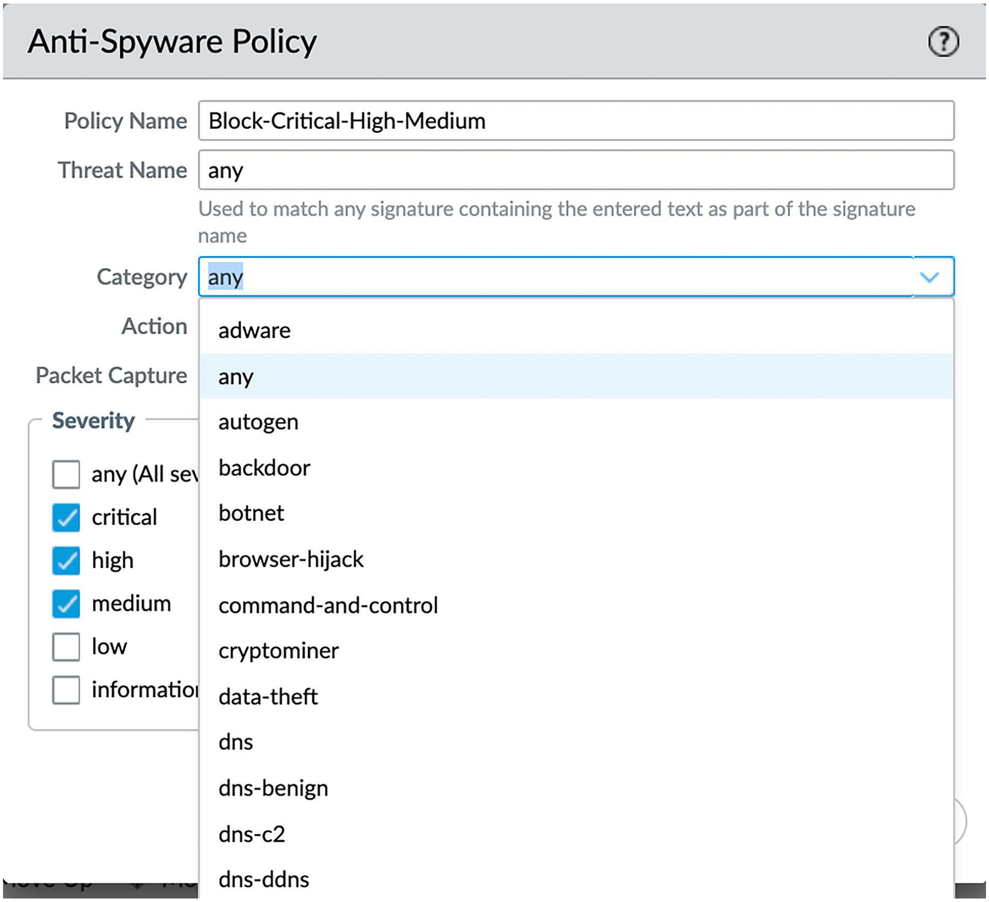


Figure 3.5: Anti-Spyware categories

The Anti-Spyware profile also contains DNS signatures, which are split into two databases for the subscription services. The content DNS signatures are downloaded with the threat prevention dynamic updates. The DNS Security database uses dynamic cloud lookups.

The elements in each database can be set to **Alert**, **Allow**, **Block**, or **Sinkhole**. **Sinkhole** uses a DNS poisoning technique that replaces the IP in the DNS reply packet, so the client does get a valid DNS reply, but with an altered destination IP. This ensures that infected endpoints can easily be found by filtering traffic logs for sessions going to the sinkhole IP. You can keep using the Palo Alto Networks default sinkhole, `sinkhole.paloaltonetworks.com`, or use your preferred IP.

The way that the DNS sinkhole works is illustrated by the following steps and diagram:

1. Client makes a DNS request for domain badwebsite.com from its preferred DNS server.
2. The internal DNS server receives the request and will in turn request the records for this domain from it's upstream servers.
3. The firewall sees the request and performs a lookup for the requested domain badwebsite.com and determines it is malware.
4. The upstream DNS server replies with the A record (IP address) of the domain badwebsite.com.
5. The firewall replaces the A record with the sinkhole IP address.
6. The internal DNS server forwards the "poisoned" reply to the client.
7. The client tries to establish a connection with the sinkhole IP instead of the actual IP address.

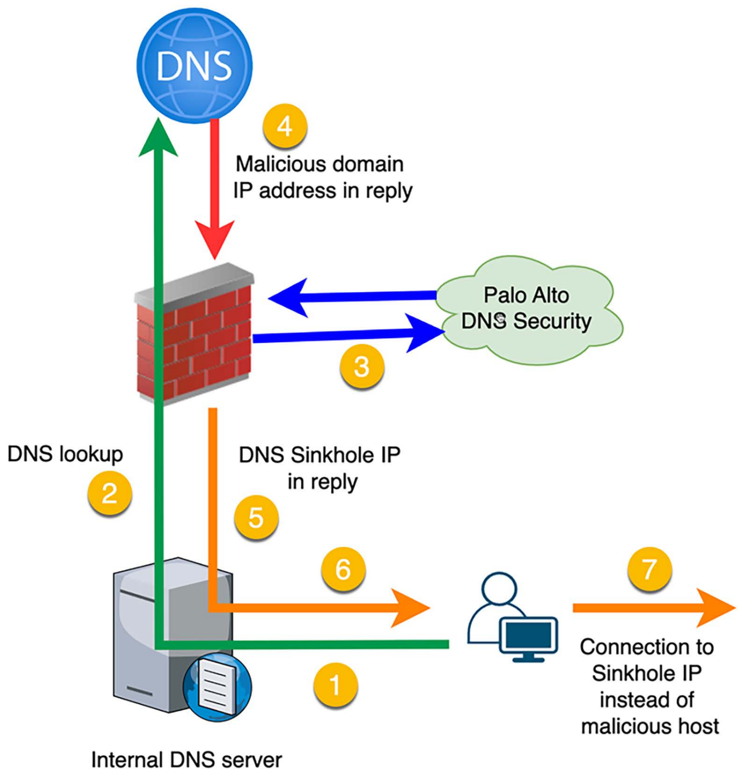


Figure 3.6: How a DNS sinkhole works

Blocking instead of sinkholing these DNS queries would implicate the internal DNS server as requests are relayed through it.

The default action for the **Command and Control** and **Malware** domains is to block the DNS reply, as sinkholing could trigger more evasive behavior. Therefore, it is recommended to leave these categories at their default settings. Select an appropriate action for all the other categories. For research purposes, you can enable packet capture.

In PAN-OS 11.2 and later, in the **DNS Zone Misconfiguration** field, you can add public-facing *parent* domains within your network. Advanced DNS security can monitor these domains for misconfigurations such as abandoned subdomains that could be abused by malicious actors.

At the bottom of this tab, we can select **SVCB** (type 64), **HTTPS** (type 65), and **ANY** (type 255), which are DNS resource record types used by encrypted DNS to encrypt the client hello during the DNS resolution process. Blocking these will prevent encrypted DNS sessions from being established, ensuring you can keep applying DNS security:

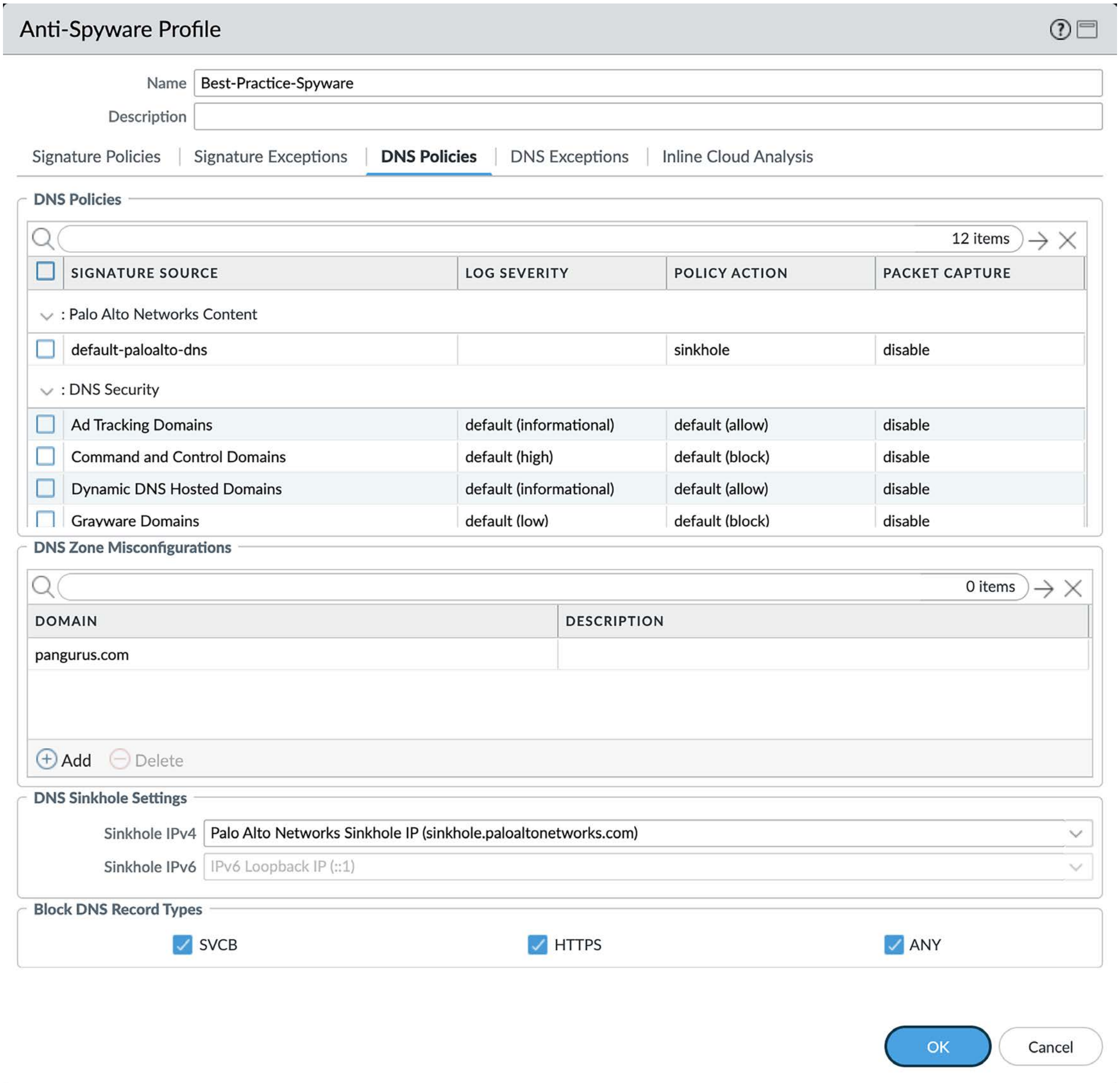


Figure 3.7: Anti-Spyware DNS signatures

Any false positives or trusted sites that somehow are malicious can be added to **DNS Exceptions** so they will no longer be blocked or sinkholed.

On the **Inline Cloud Analysis** tab, we can enable inline cloud analysis and define actions to be taken for each protocol where C2 traffic can be detected by local deep learning analysis engines. If certain domains or IP addresses should be excluded from this inspection, you can add an **external dynamic list** (**EDL**; we'll learn about these later) or IP in the fields shown in the following screenshot:

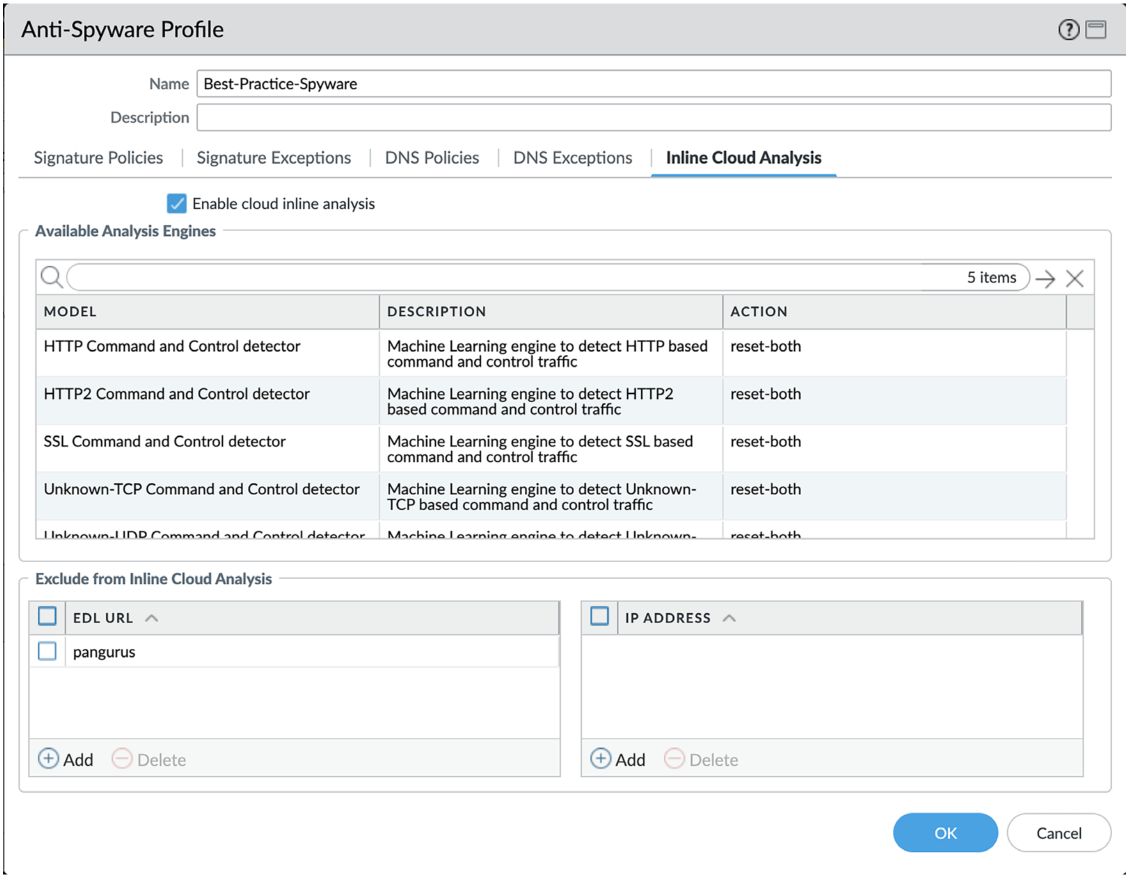


Figure 3.8: Inline Cloud Analysis

Let's now look at the Vulnerability Protection profile.

The Vulnerability Protection profile

The Vulnerability Protection profile also uses rules to control how certain network-based attacks are handled. **Action** contains the same options as Anti-Spyware: **allow**, **drop**, **alert**, **reset-client**, **reset-server**, **reset-both**, and **block-ip**.

The reset actions send TCP RST packets. **block-ip** blocks all packets coming from a source and can be set to **monitor source** to block everything, or a source destination to only block packets to a specific destination for an amount of time.

Host Type determines whether the rule applies to a threat directed at a **client** (download: for example, a browser vulnerability), **server** (upload: for example, SQL injection), or **any**.

Make sure to review the available categories because they, just like we saw in *Figure 3.4* in Anti-Spyware, can allow far more granularity if needed:

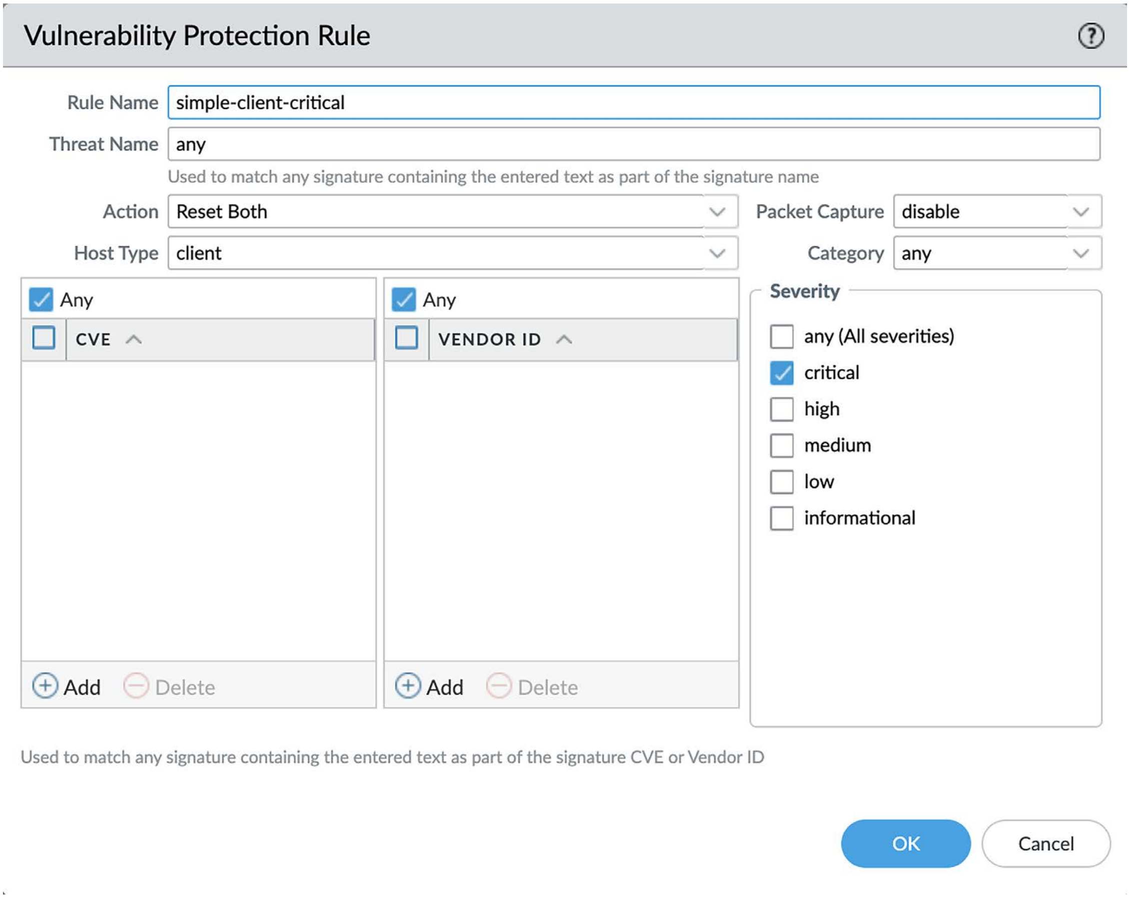


Figure 3.9: The Vulnerability Protection profile categories

Create the following rules:

- **Rule Name:** `Block-Critical-High-Medium`
    - **Host Type:** `any`
    - **Severity:** `critical`, `high`, `medium`
    - **Action:** `reset-both`
    - **Packet Capture:** `single-packet`

- **Rule Name:** `Default-low-info`
    - **Host Type:** `any`
    - **Severity:** `low`, `informational`
    - **Action:** `default`
    - **Packet Capture:** `disable`

Your profile should now look like this:

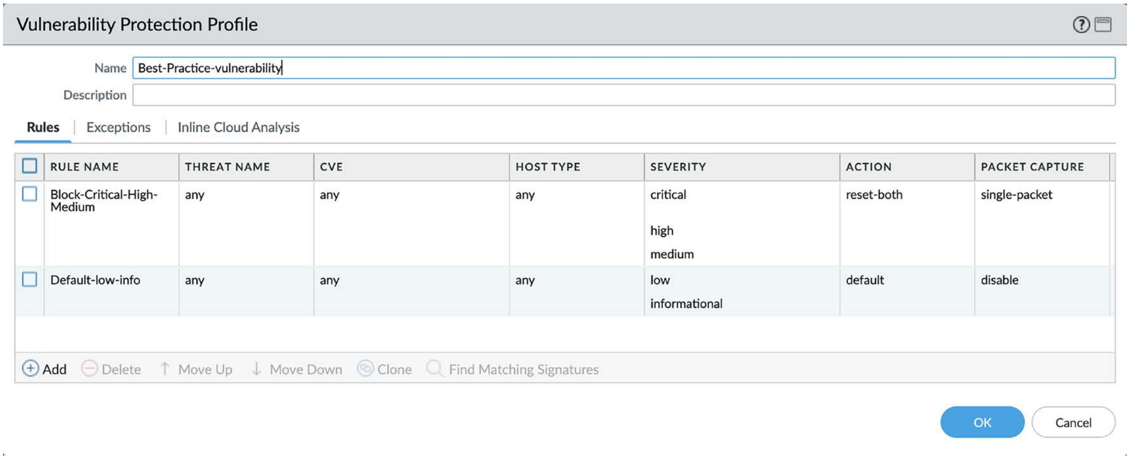


Figure 3.10: Vulnerability Protection Profile

The profile will reset any connection matching **critical**, **high**, or **medium** severity threats and will collect a packet sample of the threat for research or forensics. Low and informational threats can be left to the default settings.

While these rules will provide good coverage for all signatures that match a certain severity, you may need to add a rule for a specific CVE at some point. If you create a new rule, you can tailor it to a specific CVE or even a specific vendor.

> CVE codes are maintained by MITRE at `https://cve.mitre.org`.

The vendor IDs will usually be posted by the vendor and will have a similar format to, for example, Palo Alto's PAN-SA-yyyy-nnnn or Microsoft's MSyy-nnn. If you want to, for example, filter out a specific year, you could add PAN-SA-2024 to the vendor ID. Remember to put the more specific rules matching CVEs or vendor IDs at the top of the rules.

On the **Inline Cloud Analysis** tab, you can enable inline machine-learning-powered detection for SQL injection attacks or command injection attacks. You can add a URL-based or IP-based EDL in the **Exclude from Inline Cloud Analysis** section so that they are not inspected, which can be useful for development environments where you could encounter many false positives:

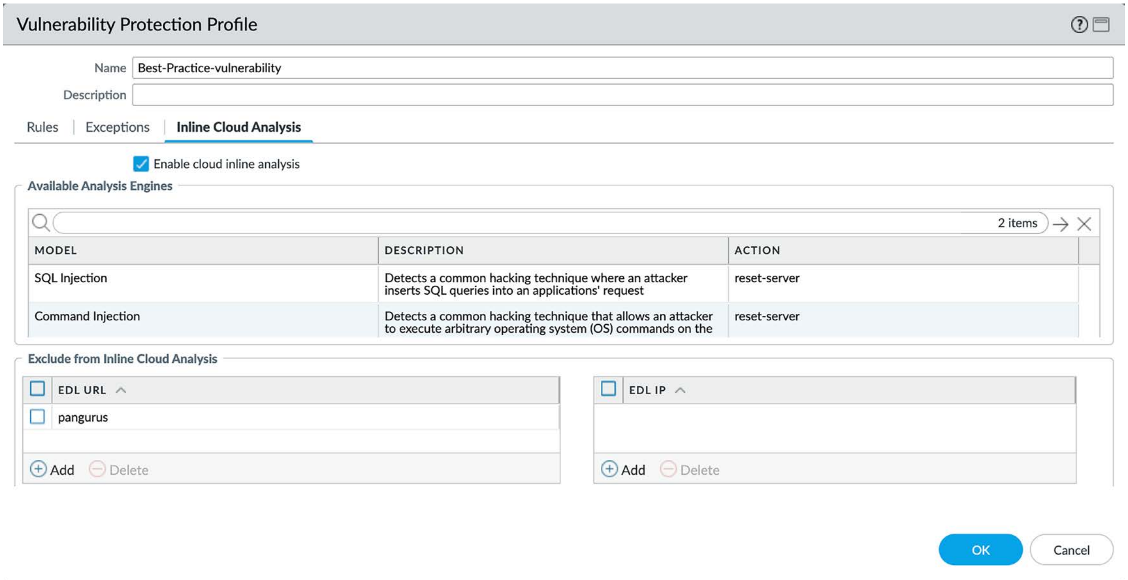


Figure 3.11: Vulnerability Protection Inline Cloud Analysis

In the next subsection, we will learn about the categories of the URL Filtering profile.

URL Filtering profile

URL filtering leverages URL categories to determine what action to take for each category.

There are two groups of categories: custom URL categories and the dynamic categories provided by the URL Filtering license.

Custom URL categories

Custom URL categories do not require a license, so you can create these objects and apply URL filtering even without access to the URL Filtering license.

Go to **Objects** | **Custom Objects** | **URL Category** to create a new custom category and add websites. There are two types of custom categories:

- **A URL list:** The URL list allows manual entry of URLs, one per line. It takes a light form of **regular expression** (**RegEx**) matched against the address, so `http://` and `https://` are not required to match.
- **A category match:** The category match allows you to combine any of the predefined categories in a custom category. This could come in handy when applying security rules or a decryption policy to a group of predefined categories.

The string used in a custom URL category is divided up into substrings, or tokens, by separators. The `./?&=;+` characters are considered separators, so `www.example.com` has three tokens and two separators. Each token can be replaced by a wildcard (*) to match subdomains or entire **top-level domains** (**TLDs**). Wildcards cannot be used as part of a token; for example, `www.ex*.com` is an illegal wildcard. Each string can be closed by a forward slash (/) or be left open by not adding an end slash. Not ending a string could have consequences if the string is very short or very common as it could match unintended longer addresses. For example, the `*.com` string could match `www.communicationexample.org`, so adding an ending slash would prevent this.

There is also an **Append Trailing Slash** function in **Device** | **Setup** | **Content-ID** | **URL Filtering** that is enabled by default. It will automatically add a trailing slash to any EDLs or custom URL categories where a slash was not added. If you do need open-ended entries, disable this function.

Having multiple wildcards would require the use of a subdomain, so `*.*.com` would match `anything.anything.com`, but not `anything.com`.

Configuring the URL Filtering profile

When configuring the URL Filtering profile, you need to select which action to apply in the **Site Access** column for each URL filtering category within the **Categories** tab, as you can see in the following screenshot:

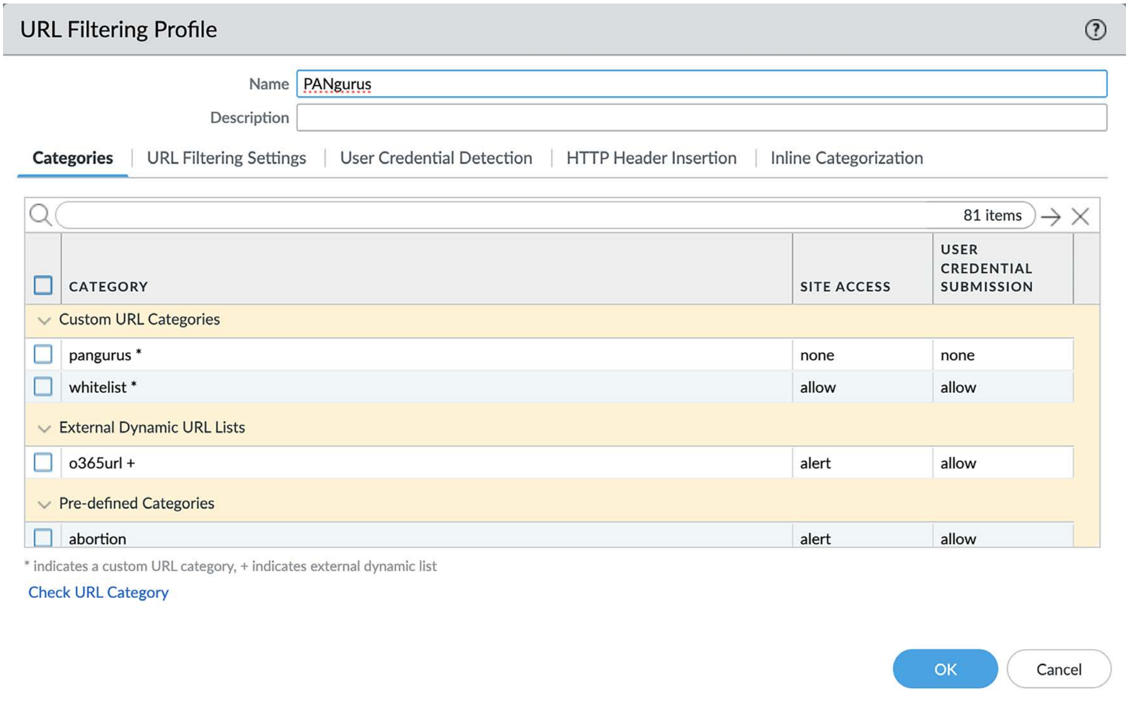


Figure 3.12: URL Filtering Profile

The available actions are as follows:

- **none:** Exclusively used for custom URL categories. This excludes the URLs contained in the custom category from being used in this URL Filtering profile.
- **allow:** Allows a category without logging.
- **alert:** Allows a category and logs the access in the URL Filtering log.
- **block**: Blocks the request, injecting an HTTP 503 error and a redirect to a page hosted on the firewall explaining to the user that their access was declined and the action was logged.
- **continue:** Injects an interactive web page informing the user that they are about to access a restricted website and provides a **Continue** button for them to acknowledge the risk associated with accessing the site.
- **override**: Injects an interactive web page that allows the user to continue if they are able to provide a password to continue. This password can be set in **Device | Setup | Content-ID | URL Admin Override**.

  Additionally, to be able to use the **override** option, an interface management profile (**Network | Network Profiles | Interface Mgmt**) needs to be created. This profile should have the **Response Pages** service enabled and added to the interface where users connect to for this page to work, as illustrated in the following screenshot:

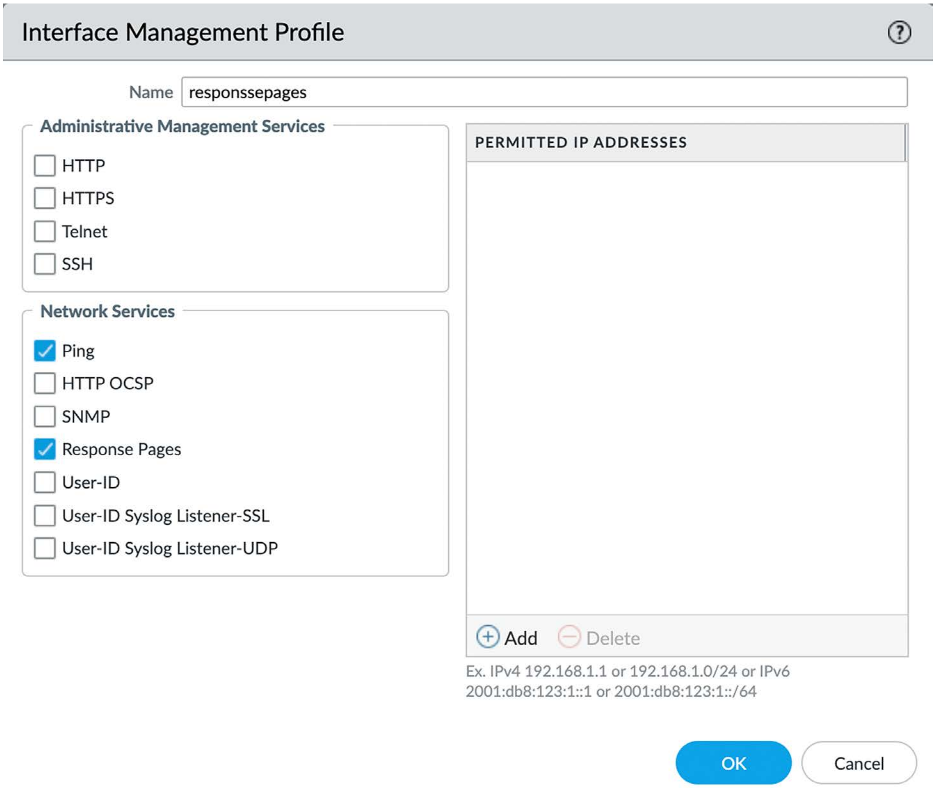


Figure 3.13: Interface Management Profile

As you saw in *Figure 3.12*, the URL Filtering profile requires each **Category** field to be set to an action individually for site access, and if **User Credential Submission** is enabled, additional filtering can be applied to decide whether a user is allowed to submit corporate credentials to a certain category. This helps prevent phishing attacks.

As you can see in the following screenshot, if you want to change a lot (or all) of the actions at once, there's a shortcut to help you. If you hover your mouse over **Site Access** or **User Credential Submission,** there will be a little arrow that lets you select **Set All Actions** or **Set Selected Actions:**

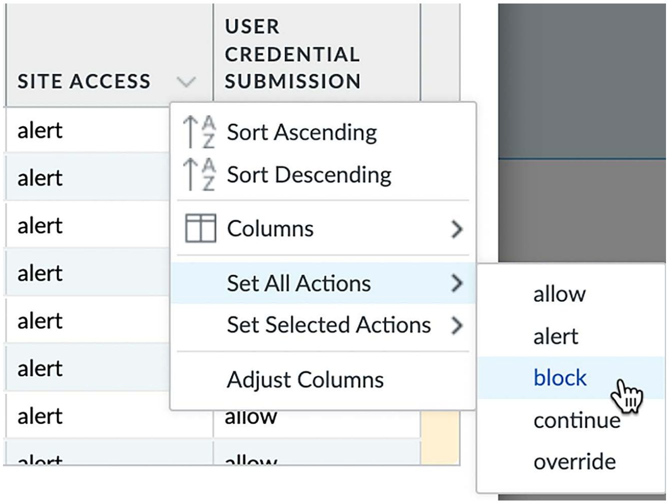


Figure 3.14: Set All Actions in the URL Filtering profile

A good baseline URL filtering policy can be set up as follows:

1. Set all of the categories to **alert.** This will ensure that all of the URL categories are logged.
2. Set **Adult**, **Command and Control**, **Copyright Infringement**, **Extremism**, **Malware**, **Peer-to-Peer**, **Scanning-activity**, **Real-time-detection**, and **Phishing and Proxy Avoidance and Anonymizers** to **Block.**
3. Set **Dating**, **Gambling**, **Games**, **Hacking**, **Insufficient Content**, **Not-Resolved**, **Parked**, **Questionable**, **Unknown**, and **Web Advertisements** to **Continue**.
4. Tweak the settings in accordance with your company policy or local laws and regulations (some URL categories cannot be logged by law, for example). The **Categories** set to **Continue** are commonly on the fringes of acceptance, but may still need to be accessed for legitimate purposes. The **Continue** action gives the user the opportunity to ensure that they intend to go to this URL before actually opening the web page.

The **URL Filtering Settings** tab contains several logging options that may come in handy, depending on your needs:

- **Log container page only:** This setting only logs the actual access a user is requesting and will suppress related web links, such as embedded advertisements and content links on the page that the user is visiting, thereby reducing the log volume.

- **Safe Search Enforcement:** This blocks access to search providers if strict safe search is not enabled on the client side. Currently, Google, Bing, Yahoo, Yandex, and YouTube are supported.

Additional logging can also be enabled:

- **User-Agent:** This is the web browser that the user is using to access a web page.
- **Referer:** This is the web page that links to the resource that is being accessed (for example, Google or CNN linking to a resource page).
- **x-forward-for:** If a downstream proxy is being used by users, this masks their original source. If the downstream proxy supports enabling the **x-forward-for** feature, it will add the client's original IP in the **c** header, allowing the identification of the original user.

The following steps and screenshot show you how to enable these settings in your URL Filtering profile:

1. Enable **Log container page only** to give your users some privacy and prevent the logging of embedded ad pages.
2. Enable **Safe Search Enforcement**.
3. Enable additional logging for **User-Agent** and **Referer**.

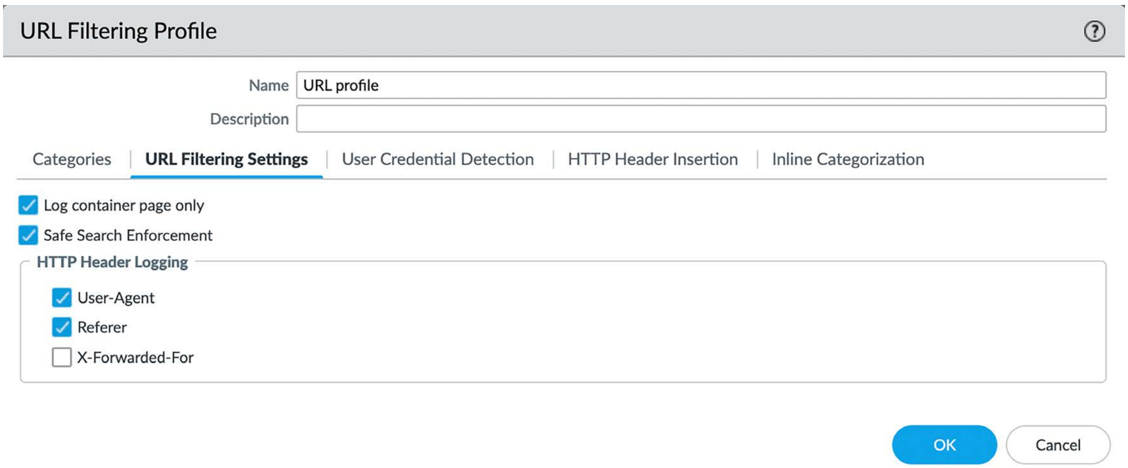


Figure 3.15: URL Filtering settings

The **User Credential Detection** tab allows you to enable credential detection (see *Chapter 6*, *Identifying Users and Controlling Access*, for more details).

HTTP Header Insertion lets you control web application access by inserting HTTP headers into the HTTP/1.x requests to application providers. As you can see in the following example, this can help you control which team IDs can be accessed in Dropbox, and which tenants and content can be accessed in Office 365 and Google app-allowed domains. You can create any URL that needs to have a certain header inserted to ensure users are accessing the appropriate instance:

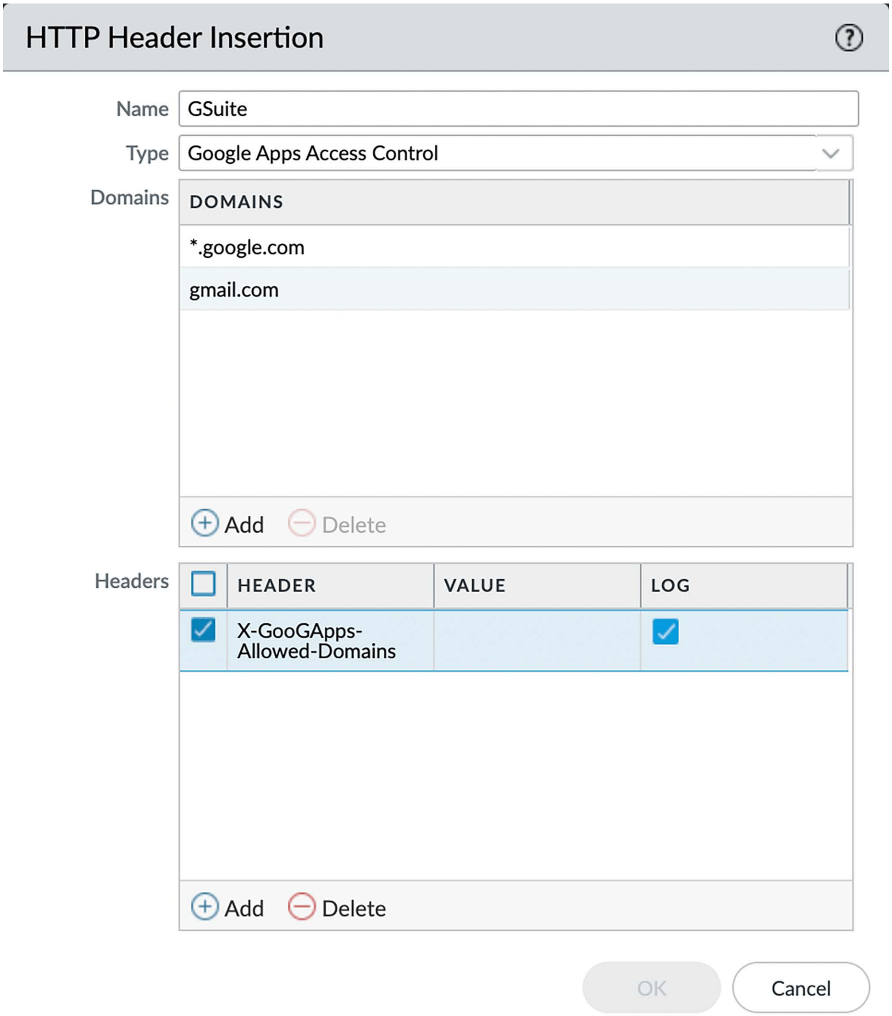


Figure 3.16: HTTP Header Insertion

The **Inline Categorization** tab lets you enable additional scanning to help identify phishing sites or malicious JavaScript and requires the Advanced URL subscription license:

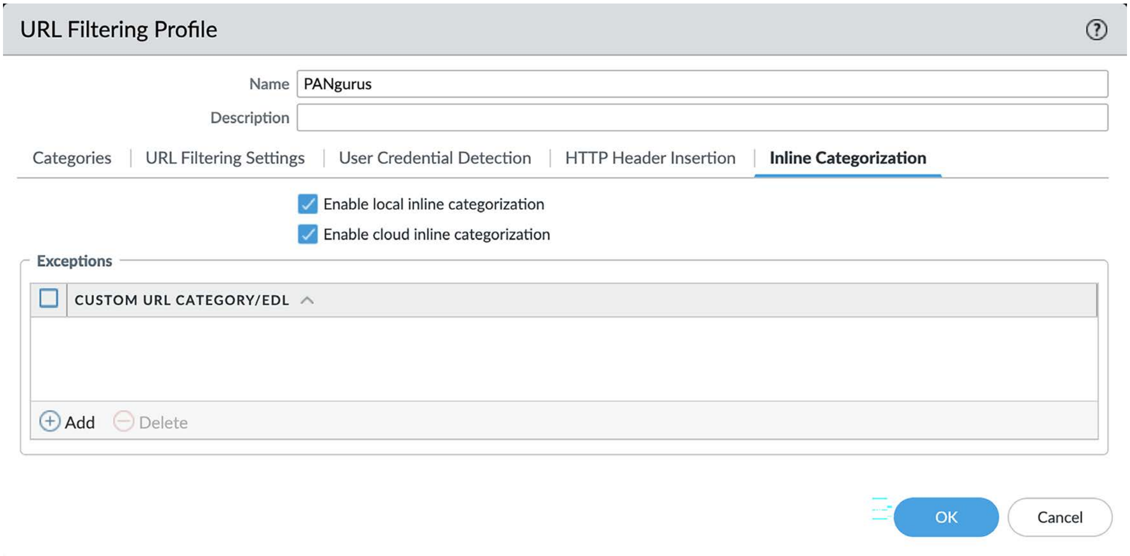


Figure 3.17: Inline Categorization

You can enable local and cloud categorization individually, in case you prefer not to perform cloud lookups or not put additional workload on the firewall.

URL filtering priorities

Some sites may fall into multiple categories and, on top of this, may be listed in a custom category or external dynamic URL list. The way URL filtering decides which action to apply is based on the severity of the action that is applied to it, and whether it is in a custom category, external dynamic URL list, or predefined category. The actions based on the severity in descending order are as follows:

1. Block
2. Override
3. Continue
4. Alert
5. Allow

The order of categories is as follows:

1. Custom URL categories
2. External dynamic URL lists
3. Predefined categories

So as an example – if a URL is present in all categories, the action of the custom URL category will be applied. If a URL is present in multiple custom URL profiles, the most severe action will be applied.

Now, let's look at the File Blocking profile.

The File Blocking profile

The default **strict file blocking** profile contains all the file types that are commonly blocked and serves as a good template to start from. Select the strict profile and click on the **Clone** action, as in the following screenshot, to create a new profile based on this one.

If any file types do need to be allowed in your organization, remove them from the block action:

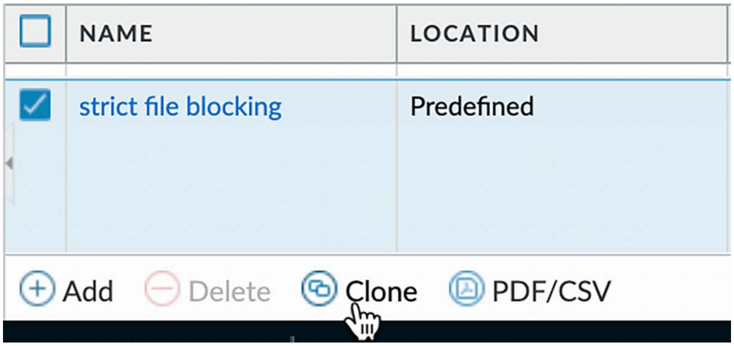


Figure 3.18: File Blocking profile clone

The direction lets you determine whether you want to only block uploads or downloads or both directions for a specific file type, as well as groups of file types. File Blocking profiles also use rules so that file types can be grouped with their own directions and actions. The default action is **Allow**, so any file type not included will be allowed to pass through (but will be scanned if an appropriate security profile is attached to the security policy). The available actions are **Alert**, **Block**, and **Continue**, which works similarly to the URL Filtering **Continue** option if the file is being downloaded from a web page that supports the HTTP redirect to serve the user a warning page before continuing with the download or upload.

Review all the file types and set the ones you want to block. Any file types that you are not sure about and would like to get a chance to review first can be set to the **Alert** action so that you can keep track of occurrences at **Monitor** | **Logs** | **Data Filtering.**

As you can see in the following screenshot, we can create sets of file types by clicking on the **Add** button, selecting the file type, and then setting the action:

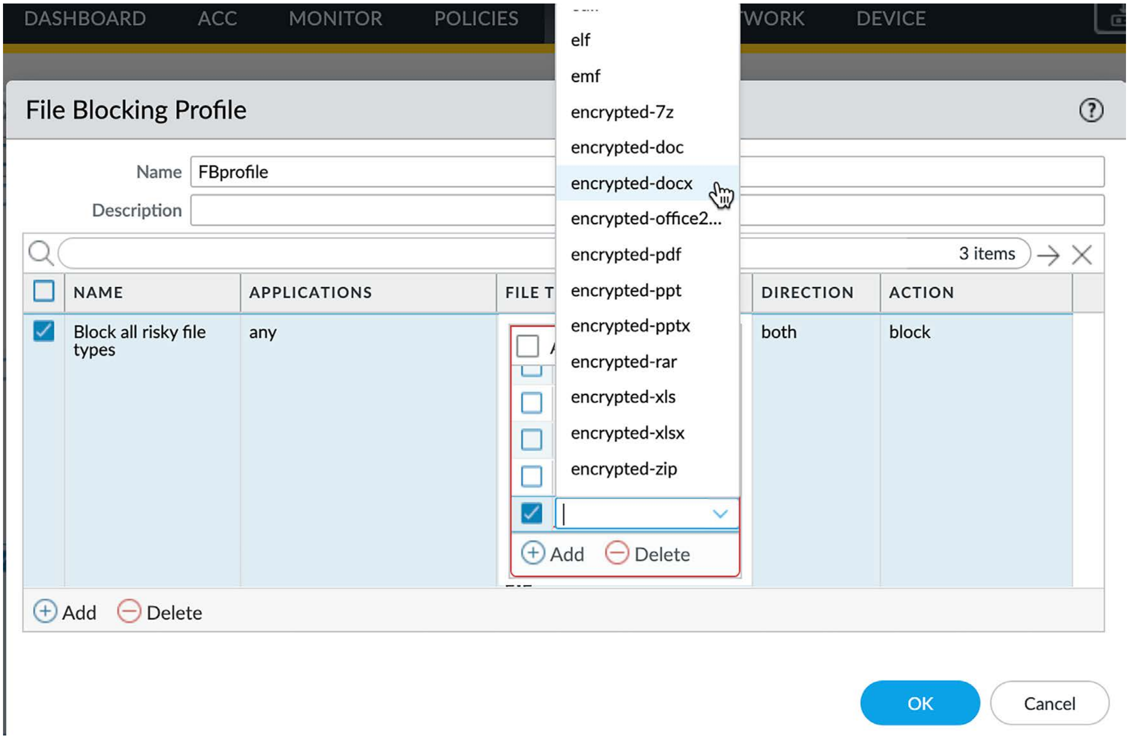


Figure 3.19: File Blocking Profile

We will now have a look at the WildFire Analysis profile.

The WildFire Analysis profile

The WildFire Analysis profile controls which files are uploaded to WildFire for analysis in a sandbox and which ones are sent to a private instance of WildFire (for example, the WF-500 appliance). Clone the default profile to upload all files to WildFire, or create a new profile if you want to limit which files are forwarded or need to redirect files to a private cloud. If no WildFire license is available, only **Portable Executables** (**PEs**) are forwarded to WildFire.

If all file types can be uploaded for inspection, simply set a rule for any application and any file type. If exceptions exist, either create a rule to divert specific files to a private cloud (if you have a WildFire appliance in your data center) or specify which files *can* be uploaded, as shown:

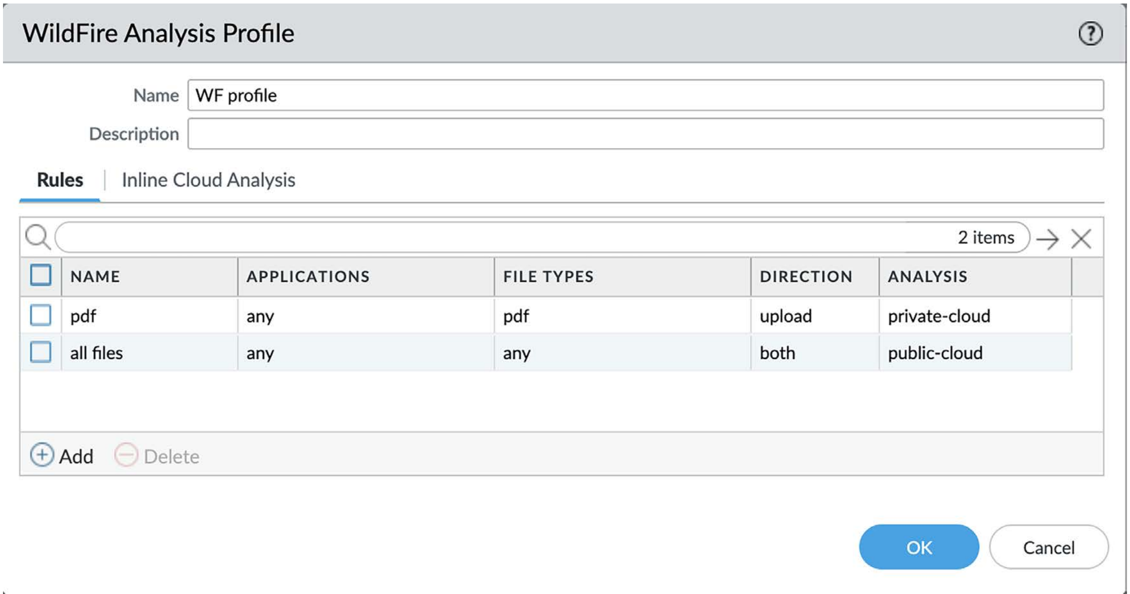


Figure 3.20: WildFire Analysis Profile

In addition to uploading files to a cloud or private sandbox, you can also send files to the cloud for inline scanning. Uploading files to the WildFire sandbox will take a little more time but will execute a file in a virtual sandbox and let it "explode" for full analysis. The inline scanning is limited to PE files but can leverage machine-learning-powered detection engines to detect advanced threats. You can create rules to determine which files for which application can be uploaded, and which action to take when malware is detected:

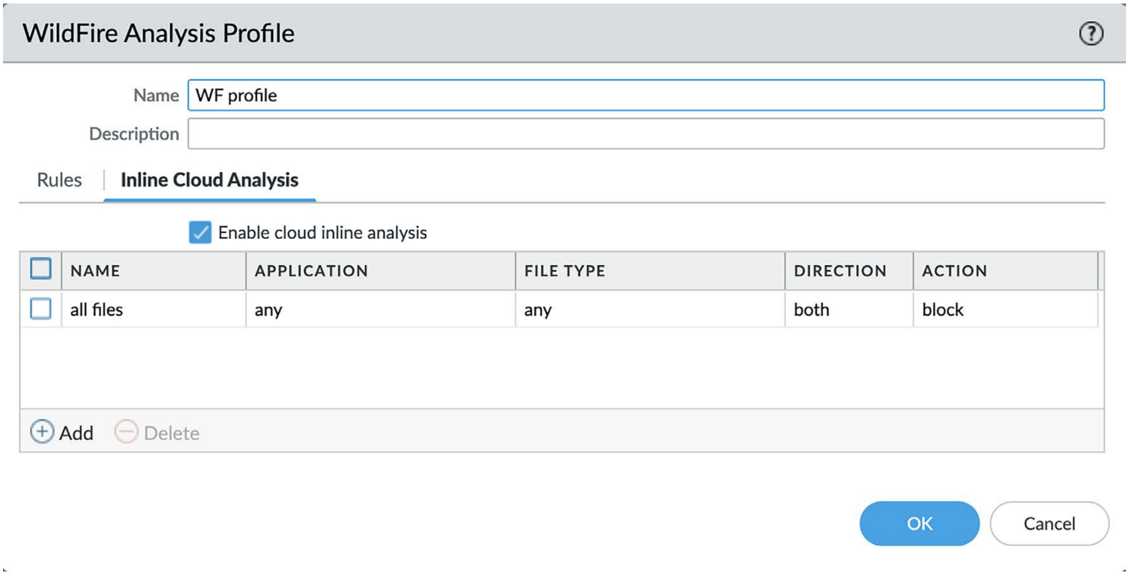


Figure 3.21: Inline Cloud Analysis

We've now covered all the security profiles, which you can use in security rules later on, but you can also create custom objects to more finely tune into specific threats or data patterns that you want to be able to identify and take action on. In the next section, we'll learn how to create these objects and what is required to be able to define the data that needs to be added for the custom object to work as expected.

Custom objects

We have already looked at custom URL categories, but, as we will discuss in this section, you can also create custom spyware and vulnerability signatures combining strings of data and RegEx to match a certain signature and take action through a security profile. This can come in handy if you are aware of a resource that could be vulnerable to a specific string of code, are subscribed to a threat feed that provides you with signatures you can add yourself, or want to control what happens when a pattern is detected.

The Custom Spyware/Vulnerability objects

You can create your own signatures using RegEx to detect spyware phone-home/C2 or network vulnerabilities. The **Configuration** tabs, as shown in the following screenshots, require basic information, such as an ID number that is between `15.000-18.000` and `6900001-7000000` for spyware and `41.000-45.000` and `6800001-6900000` for vulnerabilities, a name, a severity value, a direction, and any additional information that may be useful later on. The direction and affected client help the Content-ID engine identify which direction packets that match this signature can be expected:

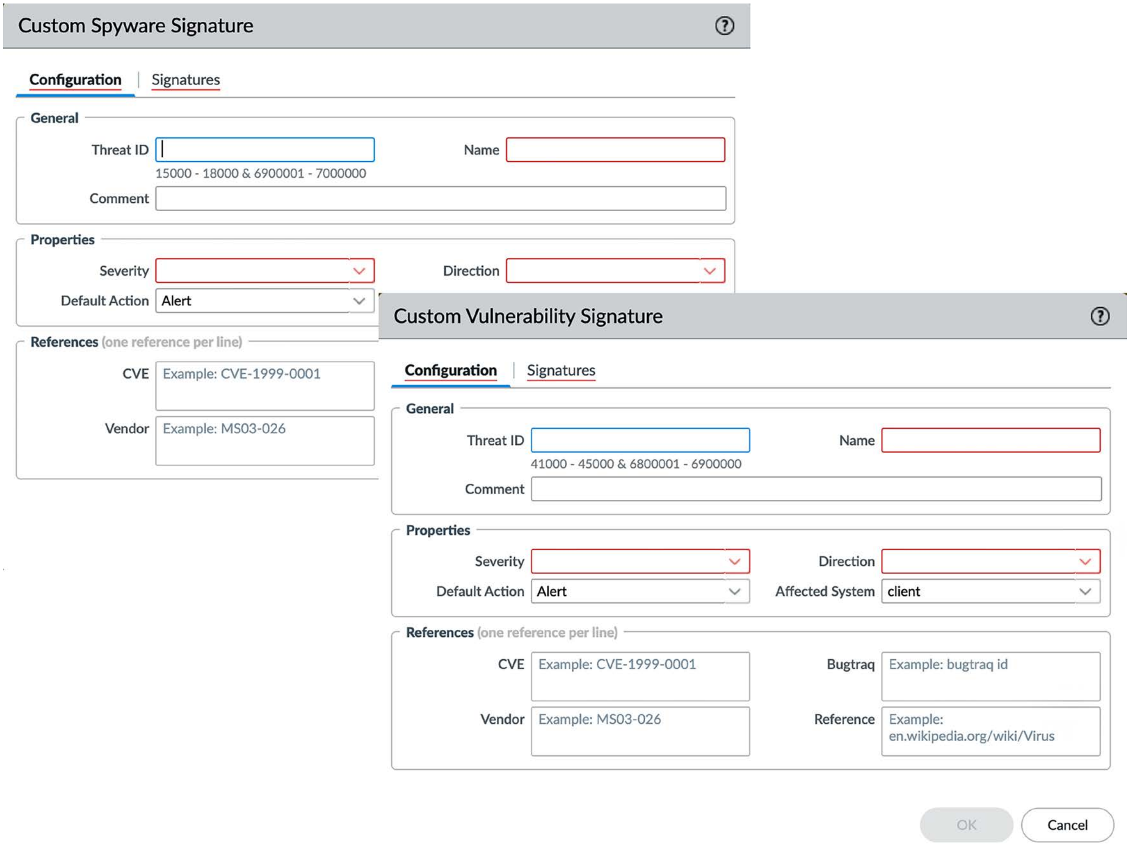


Figure 3.22: The Custom Spyware and Vulnerability objects

In the **Signatures** tab, you have two main modes of adding signatures, as you can see in the following screenshot:

- **Standard:** This adds one or more signatures, combined through logical AND or OR statements.
- **Combination:** This combines predefined (dynamic update) signatures with a timing component requiring n number of hits over x amount of time, aggregated for source, destination, or source-destination.

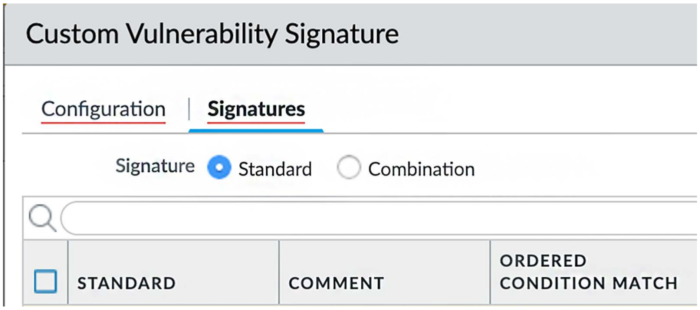


Figure 3.23: Standard or Combination signatures

Combined signatures allow you to pick predefined signatures and add a timing attribute, so action is only taken after the signature has been detected a number of times within a certain timeframe. So, let's focus on standard signatures as this is where we can build our own signatures.

From the main screen, you can add sets of signatures, which are all separated by a logical OR statement.

Once you start building a set, you need to decide on the scope. The transaction matches a signature in a single packet and the session spans all the packets in the session. If the signature you are adding to identify a threat always occurs in a single packet's payload, you should set a transaction. This will allow the Content-ID engine to stop scanning at once. If you are adding multiple strings, you can enable **Ordered Condition Match**, which requires the signatures to match from top to bottom in an ordered way. If this option is turned off, the last signature may be detected before the first. If you add multiple strings, you can link them by adding an AND condition.

A signature consists of the following:

- An **operator** is either a pattern, or a greater than, equal to, or smaller than operator. Greater than, equal to, and smaller than operators allow you to target a header, payload, payload lengths, and more. A pattern lets you match an exact string found anywhere in a packet or a series of packets.
- A **context** is where, in any of the available protocols, the signature may be found (for example, if you look for a string in `http-req-host-header`, that same string will not be matched if it is seen in the payload).

For a full list, there's a good online resource describing all the contexts at `https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClOFCA0`. However, many contexts will be self-explanatory, as you can see in the following screenshot:

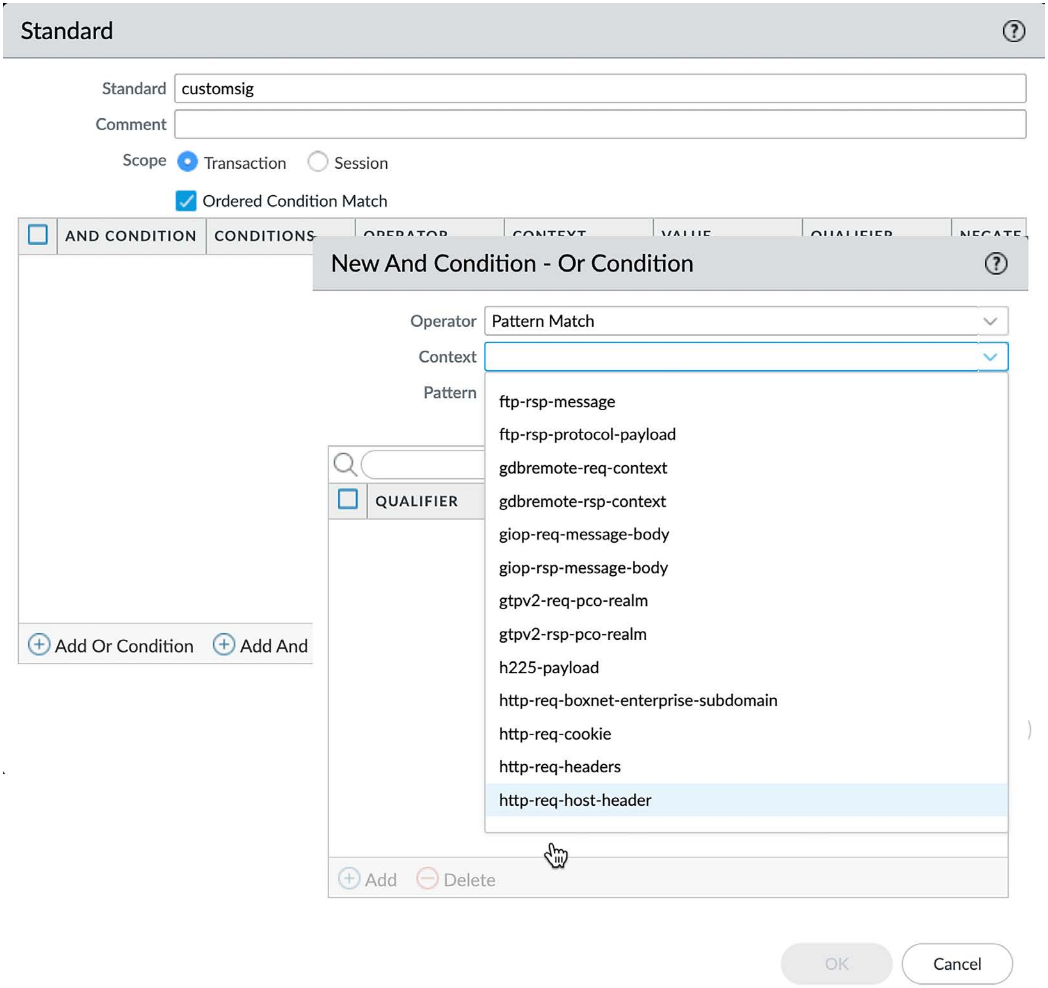


Figure 3.24: Creating signatures

- A **pattern** or **value** can also be used. If you want to, for example, match a hostname in an HTTP request header, you would use the `domain\.tld` RegEx, where the backslash indicates that the dot following it is an exact match for a dot and not a RegEx wildcard.

The available RegEx wildcard characters include the following:

<table>
<tr><th>Wildcard character</th><th>Example</th><th>Description</th></tr>
<tr><td><code>.</code></td><td><code>1.3</code></td><td>Matches a single character (e.g, <code>123</code>, <code>133</code>)</td></tr>
<tr><td><code>?</code></td><td><code>dots?</code></td><td>Matches a string with or without the last character (e.g, <code>dot</code>, <code>dots</code>)</td></tr>
<tr><td><code>*</code></td><td><code>dots*</code></td><td>Matches a string with or without the last character, and multiple repeats of the last character (e.g, <code>dot</code>, <code>dots</code>, <code>dotssss</code>)</td></tr>
<tr><td><code>+</code></td><td><code>dots+</code></td><td>Matches single or multiple repetitions of the preceding letter (e.g, <code>dots</code>, <code>dotssss</code>)</td></tr>
<tr><td><code>|</code></td><td><code>((exe)|(msi))</code></td><td>OR function to match multiple possible strings (e.g, <code>dot.exe</code>, <code>dot.msi</code>)</td></tr>
<tr><td><code>[]</code></td><td><code>x[abc]</code></td><td>Matches the preceding string followed by any character between squared brackets (e.g, <code>xa</code>, <code>xb</code>, <code>xc</code>)</td></tr>
<tr><td><code>-</code></td><td><code>x[a-z]</code></td><td>Matches any character in a range (e.g, <code>xa</code>,<code>xm</code>)</td></tr>
<tr><td><code>^</code></td><td><code>x[^AB]</code></td><td>Matches any character except the ones listed (e.g, <code>xC</code>, <code>x5</code>)</td></tr>
<tr><td><code>{}</code></td><td><code>x{1,3}</code></td><td>Matches anything after x as long as it is 1 to 3 bytes in length (e.g, <code>x1</code>, <code>x123</code>)</td></tr>
<tr><td><code>\</code></td><td><code>x\.y</code></td><td>Escape character to exactly match a special character (e.g, <code>www.\pangurus\.com</code>)</td></tr>
<tr><td>&amp</td><td></td><td>Used to match & in a string</td></tr>
</table>

Table 3.1: Supported RegEx wildcard characters

- A **qualifier** can further limit at which stage of a transaction a pattern can be matched, either in method or type. Using a qualifier is optional:

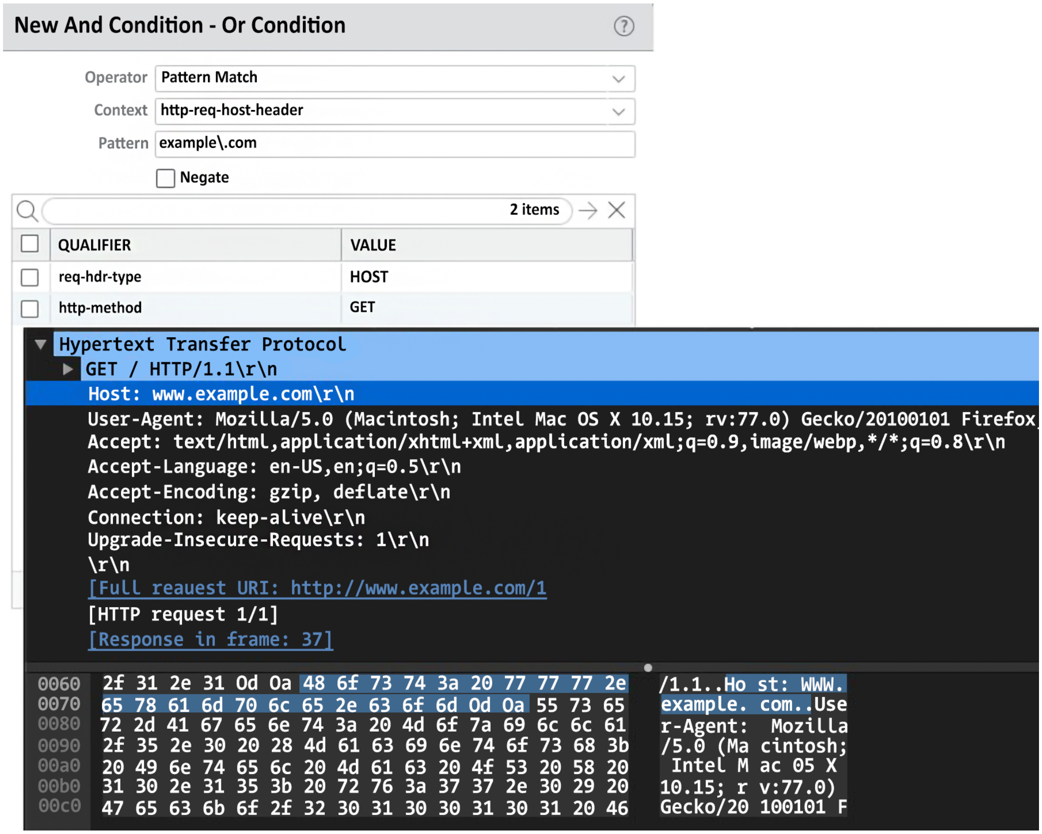


Figure 3.25: Host header pattern

Qualifiers restrict where a pattern should be located. At the top of the packet capture in the figure above we see that this is part of a HTTP **GET** (**http-method**). At the start of the payload we see the request is for **HOST** (**req-hdr-type**). Defining the qualifiers will ensure the pattern is not matched an any other type or method (e.g., POST).

A good tool to capture usable payload information from a session is Wireshark, which you can download from `https://wireshark.org`.

With the above custom objects, you are able to identify sessions behaving in a specific way, but this process can also be applied to identify information and keywords inside a session.

The custom data pattern

In the custom data pattern, you can add strings of sensitive information or indicators of sensitive information being transmitted. There is a set of predefined patterns, including social security numbers, credit card numbers, and several other identification numbers. You can use RegEx to match exact strings in documents or leverage file properties. Once the appropriate parameters have been chosen, you can add these custom data patterns to a Data Filtering profile and, as you can see in the following screenshot, assign weights. These weights determine how many times a certain marker can be hit in a session before an alert is generated in the form of a log entry and when a session should be blocked for suspicious behavior (for example, it might be acceptable for an email to go out containing one social security number, but not multiple):

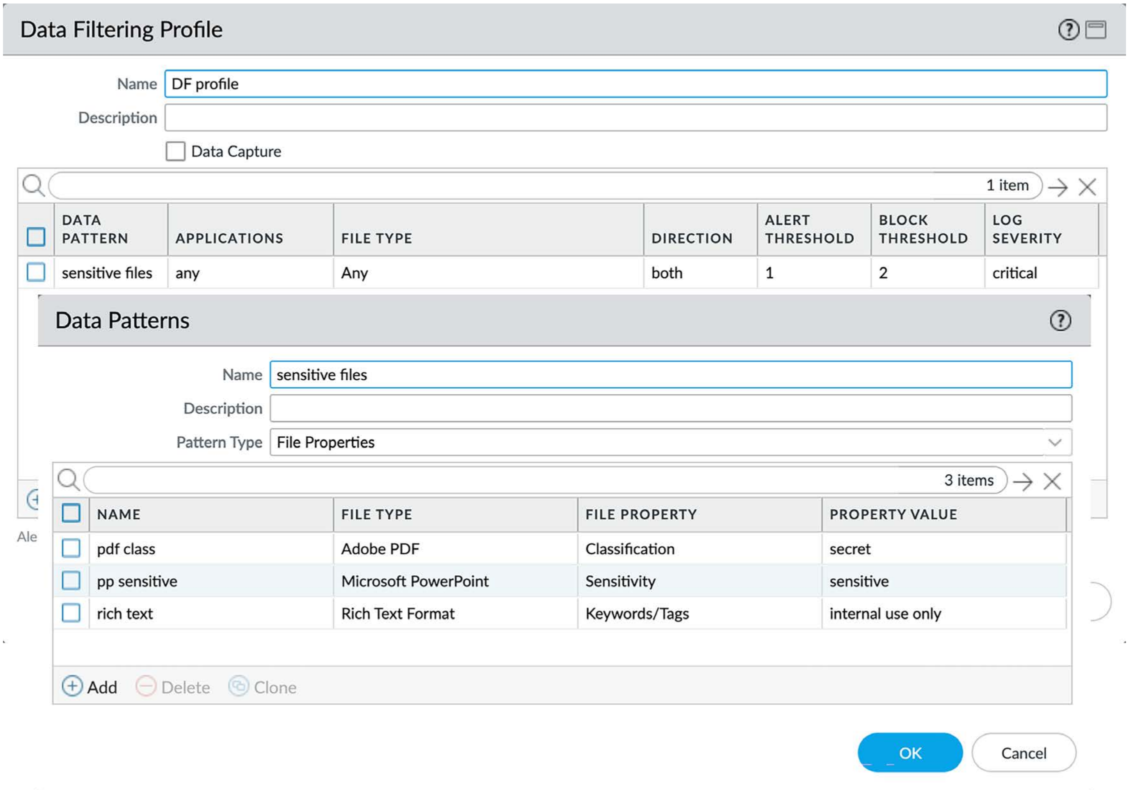


Figure 3.26: Data Filtering Profile

Now that you've had a chance to review and configure all the available security profiles, the easiest way to apply them to security rules is by using security profile groups.

Security profile groups

Now that you've prepared all of these security profiles, create a new security profile group, as shown in the following screenshot, and call it **default**. This will ensure that the group will automatically be added to every new security rule you create (it is not added to existing rules, so make sure to review them):

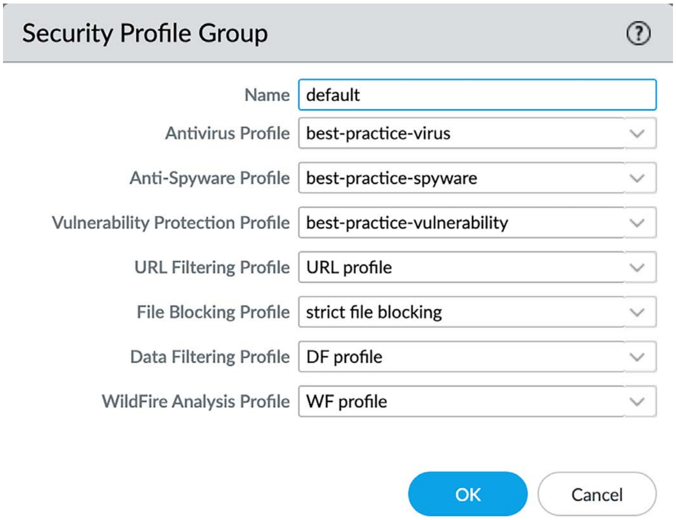


Figure 3.27: The default security profile group

It is not harmful to add *all* of the security policies to a security rule as Content-ID will intelligently only apply appropriate signatures and heuristics to applications detected in the session (for example, HTTP signatures will not be matched with FTP sessions).

Also, create a Log Forwarding profile in **Objects | Log Forwarding** called **default**, but you can leave the actual profile empty for now. This serves the same purpose as the default security profile group in that it automatically populates the log-forwarding action of each new security rule. It is easier to update the profile than to have to add a profile to each rule later on.

You are now able to build your own security profiles and add custom signatures where needed. With the information you have learned, you will be able to ensure the security rules we will be creating in the next section are set to block threats and scan content.

Understanding and building security rules

We now need to build some security rules to allow or deny traffic in and out of the network. The default rules will only allow intrazone traffic and will block everything else, as you can see here:

			Source		Destination						
	NAME	TYPE	ZONE	ADDRESS	ZONE	ADDRESS	APPLICATIO...	SERVICE	ACTION	PROFILE	OPTIONS
1	intrazone-default	intrazone	any	any	(intrazone)	any	any	any	Allow	none	none
2	interzone-default	interzone	any	any	any	any	any	any	Deny	none	none

Figure 3.28: Default security rules

In the following sections, we will start to build a rule base, making sure we first introduce some rules to drop undesirable sources and destinations, followed by adding permissive policies focused on allowing applications required by the users in a secure way, leveraging App-ID to ensure only the intended applications are let through. Finally, we'll look in more detail at which objects make up rule bases and how the rules can be kept tidy after having been in use for a while.

We will first make sure "bad" traffic is dropped by creating two new rules—one for inbound and one for outbound traffic.

Dropping "bad" traffic

The inbound rule will have the external zone as a source and the four EDLs containing known malicious addresses. These lists are updated via the threat prevention dynamic updates, so if you do not see them, make sure you have the Threat Prevention license and have downloaded and installed the latest content packages. Add a descriptive name and any relevant tags on the **General** tab. The **Source** tab should look similar to the following:

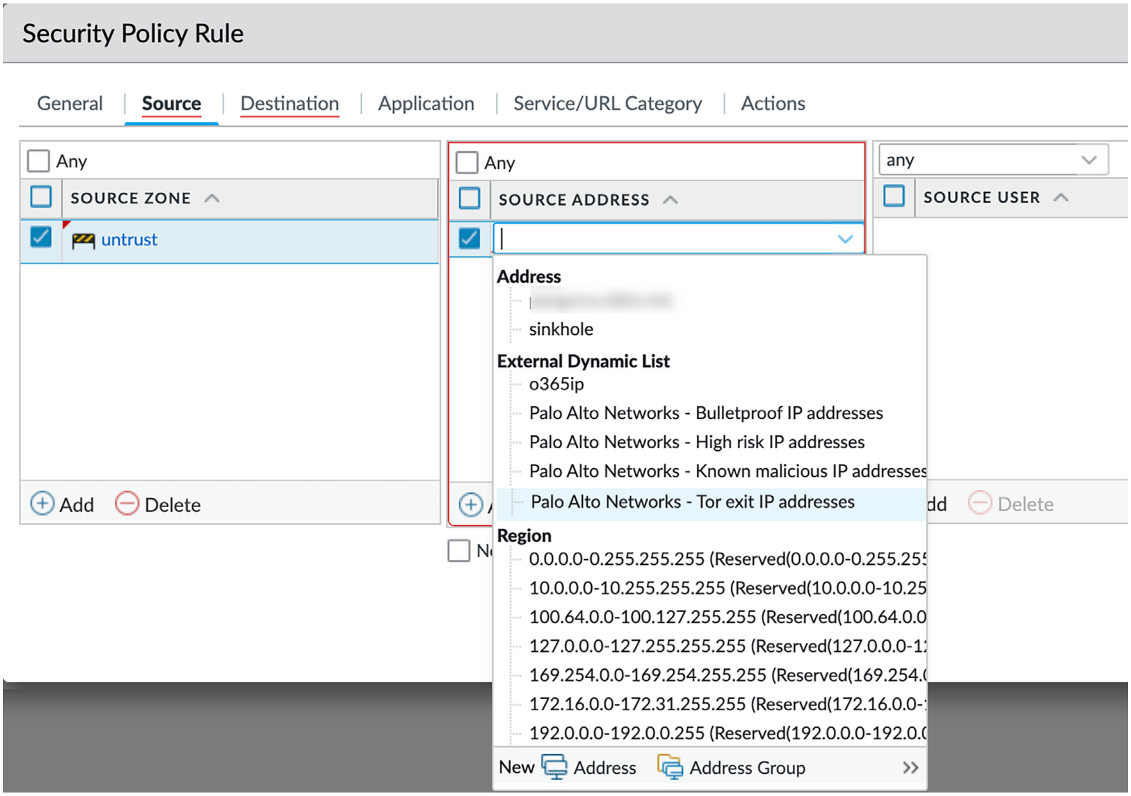


Figure 3.29: Preconfigured external dynamic lists

In the **Destination** tab, set the destination zones to both the external zone and any zone where you intend to host internal servers to which you will allow inbound NAT (for example, corporate mail or web servers) and set the destination addresses to **Any**, as in the following screenshot:

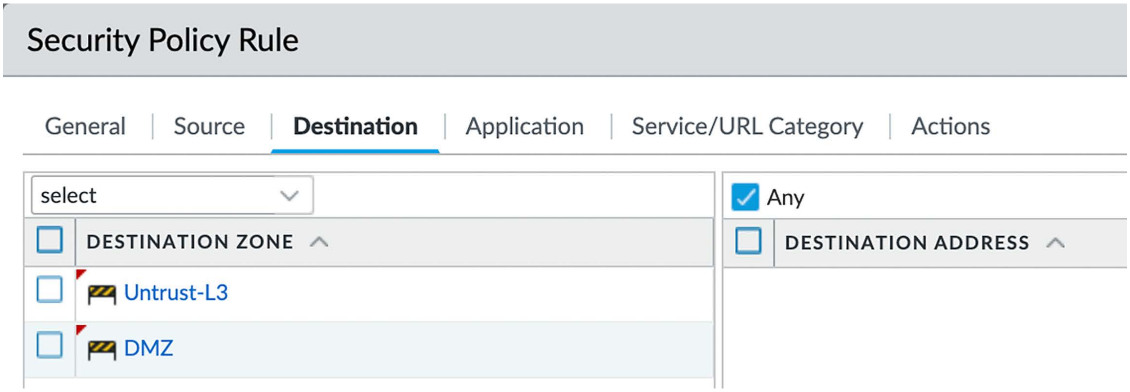


Figure 3.30: Security rule destination zones

We'll leave the **Application** and **Service/URL Category** tabs as the default for now. In the **Actions** tab, set the action to **Drop**. This will silently discard any inbound packets:

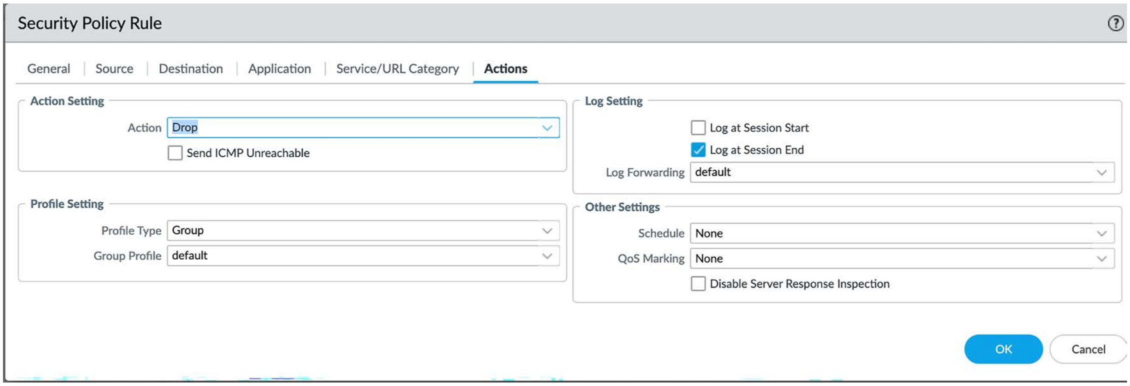


Figure 3.31: Security rule actions

Follow the next steps to create the above inbound drop rule:

1. Create a new security rule and give it a descriptive name, like **Malicious EDL Inbound Drop**.
2. Additional information can be added in the **Description** field.
3. Set the source zone to any zone that is connected to the internet (for example, **Untrust**).
4. Set the source addresses to the four (or a subset of the) predefined EDLs.
5. Set the destination zones to your internal zones that will accept inbound connections from the internet (for example, **DMZ**), also including the external zones, or simply use the **any** option.
6. Set the **Action** to **Drop**.

Important note

You may have noticed that the **Profile Setting** fields and **Log Forwarding** are filled out with the **default** profiles that you created in the previous step. In all rules where sessions are blocked, content scanning will not take place, so having these profiles will not cause overhead.

Click **OK**, and then make the reverse rule, as in the following screenshot, setting the source zones to your internal zones, the destination to the external zone, and the predefined EDL as addresses. If you changed the DNS sinkhole IP address to one of your choosing, add this IP here as well:

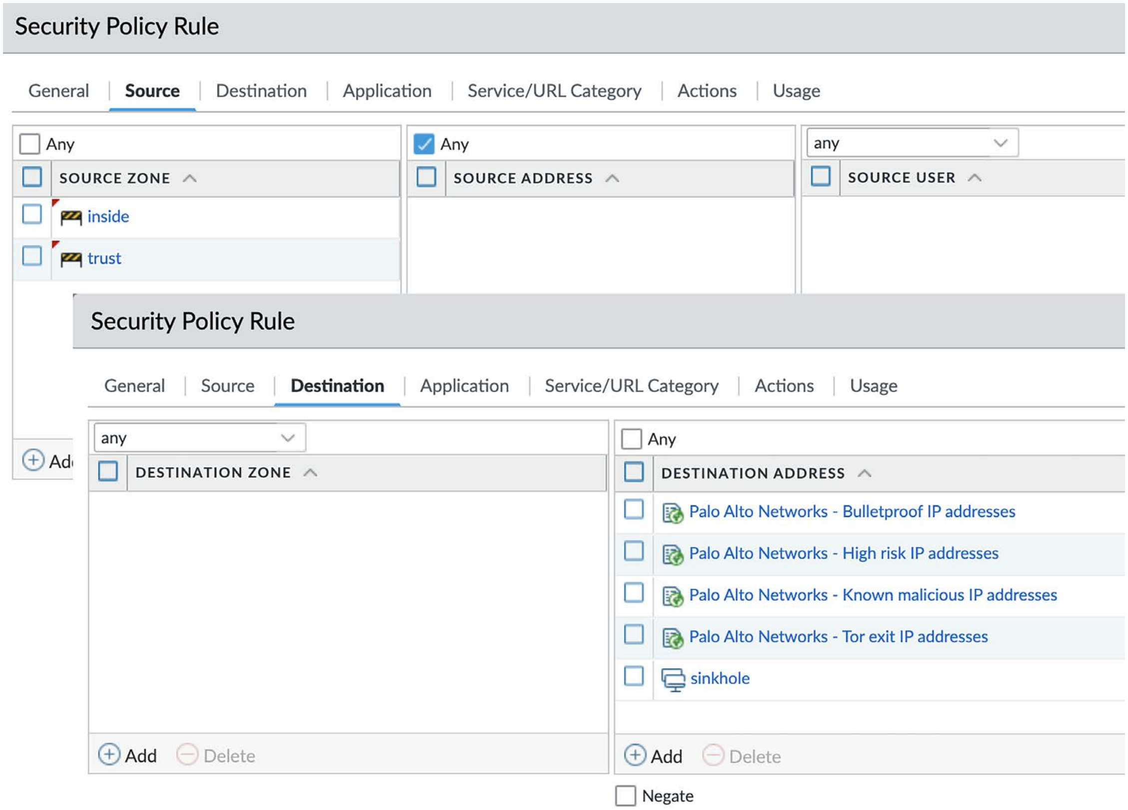


Figure 3.32: Outbound drop rules

Follow these steps to create the above outbound drop rule:

1. Create a new security rule and give it a descriptive name, like `Malicious EDL Outbound Drop`.
2. Set the source zone to your internal zones (for example, **Trust**, **DMZ**).
3. Set the destination zone to any zone leading out to the internet (for example, **Untrust**).
4. Set three destination addresses and, for each one, select one of the predefined EDLs.
5. Set **Action** to **Drop**.

A good practice is to add some **catch-all** rules to the end of your rule base, as in the following screenshot, once all the required policies have been added to catch any connections that are not allowed. From suspicious zones, one catch-all rule should be set to **application-default** and one to **any**; this will help identify standard applications that are not hitting a security policy and (more suspicious) non-standard applications that are not using a normal port (see the *Allowing applications* section to learn about the **application-default** service):

		Source		Destination							
	NAME	ZONE	ADDRESS	ZONE	ADDRESS	APPLICATION	SERVICE	URL CATEGORY	ACTION	PROFILE	OPTIONS
24	catchall	untrust	any	any	any	any	application-default	any	Drop		
25	catchall-any	untrust	any	any	any	any	any	any	Drop		
26	intrazone-default	any	any	(intrazone)	any	any	any	any	Allow	none	
27	interzone-default	any	any	any	any	any	any	any	Deny	none	

Figure 3.33: The catch-all rules at the end of the rule base

Adding an **any-any** drop rule as the last catch-all will override the intrazone default allow rule and may block some services, so ensure you account for these if you want to discard all unaccounted-for connections. Commonly overlooked connections are DHCP relay, IPSec, and GlobalProtect as intrazone.

You now have some rules actively dropping connections you do not want to get past the firewall, but there are more options available than just silently discarding packets. We'll review the other options next.

Action options

Several different actions are available to handle new connections, some of which are stealthy and some of which are noisy and informative. Choose which one to use depending on your needs:

- **Deny** will drop the session and enforce the default **Deny** action associated with an application. Some applications may silently drop while others send an RST packet.
- **Allow** allows the session to go through.
- **Drop** silently discards packets.
- **Reset Client** sends a TCP RST to the client.
- **Reset Server** sends a TCP RST to the server.
- **Reset Both** sends a TCP RST to both the client and the server.

If you check the **Send ICMP Unreachable** checkbox and the ingress interface is Layer 3, an **ICMP Unreachable** packet is sent to the client for all of the dropped TCP or UDP sessions. This can be useful to gracefully end UDP sessions as UDP does not support RST packets. It will also help the remote host to find an alternate destination or give up, whereas without the ICMP message, it may keep spamming the firewall with UDP packets.

Allowing applications

There are generally two ways to determine which applications you want to allow:

- Creating a group of known applications
- Creating an application filter to sort applications by their behavior

From **Objects** | **Application Groups**, you can create groups of known applications that can be used in security policies, as shown:

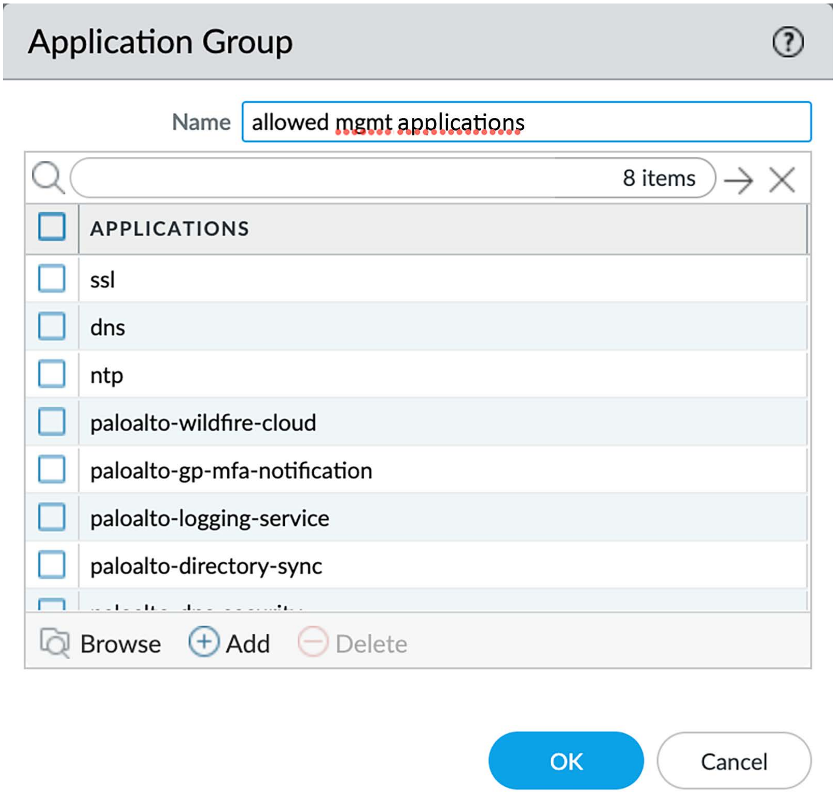


Figure 3.34: Application Group

Important note

The security rule base is evaluated from top to bottom and the evaluation is stopped once a match is found, and then the matching security rule is enforced. This means blocking rules need to be placed *above* the allowing rule if there could be an overlap.

With the widespread adoption of cloud-based hosting and cheap SaaS solutions, more traditional programs are turning into web-based applications that are accessible over a web browser. This makes it harder for an administrator to easily determine which applications need to be allowed as the needs of the business change quickly. Application filters created in **Objects | Application Filters** let you create a dynamic application group that adds applications by their attributes, rather than adding them one by one. These attributes can be selected for "good" properties to be added to allow rules (as you can see in the following screenshot) or "bad" properties to drop rules:

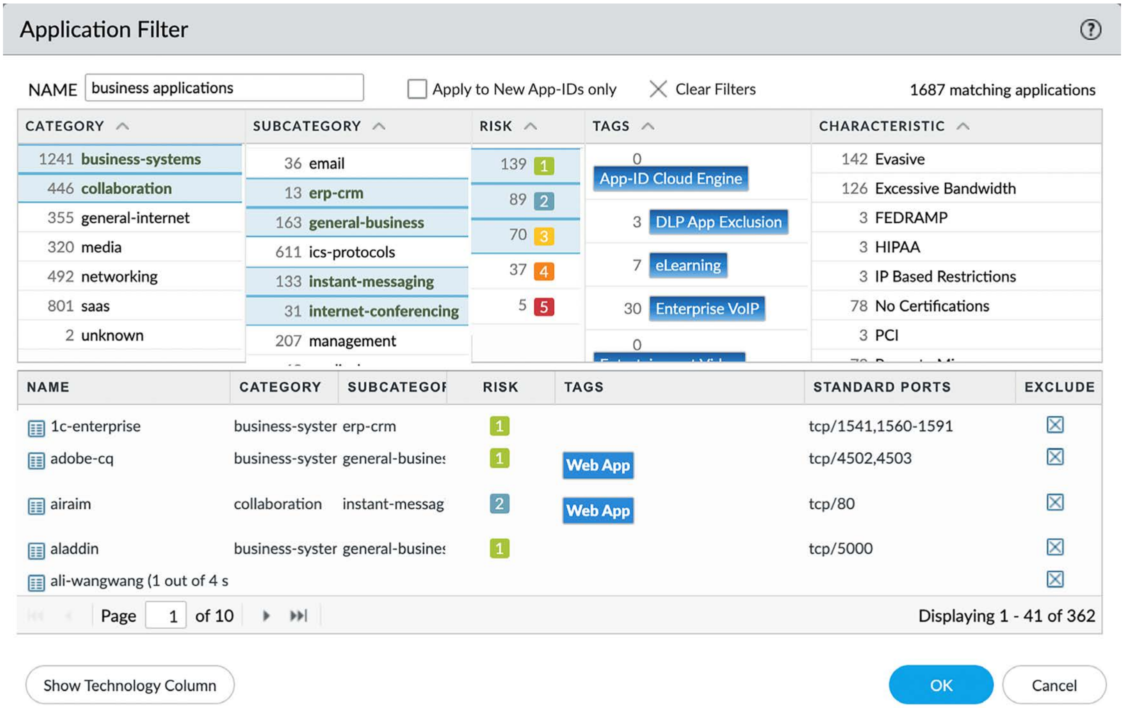


Figure 3.35: Application Filter with basic attributes

Alternatively, the filter can be based on the predefined and custom tags assigned to applications, as follows:

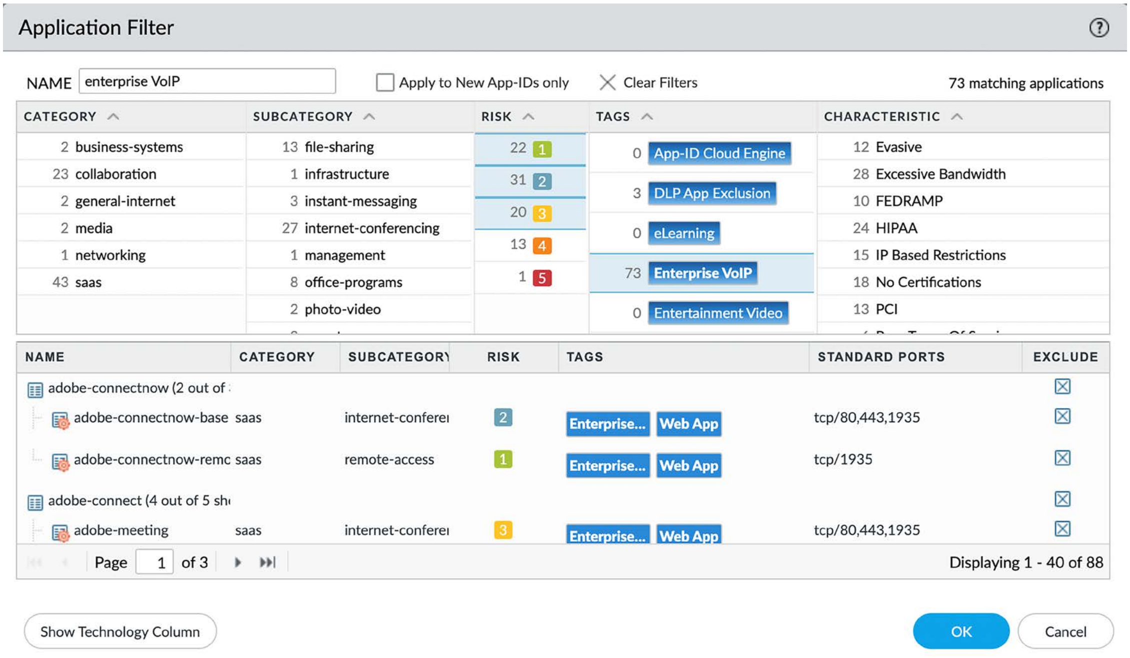


Figure 3.36: Application Filter with tags

You can mix and match application groups and filters to build further security rules by adding them to the **Application** tab, as you can see here:

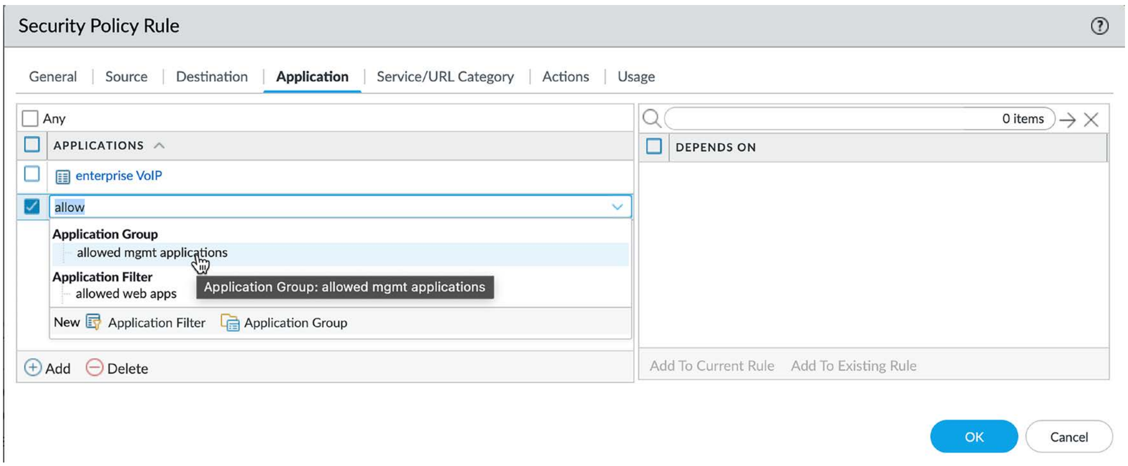


Figure 3.37: The Application tab in a security rule

To create a new **allow** rule using an application filter, do the following:

1. Create a new security rule and add a descriptive name.
2. Set the source zone to the **internal zone** that will connect to the internet.
3. Set the destination zone to **external zone**.
4. In **Applications**, add a new line and select **Application Filter**.
5. Click on all of the desired attributes and review some of the applications at the bottom. Add a descriptive name and click **OK** on the filter, and again on the security rule.

You now have an **allow** rule based on an application filter!

Application dependencies

You may notice that when you start adding applications to a security rule, some applications have dependencies. These dependencies will appear in the **Depends On** field when you add your applications. These applications rely on an underlying protocol or build on an existing, more basic application that needs to be added and allowed in the security rule base for this sub-application to work.

They do not necessarily need to be added to the same security policy.

Starting from PAN-OS 9.1, these dependencies are displayed in the security rule. As you can see in the following screenshot, they appear when you are adding new applications and can immediately be added to the same security rule or to a different one in the security rule base.

Since the firewall will go over the security rules each time a session changes to a new App-ID, you may already have the dependency satisfied in a different rule and as soon as one of the App-IDs you set in this new rule is matched, the firewall will change the session to this new rule. In older PAN-OS versions, users will only be warned about these dependencies once the configuration is committed. You can review application dependencies for individual applications via **Objects** | **Applications** too:

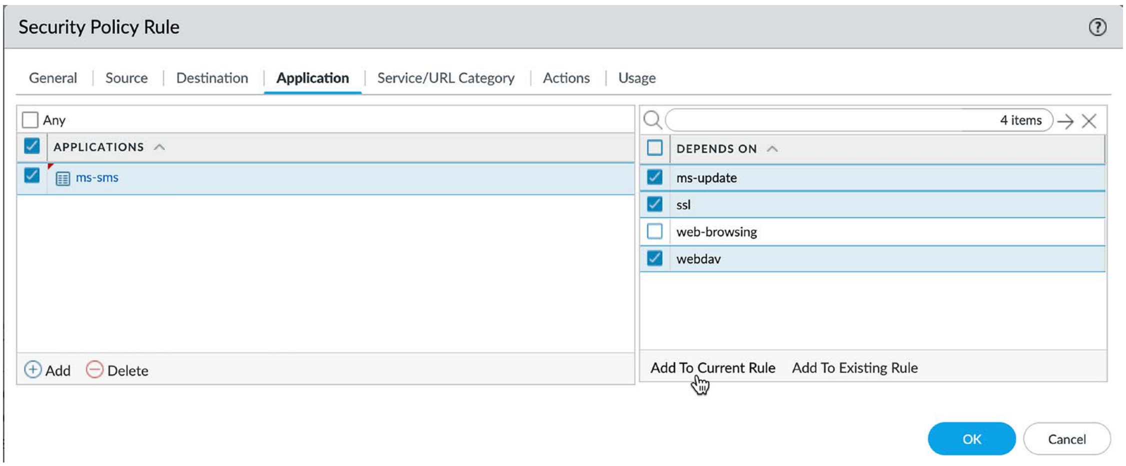


Figure 3.38: Application dependencies

Now that the applications have been set, let's look at how service ports are controlled.

Application-default versus manual service ports

Each application will use a certain service port to establish a connection. By default, each service is set to **application-default**, which forces each application to use its default ports (for example, web browsing uses ports `80` (unsecured) and `443` (SSL) secured, while FTP uses ports `21` (unsecured plaintext) and `990` (secured)).

> **Important note**
>
> Protocols that use pinholing, such as FTP, are automatically taken care of via the **Application Layer Gateway** (**ALG**), which is a part of the content decoder that is specific to this protocol. The ALG ensures "child" sessions are accepted as part of the same security rule even though the port may be different from the default port.

If an application needs a custom port, you can add a manually created service object, but this would prevent the use of **application-default**. So, any exceptions should preferably be made in individual rules to prevent applications from "escaping" via an unusual port:

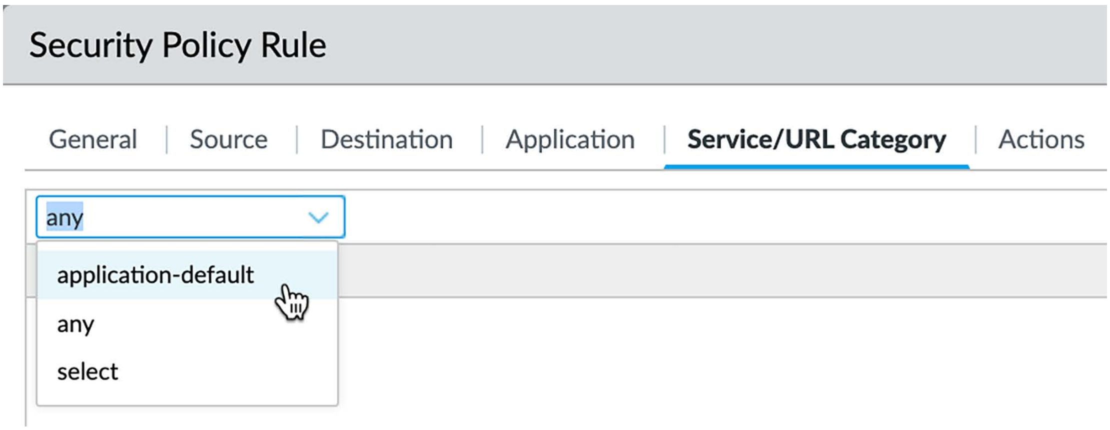


Figure 3.39: Service ports

Adding a URL category can be used to allow or block URL categories at the TCP layer:

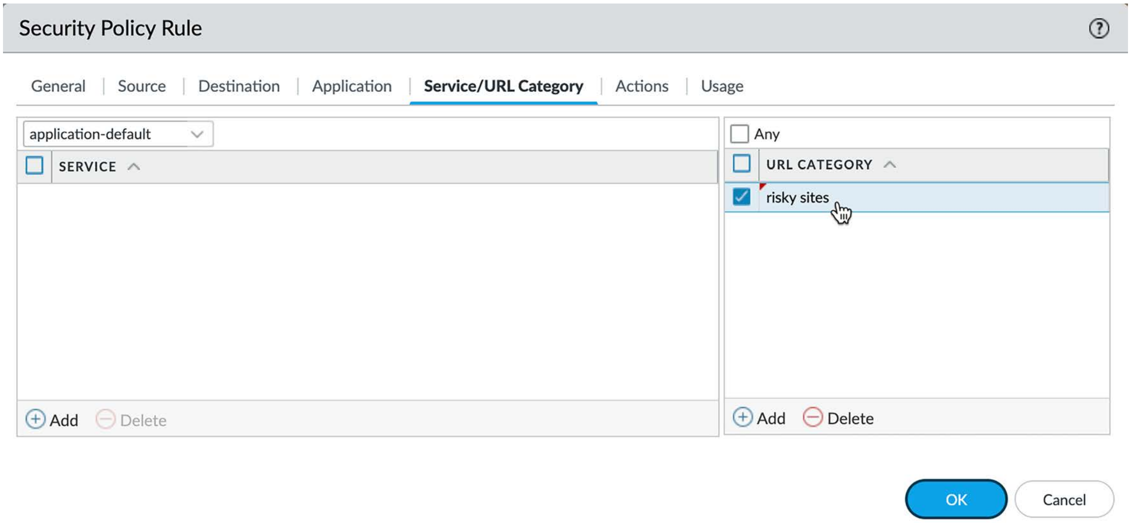


Figure 3.40: URL Category in the security rule

Rather than applying URL Filtering in (ISO) Layer 7, a URL category added directly to the security rule simplifies allowing or denying access to a specific category.

When an outbound connection is identified as web browsing (or SSL), the content decoder will identify the URL that is being accessed and a category lookup will take place. If a URL category is set in the security rule, as illustrated in *Figure 3.30*, a Layer 3 action can immediately be applied to allow or deny a session from proceeding.

This is different from URL Filtering via a security profile in that URL Filtering will compare the URL found in a session against all predefined URL categories and custom URL categories. This could introduce complications with overlapping categories or may make a rule too wide.

For example, if only a specific category needs to be allowed or denied, a traditional URL Filtering security profile would need to be set to allow the desired categories and block all the other ones, which may disrupt other more generic web browsing security rules. A typical example is a printer or other IoT device that needs access to a preset list of URLs to receive updates or report metrics, but no other URL categories. Using a custom URL category directly in the security rule limits that device's access to only these FQDNs.

Controlling logging and schedules

By default, each security rule is set to **Log at Session End**. This means that a log is only written to the traffic log once a session is broken down. For some sessions, it may be interesting to log more interactions, and so **Log at Session Start** could be enabled. This does cause quite a lot of overhead, however, as there will be a log for each new stage of a session when the SYN packet is received and for every application switch. So, there could be two to five additional log entries for a single session.

Other applications that are very chatty or less relevant may not need to be logged at all, such as DNS.

Important note

Even with both the start and end logs disabled in the **Security Policy Rule** | **Actions** tab, any threats detected in a session will still be logged to the threat log.

Log forwarding can be used to forward logs to Panorama, Strata Logging Service, or a syslog server or to send out an email. If you name one of the log forwarding profiles **default**, it will automatically be added to every new security rule that is created:

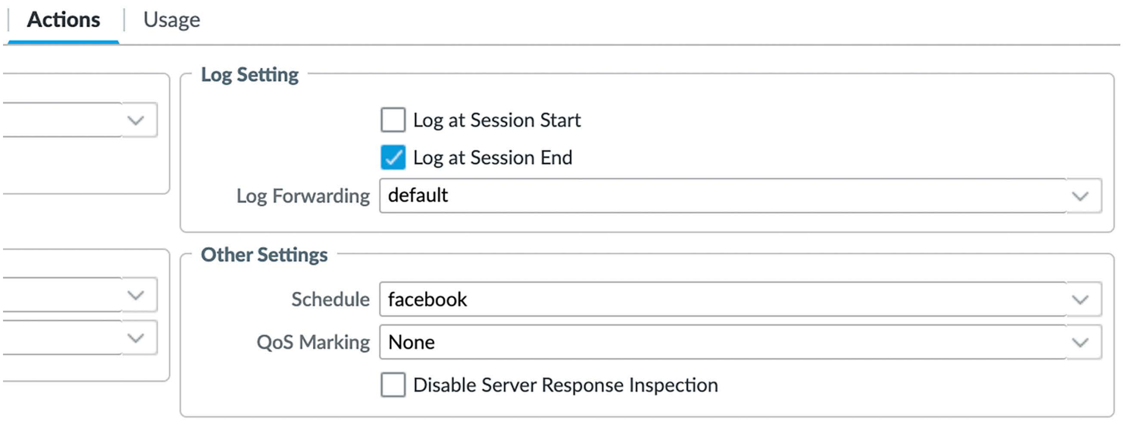


Figure 3.41: Log options and schedules

Schedule can be used to create timeframes when this security rule will be active if certain applications are only allowed at specific times of the day (for example, Facebook can be allowed during lunch and after hours):

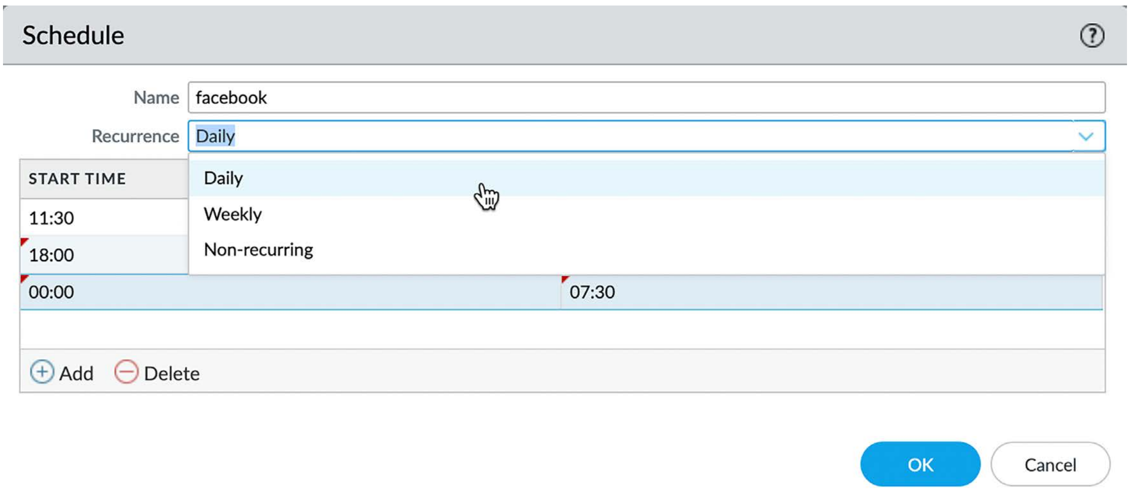


Figure 3.42: Schedules

Before you continue putting this new knowledge to work and start creating more rules, let's review how you can prepare address objects so the rule base becomes more readable and you can reuse similar objects in multiple rules.

Address objects

To make managing destinations in your security and NAT policy a little easier, you can create address objects by going to **Objects** | **Addresses**. When you create a new object here, you can reuse the same object in different rules, and if something changes, you only need to change the address object for all the security and NAT rules to be automatically updated:

1. Click on **Add** and provide a descriptive name for the address. It is good practice to set up a naming convention so that you can repeat this process for all other address objects. A good example is to prefix all server names with `S_` and all networks with `N_` so that they're easily identifiable.
2. Set a description if needed.
3. Select the type of object that this will be:
   - **IP Netmask** lets you set an IP with a subnet mask down to `/32` or `/64` for a single IPv4 or IPv6 address (no need to add `/32`).
   - **IP Range** lets you define a range that includes all the IP addresses between the first and last IP set in the range, separated by a dash (-).
   - **IP Wildcard Mask** lets you set a subnet masking that covers binary matches, where a zero bit requires an exact match in the IP bit, and **1** is a wildcard. So, for example, a wildcard subnet of `0.0.0.254` translates to `000000000.0000000.0000000.11111110`. The first three bytes are set, and in the last byte, all but the first bit are wildcards. This means that if the associated IP address is set to `10.0.0.2` (`00001010.0000000.000000.00000010`), all of the IPs in the subnet that end with `0` will be matched (that is, all of the even IP addresses). If the IP is set to `10.0.0.1`, all of the odd IPs would match. This type of object can only be used in security rules.
   - **FQDN** lets you set a domain name that the firewall will periodically resolve according to the **time to live** (**TTL**) and cache. Up to 10 `A` or `AAAA` records are supported for each FQDN object. Use the **Resolve** link to verify that the domain can be resolved.
4. Add a tag to easily identify and filter policies for this object.
5. Click **OK**.

Once you have sets of objects that are similar, you can also create groups by going to **Objects** | **Address Groups**. These groups can be used to bundle objects for use in security or other policies.

Tags

Tags can be leveraged to group, filter, or easily identify many other objects. Security zones, policy rules, or address objects can all be tagged with up to 64 tags per object. By going to **Objects** | **Tags**, you can create new tags:

1. Click on **Add** and create a descriptive and preferably short name for the tag (up to 127 characters). You can also use the dropdown to select one of the already-created security zones, which will cause the tags to be automatically assigned to this zone.
2. Select a color or leave it as **None**.
3. Add a comment.
4. Click **OK**.

As you can see in the following screenshot, tags can then be used to visually enhance your rule base or to filter for specific types of rules:

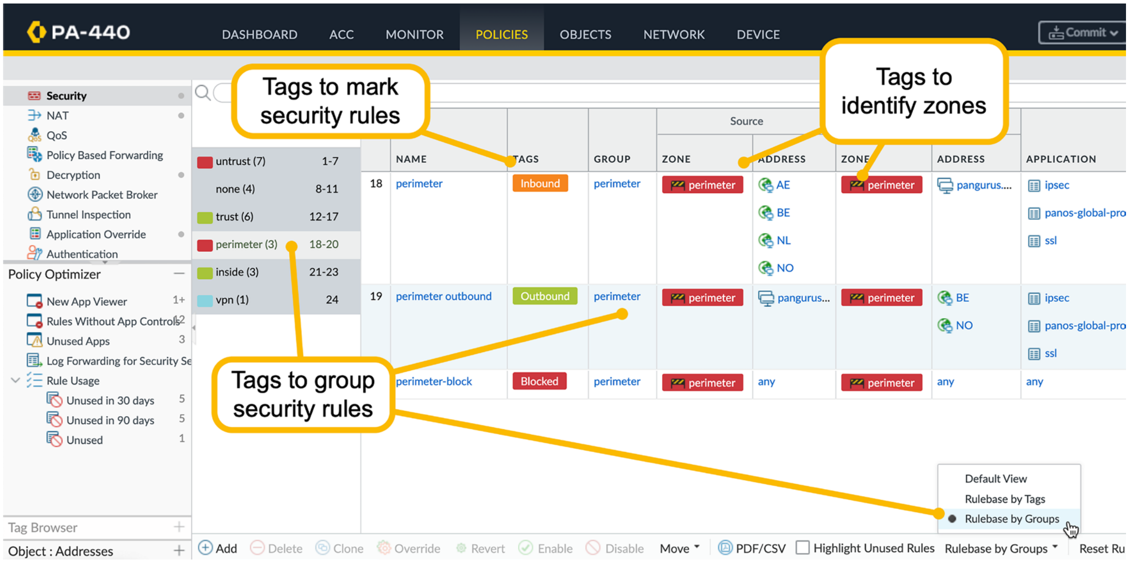


Figure 3.43: Tags in the security policy

While building security rules, objects (such as addresses, applications, and services) can be *clicked and dragged* from the object browser on the left into any rule, and from one rule to another. There is no need to open a rule and navigate to the appropriate tab to add objects.

After a while, you will want to review the security rule base you've built to make sure you haven't missed any applications, left rules too open, or have any duplicates that leave rules unused. Let's take a look at how this can be done.

Policy Optimizer

While you're on the **Security** or **NAT policy** tab, there's a tool called **Policy Optimizer** in the bottom left-hand corner that can help improve your security rules by keeping track of rule usage. Policy Optimizer records statistics relating to your rules and can report the following:

- Rules that have been unused for 30 days, 90 days, or for all time so that you can delete them
- Rules that are set up with no applications defined and the applications that were accepted by those rules

- Rules that have applications that are not being used so that you can remove these excess applications

After the rule base has been in production for a while, the output of Policy Optimizer will start to resemble the following screenshot:

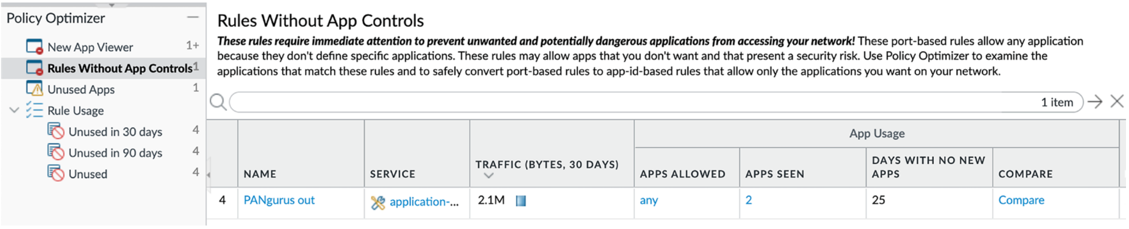


Figure 3.44: Policy Optimizer

In tandem with Policy Optimizer, each rule also has a column called **Apps Seen**.

The Apps Seen column

Each security rule will also gather information about the applications passing through it. This information can be accessed from the **Apps Seen** column in the security policy:

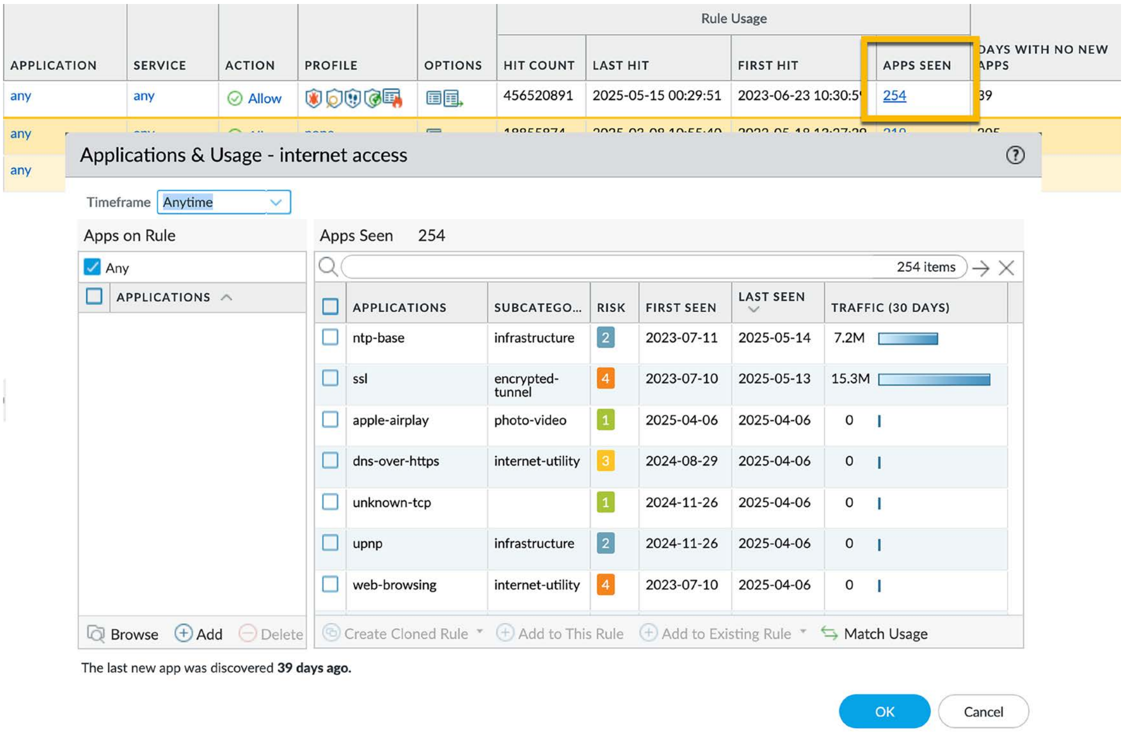


Figure 3.45: Apps Seen

All detected applications can be added to the current rule, an existing rule, or a cloned rule from the current rule.

Creating NAT rules

Unless you are one of the lucky few organizations that were able to get their very own `A` (`/8`) or `B` (`/16`) class subnets, your internal network segments will most likely be made up of one or more of the well-known RFC1918 private IP address allocations: `10.0.0.0/8`, `172.16.0.0/12`, or `192.168.0.0/16`. NAT is needed in order for your hosts to be able to reach the internet and your customers and partners to reach publicly available resources hosted in your data center. NAT rules can be configured through **Policies** | **NAT**.

For this section, keep the following interface setup in mind:

INTERFACE	INTERFACE TYPE	MANAGEMENT PROFILE	LINK STATE	IP ADDRESS	VIRTUAL ROUTER	SECURITY ZONE
ethernet1/1	Layer3			198.51.100.2/24	default	Untrust-L3
ethernet1/2	Layer3			192.168.27.1/24	default	Trust-L3
ethernet1/3	Layer3			10.0.0.1/24	default	DMZ-L3

Figure 3.46: Interface zone and IP configuration

Address translation comes in different flavors depending on the direction and purpose, each with its own nuances. Let's first review inbound NAT.

Inbound NAT

For inbound NAT, it is important to remember that the firewall is zone-based and that the source and destination zones are determined *before* the NAT policy is evaluated:

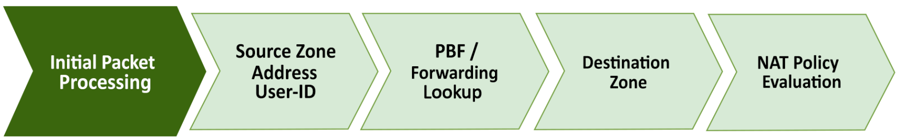


Figure 3.47: Packet flow stages

This means that for inbound NAT, the source and destination zones will be identical. The routing table will determine the source zone based on the default route and the destination zone based on the connected network, which is configured on the external interface.

For example, if the `203.0.113.1` internet IP is connecting to the `198.51.100.2` firewall IP to reach the `10.0.0.5` server, the firewall will look up `203.0.113.5` in its routing table. It will find that it only matches the default route, `0.0.0.0/0`, which points from the **ethernet1/1** interface, which is in the **Untrust-L3** zone. It will then look up `198.51.100.2` (the original destination IP in the packet header) and find it in the `198.51.100.0/24` connected network on the **ethernet1/1** interface, which is in the **Untrust-L3** zone.

The **Original Packet** tab needs to have the following:

- The same source and destination zones.
- **Source Address**, which can be **Any** for generic internet sources, specific IP addresses, or subnets if the source is known.
- **Destination Interface**, which indicate which interface the packet is headed to. This can be important in cases where there are multiple interfaces with overlapping routes.
- **Service**, which can be used to restrict which destination port is allowed in the received packets. This will help in cases where the IP space is restricted and **port address translation** (**PAT**) is required to host different services on the same external IP and will prevent over-exposing an internal host.
- **Destination Address**, which needs to be a single IP for a one-to-one destination NAT (don't add a subnet). Having a subnet-based destination NAT is possible, but only for **Session Distribution**.

The **Original Packet** tab of an incoming NAT rule will look similar to the following screenshot:

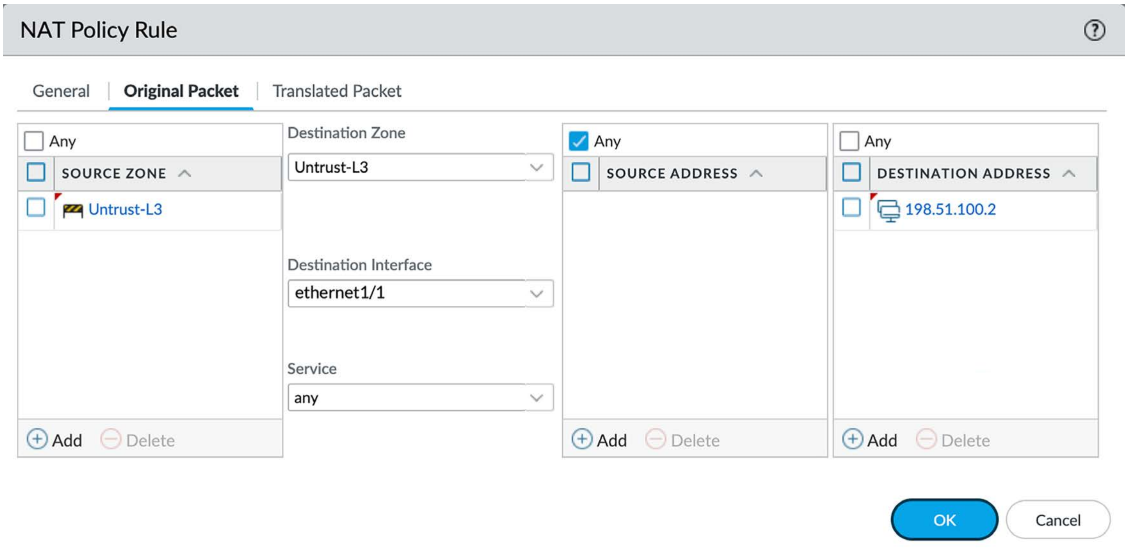


Figure 3.48: Original Packet NAT

In the **Translated Packet** tab, you can set what needs to be changed for the external client to be able to reach the internal server:

- **Source Address Translation** will usually be set to **None**, but it can be set to match an internal interface subnet or loopback interface if required. This would let the server receive a packet sourced from an internal IP, rather than the original internet IP.
- **Destination Address Translation** with **Translation Type** set to **Static IP**, also known as one-to-one NAT, changes the destination IP to a single internal server.
- **Translated Port** can be used if the internal service runs on a different port than the externally advertised one. For example, externally, a web server could be reachable on default SSL port `443`, while on the server itself, the service is enabled on `8443`.

In the **Translated Packet** tab of an inbound NAT rule, we will add the private IP of the destination server, as illustrated here:

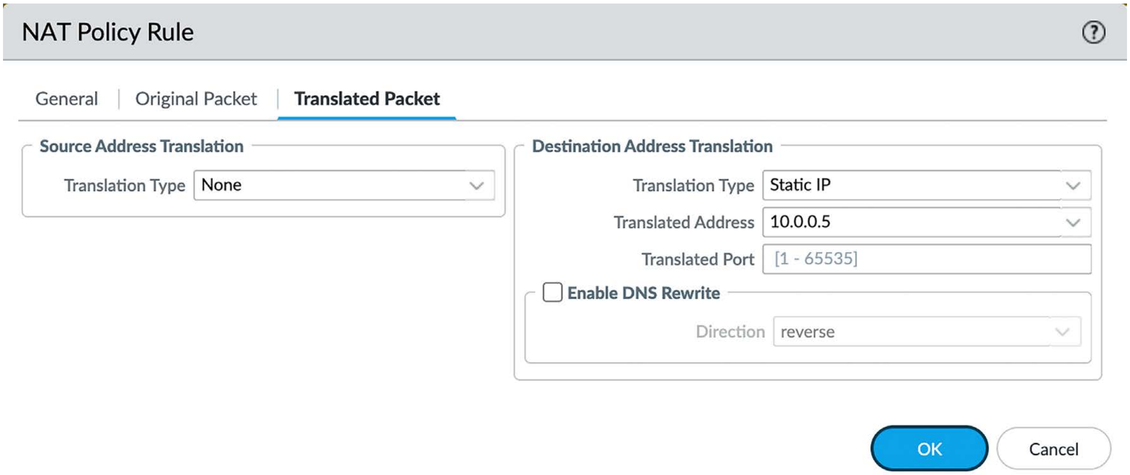


Figure 3.49: Translated Packet NAT

Next, let's take a look at address translation in the opposite direction.

Outbound NAT

Outbound NAT rewrites the source IP addresses of internal clients to the interface associated with a different zone. This could be an internet-facing zone or one connecting to a partner, VPN, or WAN, as in the following screenshot:

- The source zone will reflect the interface that the clients are connected to.

- The destination zone and destination interface will reflect the egress interface that a routing lookup determines based on the original packet.

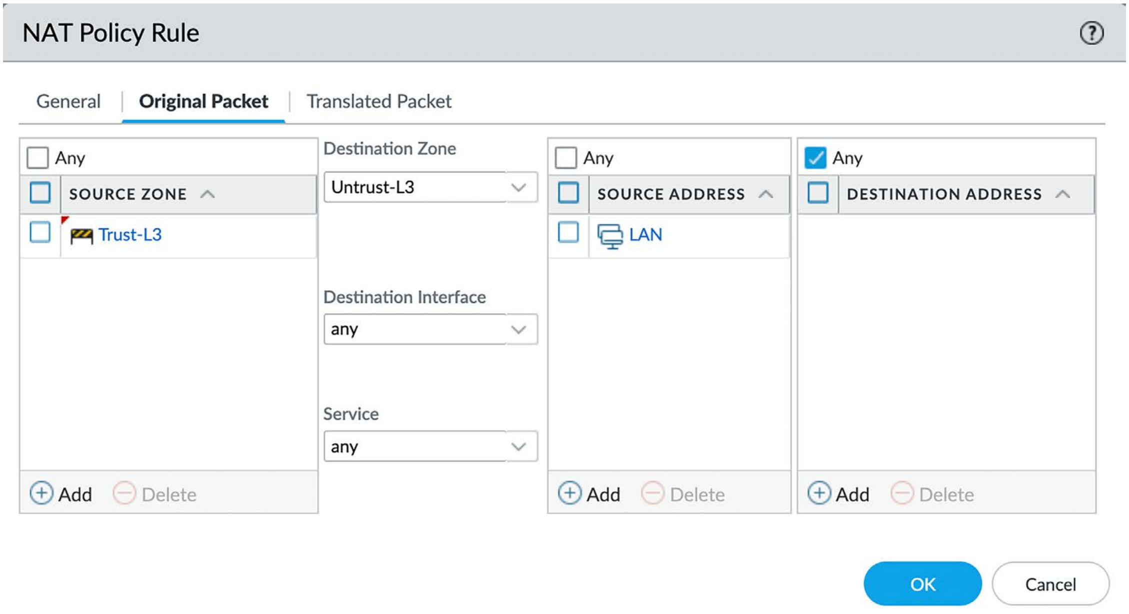


Figure 3.50: Outbound NAT Original Packet

When using an IP pool for source translation, the firewall will use proxy ARP to gain ownership of IP addresses. This means that you don't need to physically configure all of the IP addresses on an interface, but it is recommended that you have at least the subnet configured on an interface so that the firewall knows which interface is used to broadcast the proxy ARP packets. If the subnet does not exist on an interface, proxy ARP will be broadcast out of all the interfaces.

Important note

Starting from PAN-OS 10.2.8, 11.0.5 and 11.1.0 this behavior was changed; the firewall will now apply strict proxy ARP enforcement for source NAT, which means it will only send out proxy ARP for IP addresses in the same subnet as the ingress interface.

For destination NAT, the behavior remains the same.

Let's take a look at some of the common, and a few special, use cases – NAT rule configurations.

Hide NAT or one-to-many NAT

The most common implementation of outbound NAT is the infamous *hide NAT*, or many-to-one, which changes the source IP addresses of all internal clients to the external IP(s) of the firewall. It is best to place this rule near the bottom of the rule base as it will catch any non-specific sessions and rewrite the source IP to that of the firewall.

The best option for this type of NAT is **Dynamic IP and Port** (**DIPP**). DIPP rewrites the source IP to that of a selected interface or a manually entered IP, IP range, or subnet, and assigns a random source port to the session on egress, as you can see here:

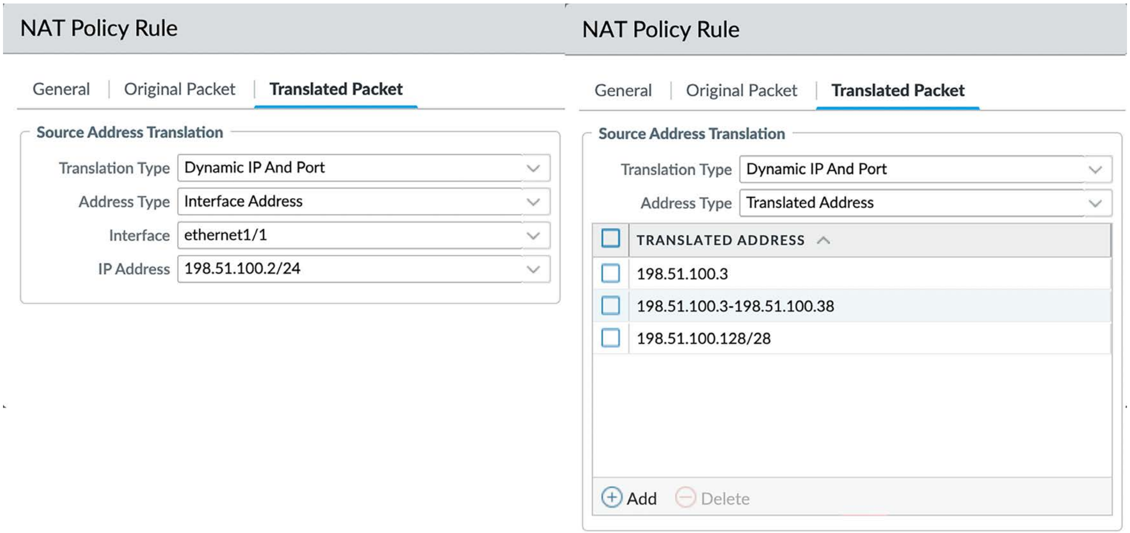


Figure 3.51: DIPP to an interface IP or manual selection

DIPP supports around 64,000 concurrent sessions per available source IP, multiplied by the oversubscription factor supported by the platform you are deploying these rules on. As a rule of thumb, small platforms commonly support 2x oversubscription, large platforms support 4x, and extra-large platforms up to 8x. When multiple IPs are available, DIPP assigns a rewrite IP based on a hash of the source IP so that the same source always gets the same translation address. Once the concurrent allowance for a given translation address is depleted, new sessions will be blocked until existing sessions are freed up.

You can check the current oversubscription ratio by using the following command:

```
admin@PA-220> show running nat-rule-ippool rule <rule name>
VSYS 1 Rule <rule name>:
Rule: <rule name>, Pool index: 1, memory usage: 20344
-----------------------------------------
Oversubscription Ratio:              2
Number of Allocates:                 0
Last Allocated Index:                0
```

Starting from PAN-OS 11.1.1 and later, **Persistent Dynamic IP and Port** (**PDIPP**) allows you to bind the private source IPs to a specific translated IP so all sessions from the client are translated behind the same post-NAT IP. This helps tremendously with applications like STUN that open multiple subsequent connections and require all the connections to be sourced from the same IP.

If more than 64,000 multiplied by the oversubscription factor concurrent sessions per source are needed, or source ports need to be maintained, you can opt to use Dynamic IP instead of DIPP. Dynamic IP will simply "hop" to the next available IP in its assigned translation addresses for a given source IP while maintaining the source port.

As a fallback, if the available IP pool does get depleted because Dynamic IP does not support oversubscription, you can enable DIPP. The IP used in the fallback should not overlap with any of the main IP pools:

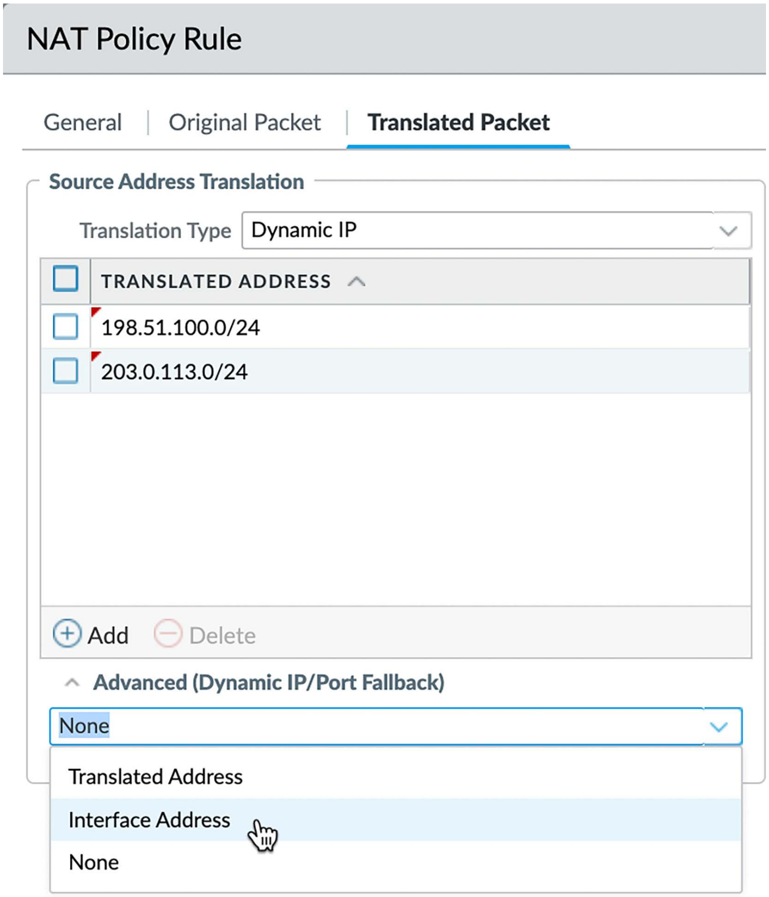


Figure 3.52: Dynamic IP with two subnets and DIPP fallback

In some cases, a server or host on the network will need to "own" its own IP address, which can be achieved with one-to-one NAT rules.

One-to-one NAT

A static IP will always translate a source into the same translation IP and maintain the source port. An IP range can be set, in which case the source IPs will be sequentially matched to the translated IPs, but it is important that the source range and translation range are identical in size; for example, `10.0.0.5-10.0.0.15` translates to `203.0.113.5-203.0.113.115`.

The **Bi-directional** option creates an *implied* inbound NAT rule to allow inbound translation for the same source/translated source pairs. This implied rule reuses the destination zone set in the rule and **any** as the new source zone. It will set the translated address as the new destination of the original source to the new translated destination.

For the outbound rule, as you can see in *Figure 3.53* you have the following:

- **Source:** `Trust-L3`
- **Destination:** `Untrust-L3`
- **Original source:** `serverfarm`
- **Translated source:** `serverfarm-public`

For rules that have a bi-directional IP set, the following implied NAT rule will be created. This rule will *not* be visible in the rule base so be very attentive to any rules that are set to bi-directional and what this implies: the source is set to **any** so the implied rule may catch unintended sessions from other zones than the ones you configured:

- **Source:** `any`
- **Destination:** `Untrust-L3`
- **Original destination:** `serverfarm-public`
- **Translated destination:** `serverfarm`

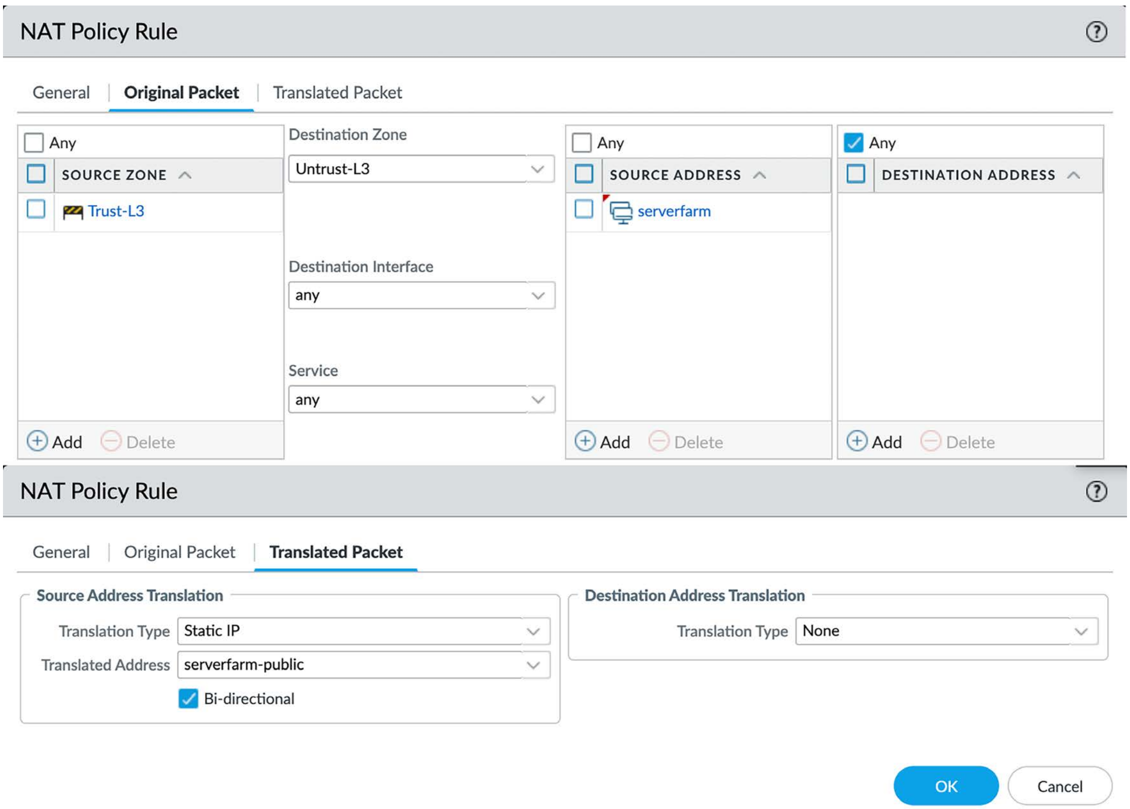


Figure 3.53: Static IP NAT with the Bi-directional option

In some cases, "double NAT" needs to be applied to sessions that need to take an unusual route due to NAT. These types of NAT rules are called U-turn or hairpin NAT rules.

U-turn or hairpin NAT

If an internal host needs to connect to another internal host on the same subnet, by using its public IP address, a unique problem presents itself. For each session, only one NAT rule can be matched. When the client connects to the public IP, the routing table will want to send the packet out to the internet, which will trigger the hide NAT rule, which translates the source IP. (It will not trigger the normal destination NAT rule, as this will have **Untrust** as the source zone.)

The packet should then go back inside as the destination public IP is also owned by the firewall, but a second NAT action can't be triggered, so the packet is discarded.

If the hide-NAT IP is identical to the destination IP, which is common in environments with few public IP addresses, the packet will be registered as a land attack:

```
admin@PA-220> show counter global | match land
Global counters:
Elapsed time since last sampling: 26.05  seconds
name     value  rate severity category  aspect  description
-------------------------------------------------------------
Flow_parse_land 1 1  drop     flow      parse    Packets dropped: land attack
```

A workaround to this problem, if changing the internal DNS record so the FQDN resolves to an internal IP while on the local network or adding an entry to the host file of the client is not possible, is to configure a U-turn or hairpin NAT.

> **Important note**
>
> If you are using PAN-OS 9.0.2 or later, take a look at an alternative solution in the *Enable DNS Rewrite* section.

This type of NAT combines the destination and source NAT and must be placed at the top of the rule base to prevent the hide NAT rule from catching these outbound sessions. The reason the source NAT is required is to make the session stick to the firewall so that no asymmetric routes are created.

If you were to configure the destination NAT to rewrite the public IP for the internal IP without translating the source, the server would receive a packet with the original source IP intact and reply directly to the client, bypassing the firewall. The next packet from the client would be sent to the firewall, which would try to perform TCP session sanity checks and determine whether the TCP session was broken, discarding the client packet.

Adding source translation would force the server to reply to the firewall, which would then forward the translated packet back to the client:

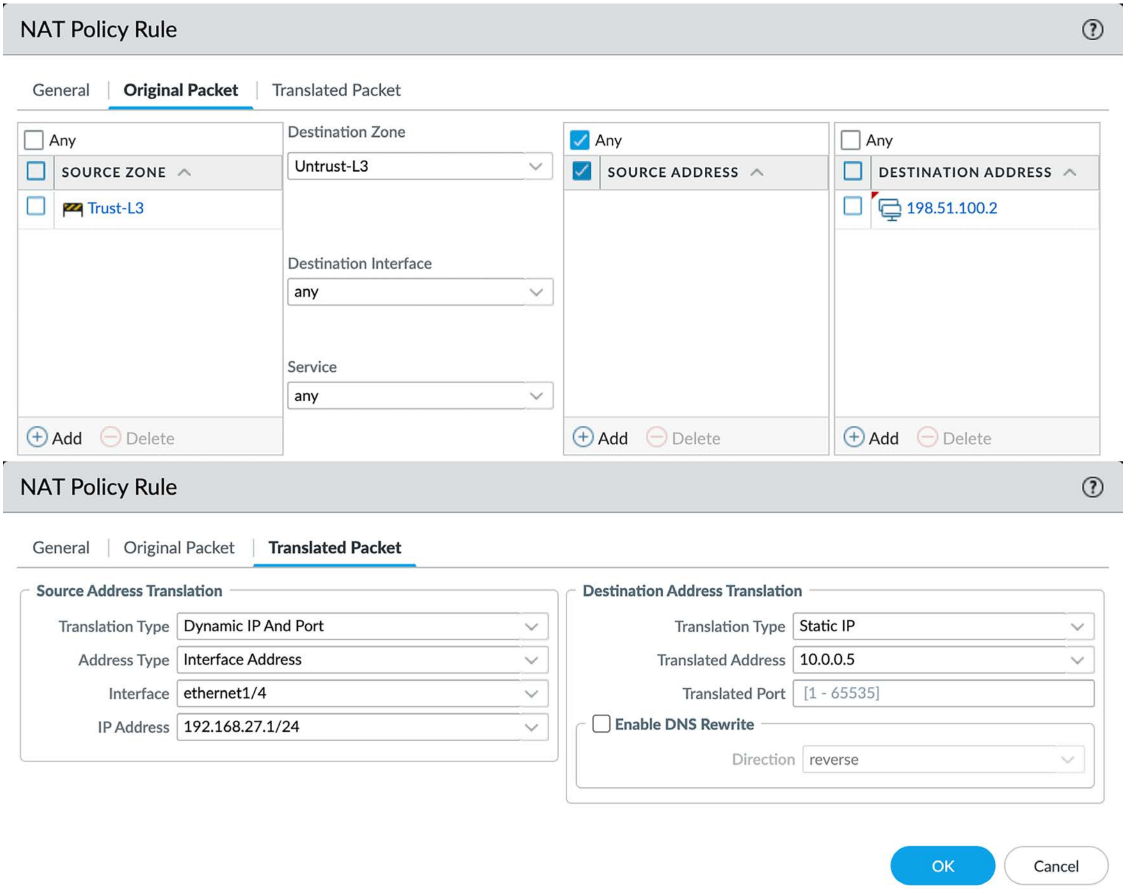


Figure 3.54: U-turn NAT

This type of complication can also be addressed by changing the DNS query to the internal IP of the final destination.

Enable DNS Rewrite

The **Enable DNS Rewrite** option was introduced in PAN-OS 9.0.2 and later, and enables the NAT policy to be applied inside DNS response packets:

- It reverse translates the DNS response that matches the *translated* destination address in the rule. If the NAT rule rewrites `198.51.100.2` to `10.0.0.5`, the reverse rewrite will change the DNS response of `10.0.0.5` to `198.51.100.2`.
- It forward translates the DNS response that matches the *original* destination address in the rule. The forward DNS rewrite changes the DNS response of `198.51.100.2` to `10.0.0.5`.

This could be useful in a scenario where internal hosts need to query a DNS server in the DMZ for an FQDN of a server also hosted in a DMZ where they receive the external IP in the DNS response. This could lead to odd routing issues (see the *U-turn or hairpin NAT* section) as the destination IP will match the external zone, but both the client and server are on internal zones:

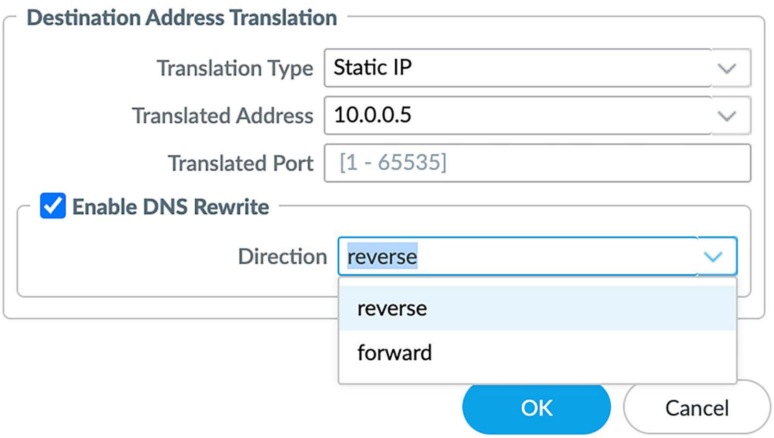


Figure 3.55: Enable DNS Rewrite

If a service is hosted on several physical servers (the original destination is an FQDN that returns several IP addresses), the destination translation settings can be set to **Dynamic IP** (**with session distribution**). The firewall will rewrite the destination IP according to the chosen method:

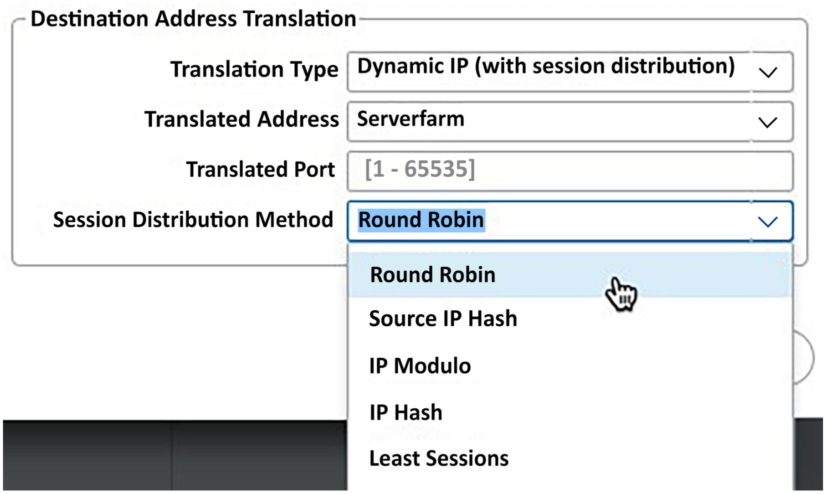


Figure 3.56: Dynamic IP (with session distribution)

This is how these session redistribution methods work:

- **Round Robin** assigns a new session an IP in rotating order.
- **Source IP Hash** assigns new sessions an IP based on a hash of the source IP. The same source IP will always connect to the same destination, so if there's a proxy, for example, or a NAT gateway between the clients and the firewall, do not use this method.
- **IP Modulo** assigns new sessions an IP based on the source and destination IP, performing an XOR and a modulo operation.
- **IP Hash** assigns new sessions using a hash of the source and destination IP.
- **Least Sessions** assigns new sessions to the IP addresses that have the least number of active concurrent sessions in the session table.

With this information, you will now be able to resolve any NAT challenges you may face.

Summary

In this chapter, you learned how to create security profiles and how to build a set of profiles that influence how your firewall processes threats. You learned how to create security profiles that leverage best practices and add these to a default security profile group so that your security rule base starts off with a strong protection stance. You are also able to create complete security rules that leverage reusable objects, have easy-to-identify tags, and are set to allow all desirable access based on application identification rather than ports. You can now make complex NAT policies that cater to the needs of your inbound and outbound connections.

If you're studying for the PCNSE, take specific note of how the best practice security profiles are set with **reset-both** and **single-packet** packet capture for **critical**, **high**, and **medium** severity, while **low** and **informational** are set to **default** with no packet capture. Remember how zones play an important role in the **Original Packet** tab of NAT rules. Remember the importance and implications of App-ID with **application-default** and how application filters can be used to define behavior rather than needing to account for all applications manually.

In the next chapter, we will see how to take even more control of your sessions by leveraging policy-based routing to segregate business-critical sessions from the general internet, limit bandwidth-hogging applications with quality of service, and look inside encrypted sessions with SSL decryption.

4

Taking Control of Sessions

In this chapter, you will see how you can ensure business-critical or latency-sensitive applications do not run out of bandwidth and less important applications are prevented from consuming too much. You will learn how to bypass the routing table and make exceptions for certain sessions, as well as how to decrypt encrypted sessions and look within them to determine actual applications and stop threats.

In this chapter, we're going to cover the following main topics:

- Controlling the bandwidth with quality-of-service policies
- Leveraging SSL decryption to look inside encrypted sessions
- Redirecting sessions over different paths using policy-based forwarding

By the end of this chapter, you will have learned how to look inside encrypted TLS sessions so threats can be stopped, manipulate how sessions are forwarded regardless of the routing table, and make the best of your available bandwidth.

Technical requirements

This chapter requires a working knowledge of measuring network bandwidth and available resources. You should understand the implications of sending packets over different interfaces rather than where routes are pointing to, and more specifically, the paths reply packets may take that could introduce asymmetric routing. You should also have a good understanding of certificate chains.

Controlling the bandwidth with quality-of-service policies

Quality of Service (**QoS**) is the collective name for several technologies that can help improve the quality of applications, and the data flows that they are applied to, by prioritizing them over other flows or reserving bandwidth to ensure adequate throughput and acceptable latency. In this section, you will learn how QoS marking can be applied to a firewall to interact with network devices downstream.

There are two ways for a firewall to participate in applying QoS to network traffic:

- **Differentiated Services Code Point** (**DSCP**) and **Type of Service** (**ToS**) headers, which are "external" markings introduced by a network device or host and intended to be carried all the way to the final destination or until the header is stripped, ensuring all devices in the path are aware of the weight or priority of a packet
- Built-in capabilities, which do not alter the header of the packets and take place by internally prioritizing or slowing down packets flowing through the firewall

Let's review external headers first.

DSCP and ToS headers

DSCP headers allow the firewall to let upstream and downstream devices know that certain sessions have a certain priority. These headers can be set in the security policies in the **Actions** tab, as shown in the following screenshot:

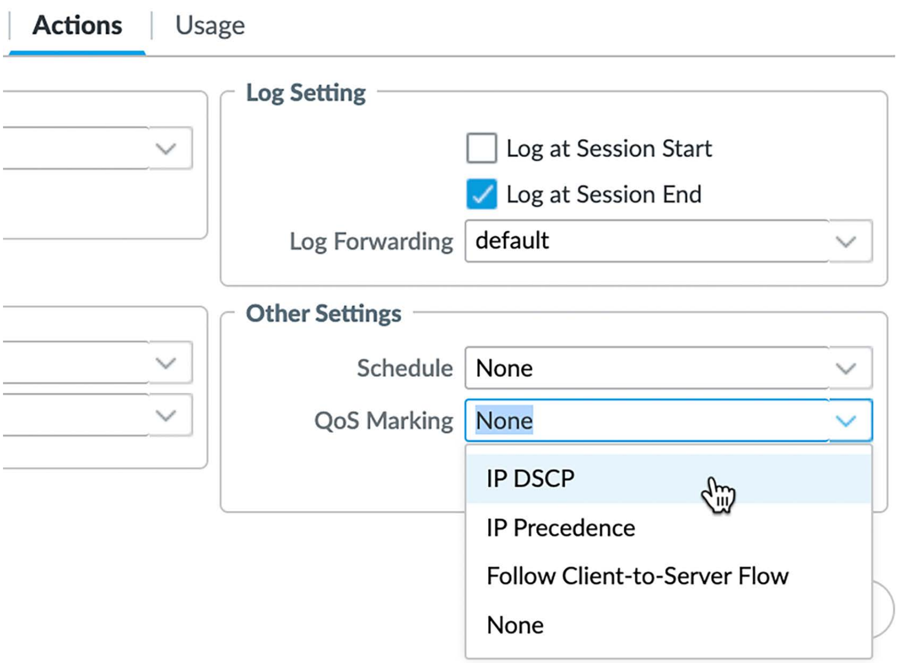


Figure 4.1: QoS marking options

In DSCP, you can set **Assured Forwarding** (**AF**) from af11 to af43, **Expedited Forwarding** (**EF**), or **Class Selector** (**CS**) from cs0 to cs7 code points. The **IP Precedence** (cs0 to cs7) ToS can be used when communicating with legacy network devices and **Follow Client-to-Server Flow** can be used to apply inbound DSCP marking to a returning outbound flow.

In the next section, we will cover controlling flows directly in the firewall.

QoS enforcement in the firewall

The firewall can also enforce bandwidth restrictions or guarantees, and that's what we will focus on here. The Palo Alto Networks firewall uses a system of eight classes combined with policies. A class is used to identify flows that belong to a specific QoS profile, and the number of the class has no bearing on what actions it performs. You can choose to make Class 1 the crown jewels or you can choose to make it the "garbage bucket" – the choice is yours. The only exception is Class 4, which will be the default class for anything that's not been assigned a specific class.

Each interface is set up with a QoS profile that mandates how each class is treated, and then policies are created to identify sessions as belonging to a certain class. The default class is Class 4, so anything that is not caught by a QoS rule will automatically become Class 4 and be subject to the restrictions for that class.

We'll use the following topology to illustrate an example QoS policy. Map out your own network throughput so you can apply the following example scenario to your own environment:

- An internet link on `eth1/1` with a download bandwidth of 200 Mbps per second and an upload bandwidth of 50 Mbps
- A DMZ network containing some servers on `eth1/2` connected to a 1 Gbps interface
- A LAN where the users sit on `eth1/3` connected to a 1 Gbps interface
- Users need 20 Mbps of guaranteed upload and download bandwidth for their enterprise **Voice over Internet Protocol** (**VoIP**), but some internet downloads need to be limited to 50 Mbps
- Fileshare traffic between users and servers needs to be limited to 300 Mbps
- Site-to-site VPN connections need a 20 Mbps guarantee for business-critical applications

This topology is illustrated as follows:

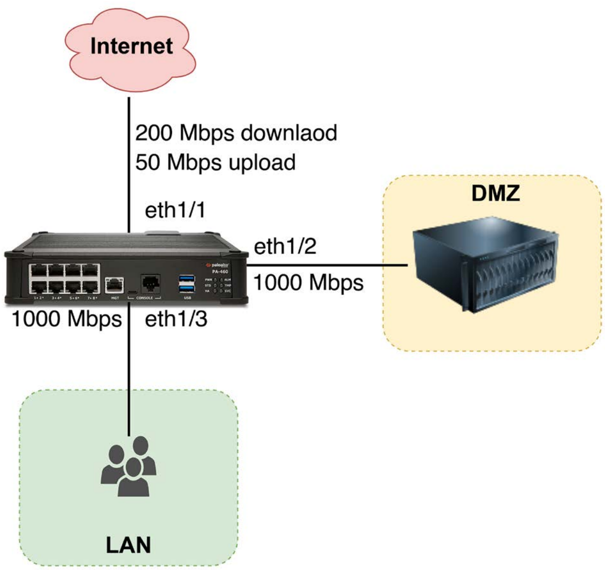


Figure 4.2: Example topology

Next, we will start laying down the groundwork for what will eventually become QoS enforcement.

Creating QoS profiles

Go to **Network** | **Network Profiles** | **QoS Profile**; you need to create at least one new profile to get started. The classes themselves do not carry any weight, so Class 1 could be your most important class, but also your lowest, depending on how you configure its parameters.

The **priority** setting does require special consideration; the **real-time** priority has its own queue in packet processing, making sure that any packets that end up in the queue (due to bandwidth congestion) go out first. All the lower priorities (high to low) share the main queue, with the lowest-priority packets being discarded first if packets need to be let go in favor of higher-priority sessions.

As shown in *Figure 4.3*, the **Egress Max** option at the top of the profile refers to the total of the maximum and reserved bandwidths for the whole profile, while the **Egress Max** column next to the class indicates how much bandwidth all of the sessions in that class get to share.

Let's create a few profiles first:

1. Create a profile called `internet-upload`.
2. Set the profile's **Egress Max** value to `50` Mbps to limit the total bandwidth usable by the profile to 50 Mbps. This tells the QoS engine that it needs to use its queuing mechanism and prioritize packets once it reaches the maximum limit.
3. Create `class1`, set it to **real-time**, and set a guarantee of `20` Mbps. This will ensure that the sessions we select later as our Class 1 applications will have a guaranteed bandwidth of 20 Mbps available to them.

> This profile can also be created with the following commands in the **Command-Line Interface** (**CLI**):
>
> ```
> reaper@pa-440# set network qos profile internet-upload
> aggregate-bandwidth egress-max 50
> reaper@pa-440# set network qos profile internet-upload
> class-bandwidth-type mbps class class1 priority real-time
> class-bandwidth egress-guaranteed 20
> ```

4. Create a profile called `internet-download`.
5. Set the profile's **Egress Max** value to `200` Mbps.
6. Create `class1`, set **Priority** to **real-time**, and set its guarantee to `20` Mbps.

7. Create `class5` and set the **Egress Max** value to `50`.

> This profile can also be created with the following commands:
>
> ```
> reaper@pa-440# set network qos profile internet-download
> aggregate-bandwidth egress-max 200
> reaper@pa-440# set network qos profile internet-download
> class-bandwidth-type mbps class class5 priority medium
> class-bandwidth egress-max 50
> reaper@pa-440# set network qos profile internet-download
> class-bandwidth-type mbps class class1 priority real-time
> class-bandwidth egress-guaranteed 20
> ```

8. Create a profile called `internal`.
9. Do not set this profile's **Egress Max** value; we will be mixing this profile with the internet one, so we will let the interface maximum egress determine the maximum for this profile.
10. Create `class8`, set it to **low** priority, and set **Egress Max** to `300`.

> `internal` can also be created in the CLI as follows:
>
> ```
> reaper@pa-440# set network qos profile internal class-
> bandwidth-type mbps class class8 priority low class-
> bandwidth egress-max 300
> ```

11. Create a profile called `vpn`.
12. Create `class4` and set it to guarantee `20` Mbps and to **real-time** priority; for this profile, we will let IPSec connections default to `class4`.

> `vpn` can be created in the CLI as follows:
>
> ```
> reaper@pa-440# set network qos profile vpn class-bandwidth-
> type mbps class class4 priority medium class-bandwidth
> egress-guaranteed 20
> ```

The QoS profiles should look as follows:

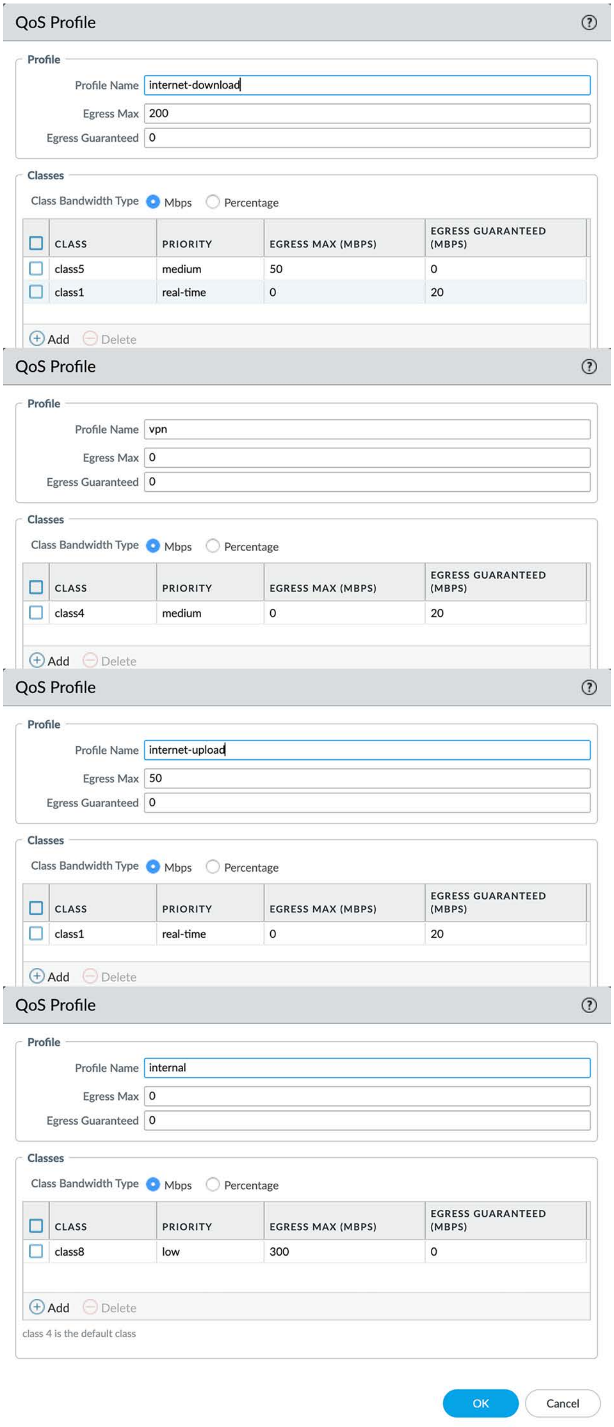


Figure 4.3: QoS profiles

Next, the interfaces need to be set to enforce QoS. In **Network** | **QoS**, add all the interfaces. Then, for `ethernet1/1`, the internet-facing interface, do the following, as illustrated in the following screenshot, or by executing the following CLI commands:

1. Check the **Turn on QoS feature on this interface** box.

```
reaper@pa-440# set network qos interface ethernet1/1 enabled yes
```

2. Set the interface **Egress Max** value to `50` Mbps to limit uploads to the internet.

```
reaper@pa-440# set network qos interface ethernet1/1 interface-bandwidth
egress-max 50
```

3. Set the `internet-upload` profile as a **Clear Text** profile so that classes can be applied.

```
reaper@pa-440# set network qos interface ethernet1/1 regular-traffic
default-group qos-profile internet-upload
```

4. Set the `vpn` profile as the **Tunnel Interface** profile.

```
reaper@pa-440# set network qos interface ethernet1/1 tunnel-traffic
default-group per-tunnel-qos-profile vpn
```

This applies QoS to any site-to-site VPN connections sourced from the firewall to a remote peer (on a local tunnel interface):

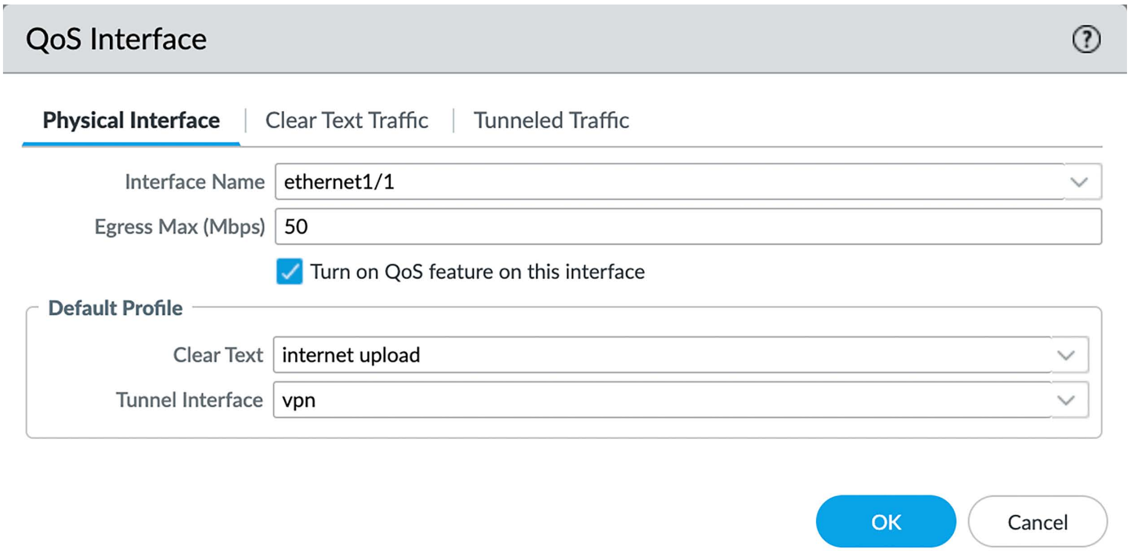


Figure 4.4: eth1/1 QoS configuration

For `ethernet1/2`, the DMZ-facing interface, do the following, as illustrated in the following screenshot, or by using the following CLI commands:

1. Check the **Turn on QoS feature on this interface** box.

```
reaper@pa-440# set network qos interface ethernet1/2 enabled yes
```

2. Set the interface **Egress Max** value to `1000` Mbps, but leave **Clear Text** as `default` and **Tunnel Interface** as none.

```
reaper@pa-440# set network qos interface ethernet1/2 interface-bandwidth
egress-max 1000
reaper@pa-440# set network qos interface ethernet1/2 regular-traffic
default-group qos-profile default
```

3. In the **Clear Text** tab, set the **Egress Max** value to `1000` Mbps.

```
reaper@pa-440# set network qos interface ethernet1/2 regular-traffic
bandwidth egress-max 1000
```

4. Add a new profile line:

   - Call it `userupload`
   - Assign the `internal` QoS profile
   - Set the source interface to `ethernet1/3`

```
reaper@pa-440# set network qos interface ethernet1/2 regular-traffic
groups regular-traffic-group members userupload match local-address
address any
reaper@pa-440# set network qos interface ethernet1/2 regular-traffic
groups regular-traffic-group members userupload match local-address
interface ethernet1/3
reaper@pa-440# set network qos interface ethernet1/2 regular-traffic
groups regular-traffic-group members userupload qos-profile internal
```

5. Add a second profile line:

   - Call it `internet`
   - Assign the `internet-download` profile
   - Set the source interface to `ethernet1/1`

```
reaper@pa-440# set network qos interface ethernet1/2 regular-traffic
groups regular-traffic-group members internet match local-address address
any
reaper@pa-440# set network qos interface ethernet1/2 regular-traffic
groups regular-traffic-group members internet match local-address
interface ethernet1/1
```

```
reaper@pa-440# set network qos interface ethernet1/2 regular-traffic
groups regular-traffic-group members internet qos-profile internet-
download
```

These settings allow different profiles to be applied, as you can see in the following screenshot, depending on where the packets originate from. Downloads from the internet will be limited to `200` Mbps in total, and `class5` can be applied to limit sessions to `50` Mbps as needed, while sessions from the user's LAN can use up to `1000` Mbps and limit the bandwidth to `300` Mbps uploads for the `class8` sessions:

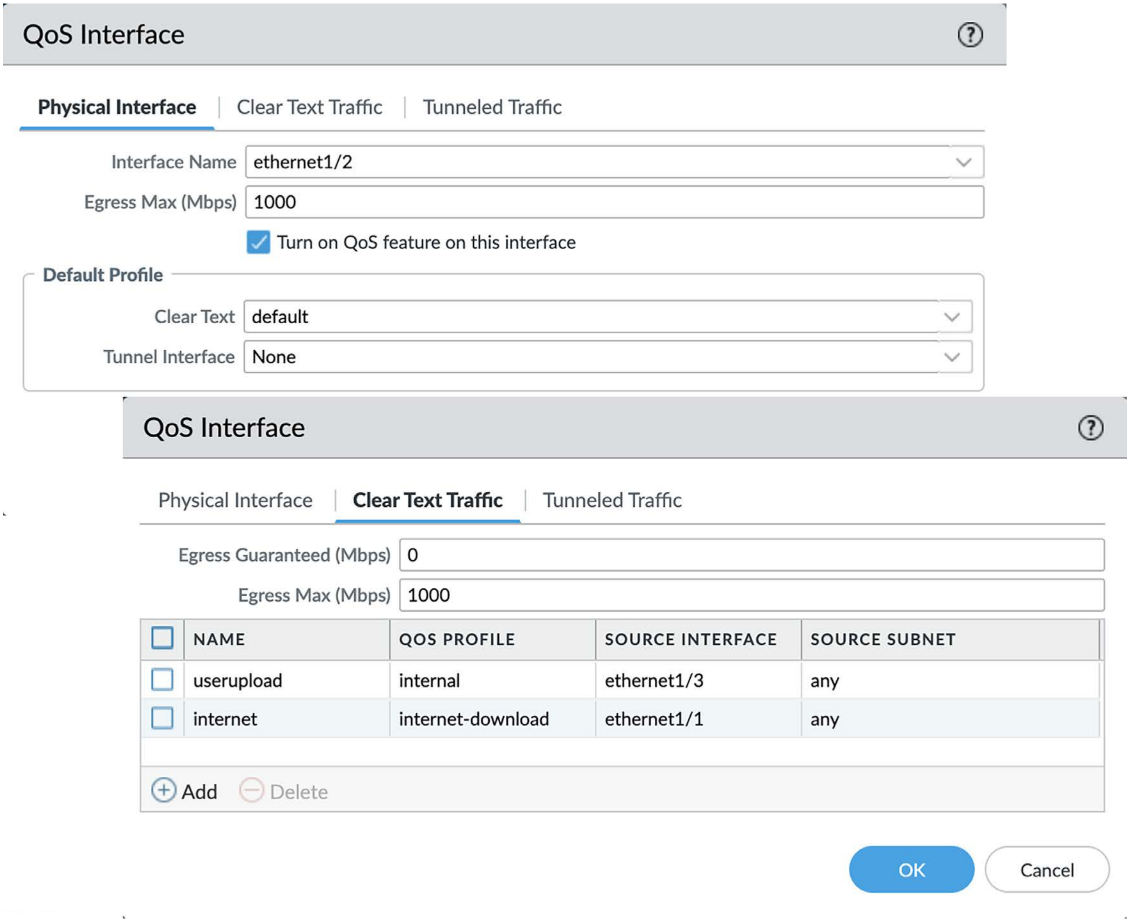


Figure 4.5: eth1/2 QoS configuration

For `ethernet1/3`, the user-facing interface, do the following, as illustrated in the following screenshot, or by using the following CLI commands.

1. Check the **Turn on QoS feature on this interface** box.

```
reaper@pa-440# set network qos interface ethernet1/3 enabled yes
```

2. Set the interface's **Egress Max** value to `1000` Mbps, but leave **Clear Text** as `default` and **Tunnel Interface** as none.

```
reaper@pa-440# set network qos interface ethernet1/3 interface-bandwidth
egress-max 1000
reaper@pa-440# set network qos interface ethernet1/3 regular-traffic
default-group qos-profile default
```

3. In the **Clear Text** tab, set **Egress Max** to `1000` Mbps.

```
reaper@pa-440# set network qos interface ethernet1/3 regular-traffic
bandwidth egress-max 1000
```

4. Add a new profile line:

   - Call it `userdownload`
   - Assign the `internal` QoS profile
   - Set the source interface to `ethernet1/2`

```
reaper@pa-440# set network qos interface ethernet1/3 regular-traffic
groups regular-traffic-group members userdownload match local-address
address any
reaper@pa-440# set network qos interface ethernet1/3 regular-traffic
groups regular-traffic-group members userdownload match local-address
interface ethernet1/2
reaper@pa-440# set network qos interface ethernet1/3 regular-traffic
groups regular-traffic-group members userdownload qos-profile internal
```

5. Add a second profile line:

   - Call it `internetdownload`
   - Assign the `internet-download` profile
   - Set the source interface to `ethernet1/1`

```
reaper@pa-440# set network qos interface ethernet1/3 regular-traffic
groups regular-traffic-group members internetdownload match local-address
address any
reaper@pa-440# set network qos interface ethernet1/3 regular-traffic
groups regular-traffic-group members internetdownload match local-address
interface ethernet1/1
reaper@pa-440# set network qos interface ethernet1/3 regular-traffic
groups regular-traffic-group members internetdownload qos-profile
internet-download
```

These settings will limit the maximum Mbps when downloading (or streaming) things from the internet while guaranteeing that the `class1` sessions are not deprived of bandwidth and that the bandwidth from the DMZ server is also maximized for all of the sessions to 1 Gbps, except `class8`, which is limited to `300` Mbps downloads. This should look as follows:

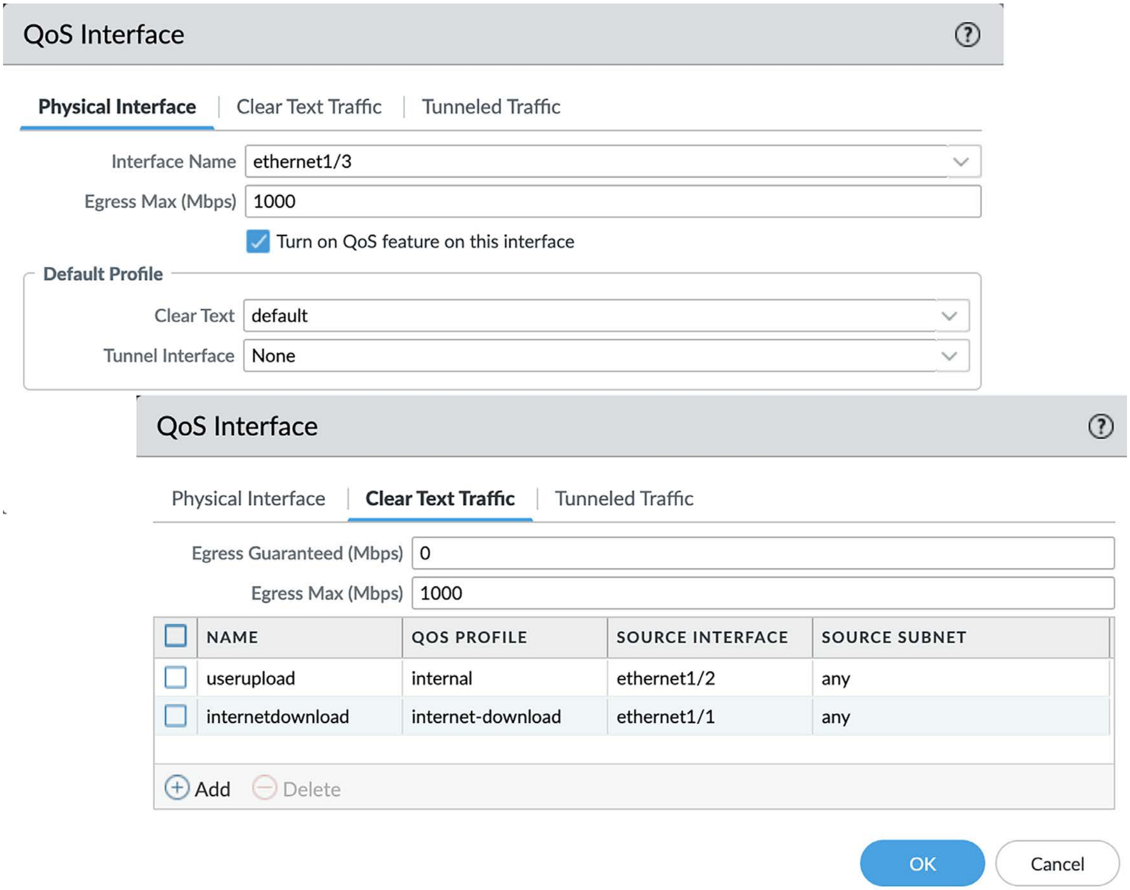


Figure 4.6: eth1/3 QoS configuration

We have now created a framework that can apply traffic shaping to sessions. These profiles can now be used in QoS policies that actually apply the enforcement and classification of applications in traffic flows. They can be mixed and matched as needed, as we will see in the following section.

Creating QoS policies

Without any QoS rules, only `class4` will be enforced, which, in the previous case, will only set **Egress Max** to the maximum internet speed, but with no guarantees. The first policy we need to set will define **enterprise VoIP** as `class1` so that we can guarantee 20 Mbps downloads over the internet link:

> The following example uses zone names and an application filter we created in the previous chapter. If you skipped these, go and create them or adjust the parameters in the following examples to match your environment.

1. Create a new rule by going to **Policies** | **QoS**.
2. Call the rule **enterprise voip**.

3. Set the zone(s) to the Trust-L3 and DMZ-L3 zones so that outbound calls are classified as **class1**.
4. Set the destination zone where the sessions will egress the firewall.
5. Set the class to **class1**:

```
reaper@pa-440> configure

[edit]
reaper@pa-440# set rulebase qos rules "enterprise voip" from [ DMZ-L3
Trust-L3 ] to Untrust-L3 source any destination any category any
application "enterprise voip" service application-default action class 1
```

Your policy should look similar to the following:

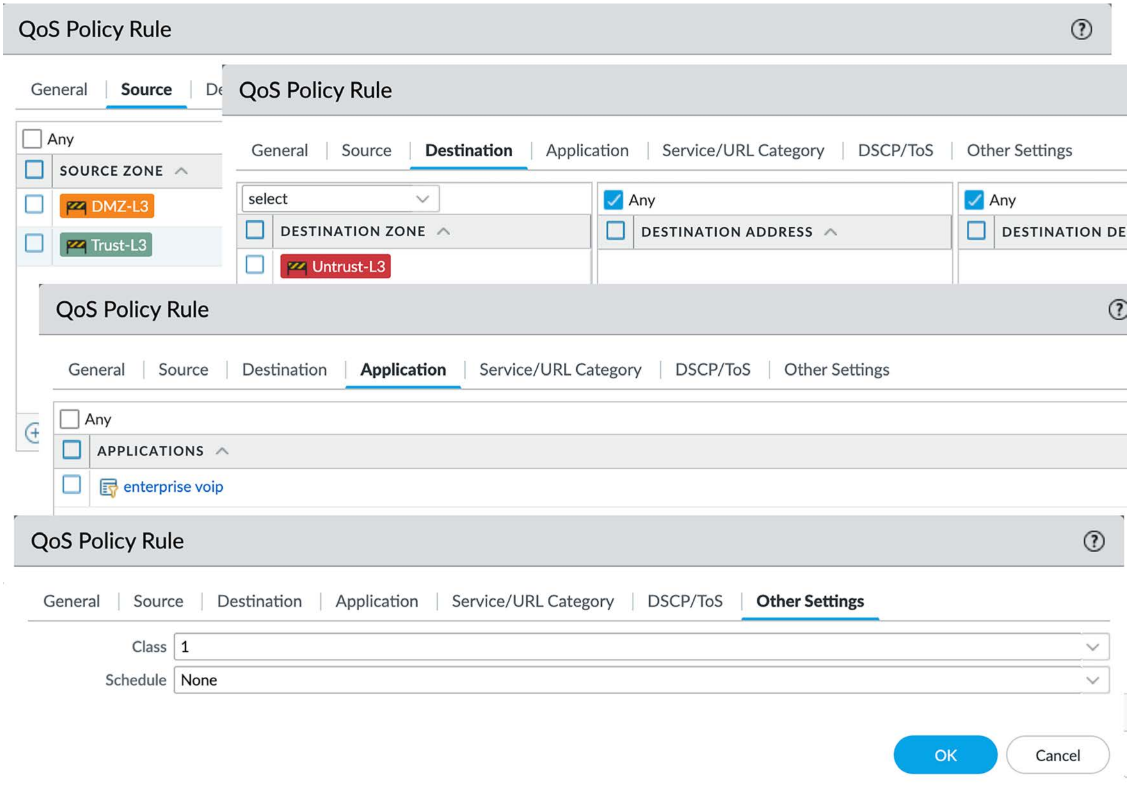


Figure 4.7: Setting VoIP to Class 1 outbound

The second rule sets the same guarantee, but for sessions that are started from the internet (such as an inbound SIP call). Follow these steps to create an inbound rule (if inbound sessions are not allowed by the security policy, you can skip this rule):

1. Create a rule and call it `enterprise voip in`.
2. Set the source zone to the `Untrust-L3` zone.

3. Set the destination zone to the internal zones where calls can be accepted (the internal client or DMZ gateway).
4. Set the class to `class 1`.

```
reaper@pa-440> configure

[edit]
reaper@pa-440# set rulebase qos rules "enterprise voip inbound" to [
DMZ-L3 Trust-L3 ] from Untrust-L3 source any destination any category any
"enterprise voip" action class 1
```

The inbound rule will look as follows:

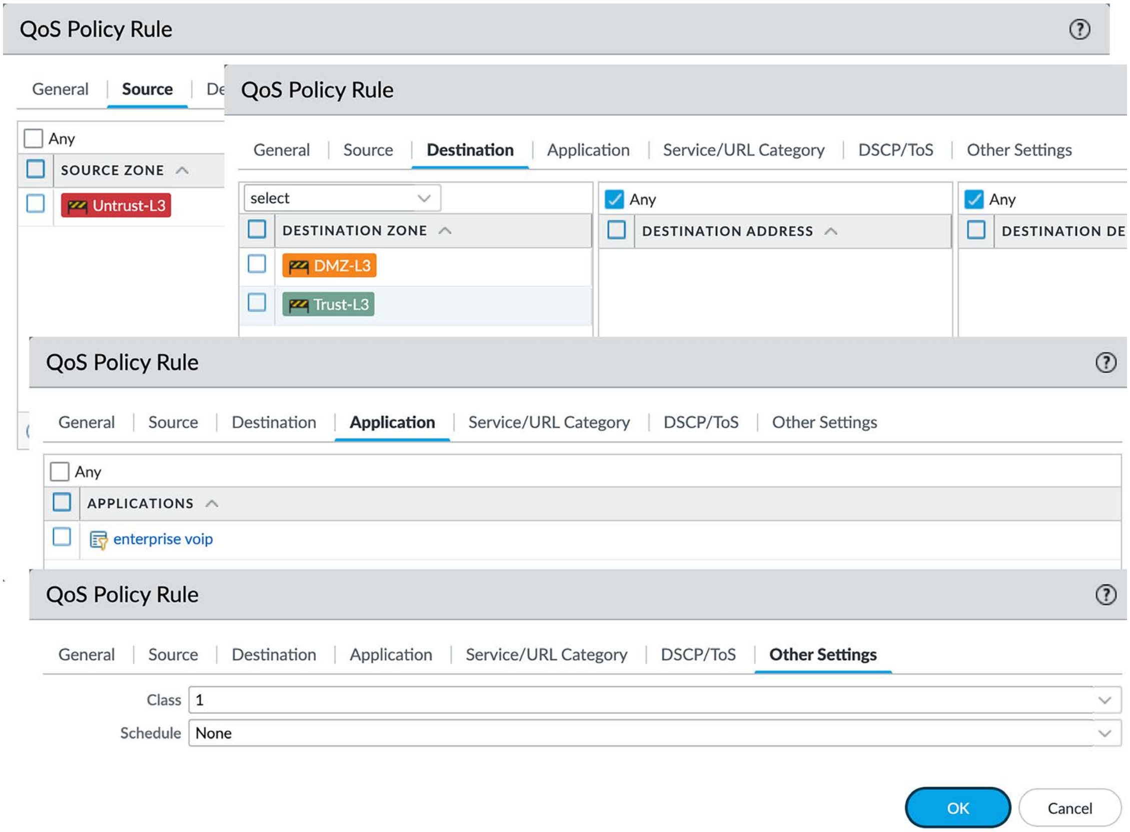


Figure 4.8: Setting VoIP to Class 1 inbound

We will also need to limit certain sessions between the user's LAN and DMZ networks. Assuming the security policy only allows users to connect to the DMZ and no sessions to be allowed from the DMZ to the user network, only one QoS rule will be needed as QoS classes are assigned to all packets in a session, regardless of their direction (so, `class8` will be applied in both directions even if you only have your QoS rule set in one direction).

Follow these steps to create an internal QoS rule:

1. Create a new QoS rule and call it `fileshares`.
2. Set the source zone to the `Trust-L3` network.
3. Set the destination zone to the DMZ network.
4. Add the appropriate filesharing applications.
5. Set the class to `class 8`.

```
reaper@pa-440# set rulebase qos rules fileshares from Trust-L3 to DMZ-L3
source any destination any application [ ftp ms-ds-smb scps ] service
application-default action class 8
```

6. Save the changes.

Your internal rule will look as follows:

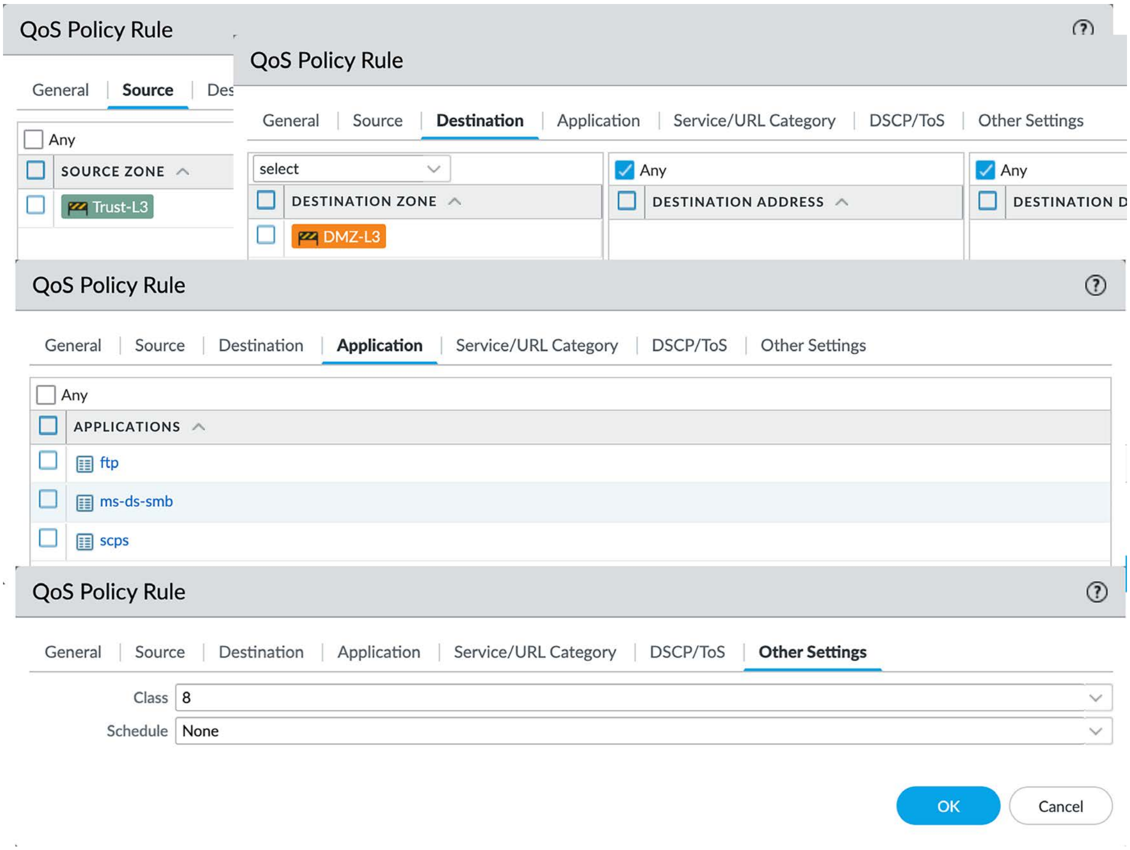


Figure 4.9: Setting file transfer applications to Class 8

To quickly check whether the limitations and guarantees are being enforced properly, you can access a live graph next to each enabled interface from **Network** | **QoS** | **Statistics:**

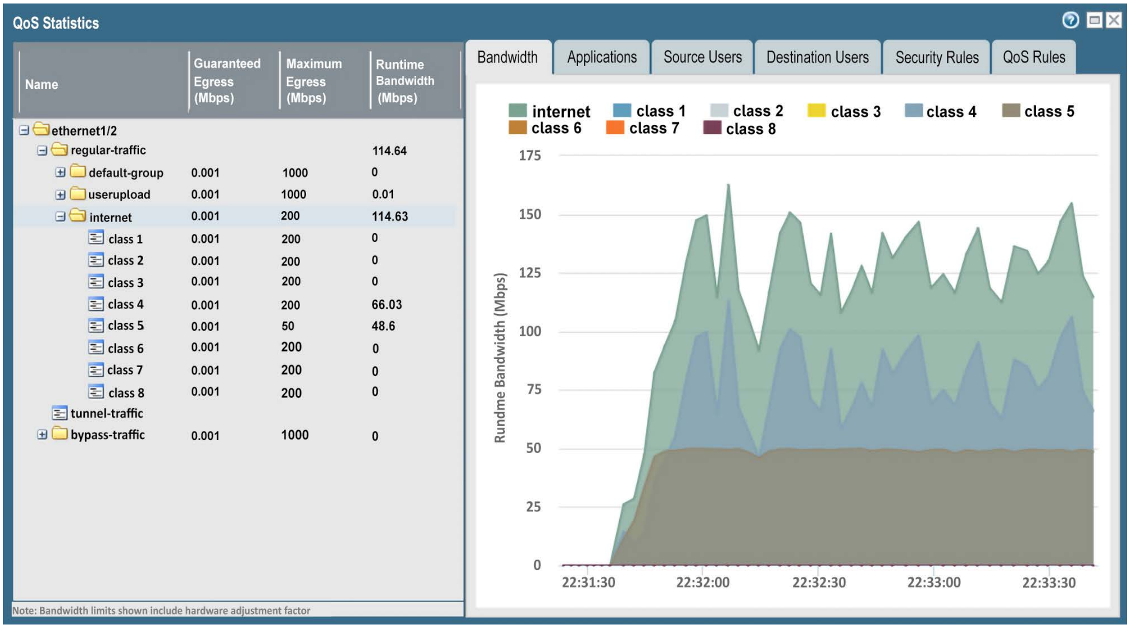


Figure 4.10: Class 5 is limited to 50 Mbps

To recap, we have learned the following:

- QoS is applied to the egress interface.
- Bandwidth restrictions and guarantees are shared within a class, not per session.
- Real-time priority has its own queue; the others share a queue.
- **Egress Guaranteed** or **Egress Max** cannot exceed the interface maximum.
- `Class 4` is the default class.
- Classes may have different guarantees or limitations, depending on the direction of the packet.
- If a guarantee in a class is not filled, other classes may consume more bandwidth (without exceeding their max) until the guarantee is required.
- While we apply bandwidth limitations and guarantees at the interface level, class is assigned at the session level and bandwidth restrictions will be applied differently depending on the egress interface, even for the same class.

You can now create QoS profiles and understand the implications of priorities, guarantees, and the egress maximum. You can apply these profiles to interfaces and define different profiles depending on the source interface. You can also create rule sets that classify applications so that they can be shaped according to your profiles.

In the next section, we will see how encrypted sessions can be decrypted and inspected and how applications within an SSL session can be determined and threats stopped.

Leveraging SSL decryption to look inside encrypted sessions

Secure Sockets Layer/Transport Layer Security (**SSL/TLS**) and **Secure Socket Shell** (**SSH**) decryption perform a man-in-the-middle attack, but for good instead of evil—an encrypted session is intercepted, then, through the use of trusted certificates, the session is deciphered and the payload is made visible for content inspection and App-ID to take a look at. There are three modes of decryption currently available:

- SSH proxy
- SSL forward proxy
- SSL inbound inspection

Let's look at each of them in detail.

SSH proxy

SSH proxy allows the interception of SSH sessions. The SSH connection will be proxied, meaning that the client connects to the firewall and the firewall establishes a new session to the server. This allows you to control (allow or block) tunneling over the SSH session by setting a security policy for the `ssh-tunnel` application.

SSL forward proxy

SSL forward proxy is used for all outbound sessions. There are two distinct directions (inbound and outbound), and the outbound option is proxied because of how certificates are used to sign a website's SSL/TLS certificate. In the world of certificates, a handful of trusted organizations hold "trusted root signing certificates," which are considered to have the same authority as a notary with regard to signing documents. They sign off on a subset of subordinate or intermediary certificates, which are then used to sign off on server certificates, which represent domain names such as `www.google.com`, `pangurus.com`, or `packtpub.com`. This chain of trust needs to be resistant to attack so that "bad actors" can't set up a fake website and dupe visitors into trusting it, which makes legitimate interception difficult. For more details, refer to `https://en.wikipedia.org/wiki/Public_key_certificate`.

When accessing a website on the internet, the root and intermediary certificates can be any of the dozens of available options, so the only way to get the internal client to trust an intercepted connection is to replace the entire chain and make the client trust the root signing certificate. This can be accomplished in several ways, with the most straightforward method being a manual import, but this may be more difficult to accomplish in a large environment. You can also leverage Microsoft **Group Policy Objects** (**GPOs**) and several other deployment methods, toolkits, scripts, and software packages; but whatever you do, don't let your users get used to ignoring a browser certificate warning—that is a dangerous habit to get into! Put in the time to install your organization's root and intermediary certificates, or your self-signed root certificate and intermediary, into the *trusted root signing certificate store* of all your clients' computers (and Firefox, as it uses its own certificate store). It will pay off in the long run, I promise.

If your organization already has a CA set up, you can simply have it create a new, easily identifiable intermediary that can be used for decryption:

1. Export it with its private key and export the root certificate without a private key.
2. Go to **Device** | **Certificate Management** | **Certificates** and import both, starting with the root.

If you do not have a CA available or you want to test the waters before you take a dive, we'll set SSL decryption up with a self-signed certificate for you to play with:

1. Go to **Device** | **Certificate Management** | **Certificates** and generate a root signing certificate by checking the **Certificate Authority** box and calling it `root signing certificate`.
2. Then, create a subordinate certificate by setting **Signed By** as the root signing certificate, checking the **Certificate Authority** box, and calling it `decryption subordinate`.
3. Finally, make a third certificate that is not signed by the root signing certificate, set it as a CA, and name it `untrusted cert`.

You will need one certificate that your users will trust to decrypt websites. You also need an untrusted certificate because, during decryption, the entire certificate structure is replaced with your own. If the real certificate has any problems, the firewall will keep decrypting but will use the untrusted certificate instead, so the user gets a certificate warning in their browser, making them halt and think about continuing.

These are the steps to create all the certificates you need:

1. Create a new certificate and call it `root signing cert`.
2. Set the CA flag.
3. Fill out the attributes and click **Generate**.
4. Create a new certificate and call it `decryption`.
5. Set the CA flag.
6. Set the **Signed By** field to `root signing cert`.
7. Fill out the attributes and click **Generate**.
8. Create a third certificate and name it `untrusted cert`.
9. Set the CA flag.
10. Make sure you do not set this one as signed by the root.

As a minimum, set the **Email** attribute. This will help savvy users who investigate why they received a certificate warning find the relevant contact details.

Your certificates should look similar to the following:

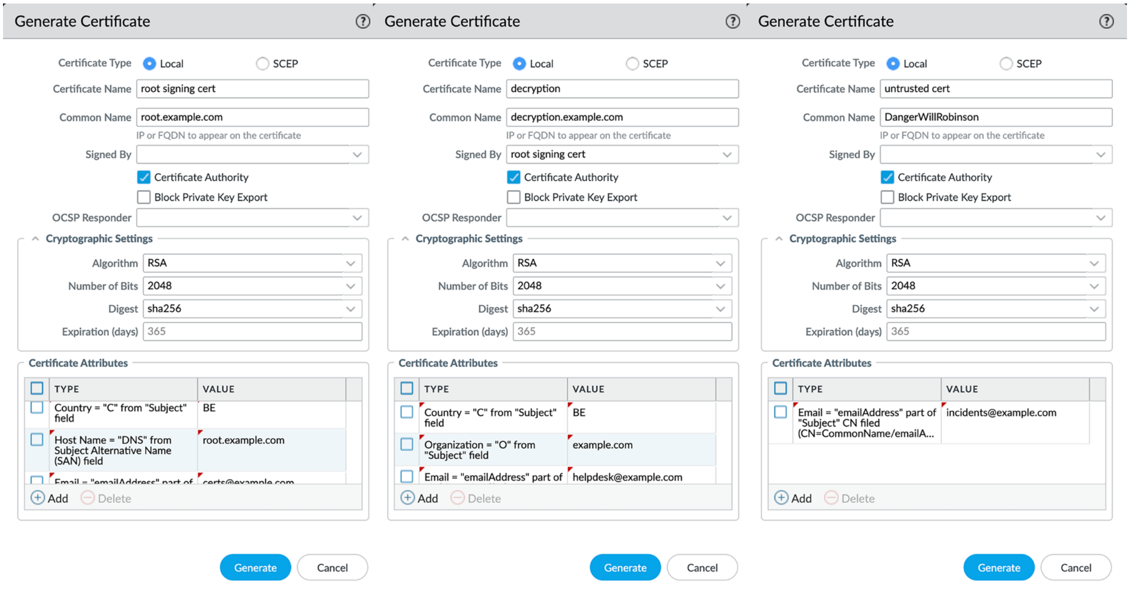


Figure 4.11: The root, decryption, and untrusted certificates

Once the certificates are generated, you can click them to reveal some additional settings. You can select from three different options:

- **Forward Trust Certificate**
- **Forward Untrust Certificate**
- **Trusted Root CA**

Forward Trust Certificate is used for decryption and **Forward Untrust Certificate** is used if there is a problem with the upstream certificate and a warning should go out to users.

If an upstream certificate is problematic or suspicious, using **Forward Trust Certificate** would not prompt the user that there is something up as the firewall takes the responsibility of interacting with the endpoint. Never assign a single certificate to both roles as that would hide certificate issues from the user (the Forward Untrust Certificate should also never be signed by a trusted root certificate).

Trusted Root CA can be set so that the firewall itself trusts the root CA, which comes in handy if, for example, the dynamic update sessions go through the firewall and are decrypted.

These options are shown in the following screenshot:

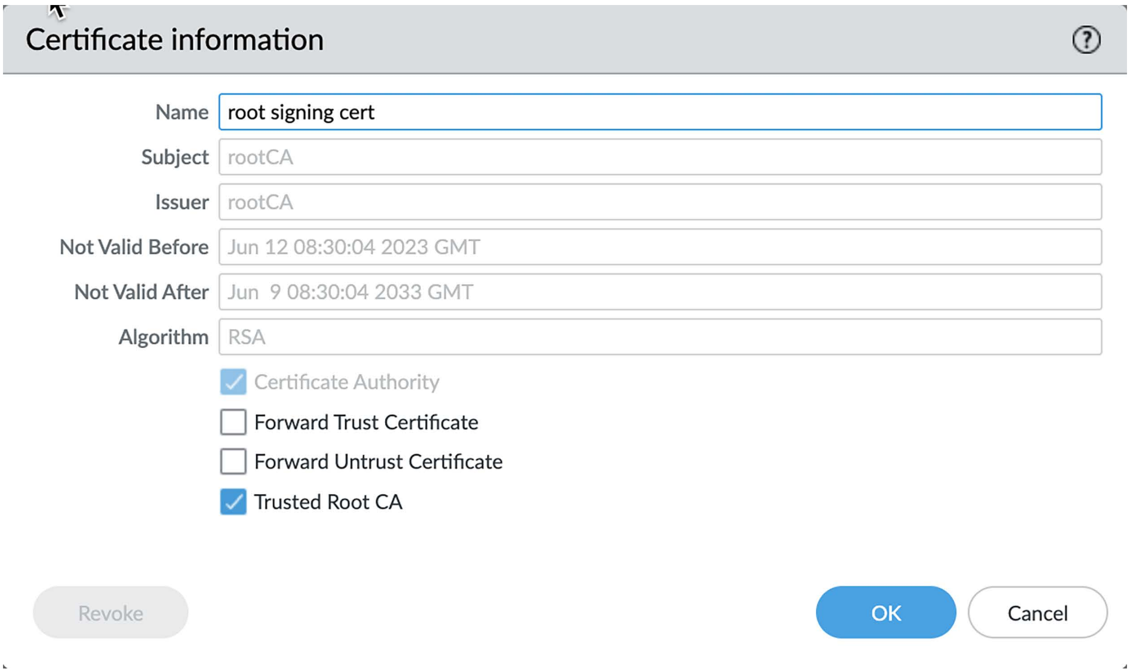


Figure 4.12: Certificate options

Set each of the three certificates to their appropriate roles:

1. Set the root signing certificate as **Trusted Root CA**.
2. Set the decryption subordinate certificate as **Forward Trust Certificate**.
3. Set the untrusted certificate as **Forward Untrust Certificate**.

> *Never* set a single certificate as both the forward trusted and forward untrusted certificate.

You will now need to select the root signing certificate and export it to your computer. When asked whether you want to include the key, select **No** as you do not need it on your endpoints.

As the following screenshot shows, check the box in front of the certificate and click **Export Certificate** at the bottom:

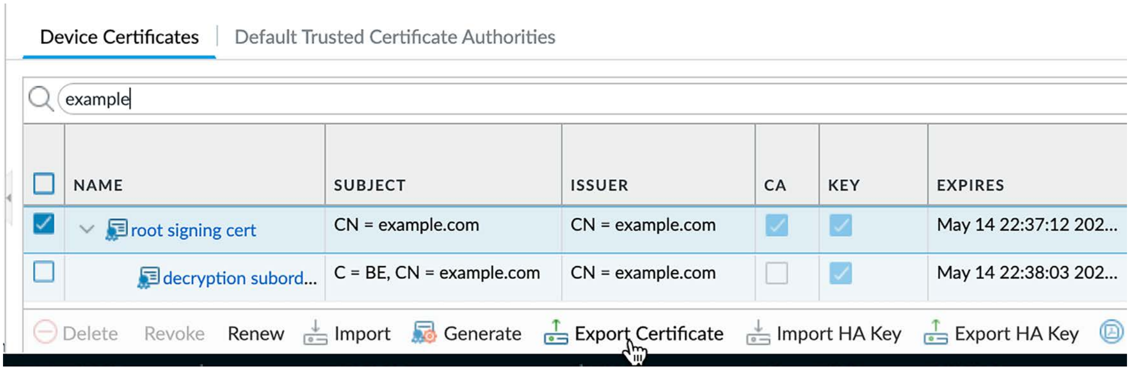


Figure 4.13: Exporting the root signing certificate

Once exported, you need to import the certificate onto your test machine's **trusted root certificate store.** If you intend to use Firefox, remember to add it to Firefox separately as Firefox doesn't use the machine certificate store (Internet Explorer, Edge, Chrome, and Safari do use the machine store).

Next, you need to create a decryption profile by going to **Objects** | **Decryption** | **Decryption profiles**. The default one is a bit weak and we want to ensure that certificate enforcement is a bit more robust:

1. Create a new SSL decryption profile and give it a descriptive name.
2. In the **SSL Forward Proxy** tab, enable Block expired certificates, untrusted issuers, unknown certificate status, certificate status check timeout, unsupported versions, and unsupported cipher suites. Enable any remaining options for even more security.
3. In **SSL Inbound Inspection**, enable **Block sessions with unsupported versions** and **Block sessions with unsupported cipher suites.**
4. In **SSL Protocol settings**, disable `RSA`, `3DES`, `RC4`, and `SHA1` and set the minimum version to `TLS1.2`.
5. In the **No Decryption** tab, set the flags to block expired and untrusted certificates.
6. Finally, block unsupported versions and algorithms on the **SSH Proxy** tab.
7. Click **OK.**

8. Your profile should look similar to the following screenshot:

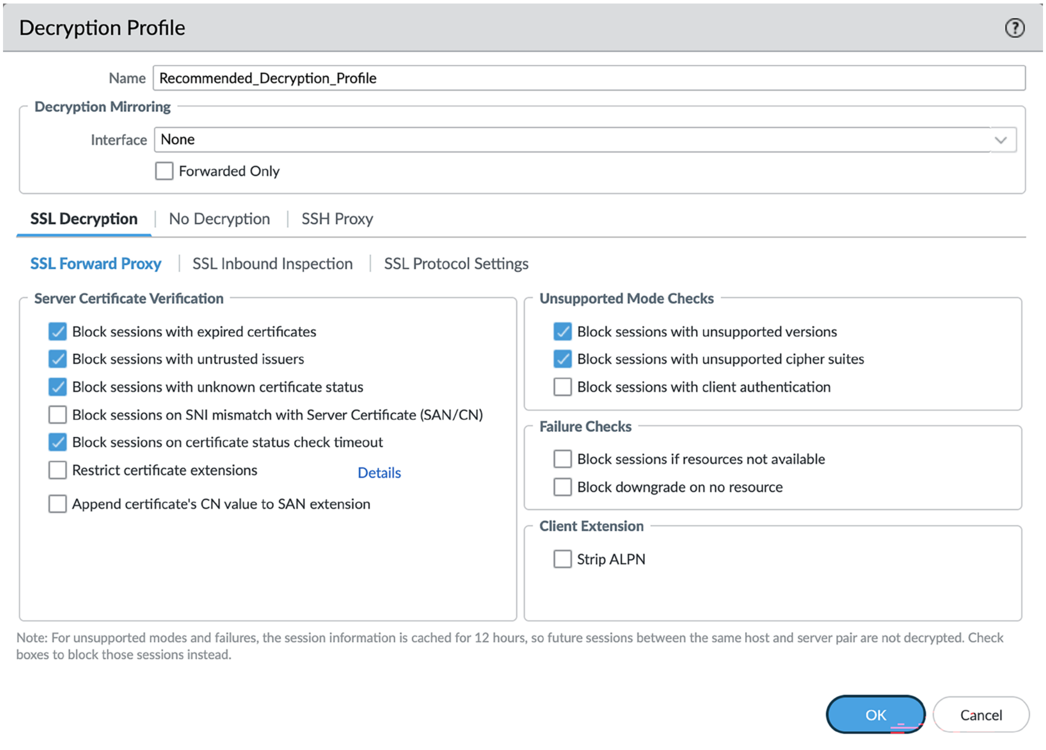


Figure 4.14: Decryption profile

Now that the certificates are loaded and the decryption profile is created, you can go ahead and create the decryption rules by going to **Policy** | **Decryption**.

Building a decryption rule is pretty much the same as building a security rule. There's a source zone and network, a destination zone and network, and a service or URL category (no applications here). However, the options are a little different.

You can choose to perform **No Decrypt**, which comes in handy if you need to account for privacy-sensitive topics, such as online banking or religion.

You will need to build a policy when you need to carefully balance work and private life, which is usually a mixture of local law and company policy. Consider whether your organization will allow certain URL categories to be accessed from company equipment or on the company's network. Also, consider whether decryption should be applied to some personal categories as it may be prohibited by law to inspect certain sessions.

Commonly, some categories are allowed to be accessed but are not decrypted for privacy reasons. These categories should be added to a **No Decrypt** rule and placed at the top of the `decryption` rule base. For everything else, create an **SSL Forward Proxy** rule.

These are the steps to set up your basic decryption policy:

1. Create a new rule and call it `no-decrypt`.
2. Set the source zone to `Trust-L3`.
3. Set the destination zone to `Untrust-L3`.
4. Set the URL categories to `financial-services` or any category (**health-and-medicine**, **government**, etc.) that is accessible but should be treated as private.
5. For the options, set the action to `no-decrypt`, type `SSL Forward Proxy`, and set the decryption profile:

```
reaper@pa-440# set rulebase decryption rules no-decrypt from Trust-L3
source any to Untrust-L3 destination any category financial-services
profile "decryption profile" action no-decrypt type ssl-forward-proxy
```

6. Create a second rule and call it `decryption`.
7. Set the source zone to the `Trust-L3` zone.
8. Set the destination to the `Untrust-L3` zone.
9. Leave the URL categories as `any`.
10. Set the action to `decrypt`, type `SSL Forward Proxy`, and set the decryption profile:

```
reaper@pa-440# set rulebase decryption rules decrypt from Trust-L3 source
any to Untrust-L3 destination any category any profile "decryption
profile" action decrypt type ssl-forward-proxy
```

11. Commit the changes.

When you open a web page now, you should see that the root signing certificate has replaced the original CA:

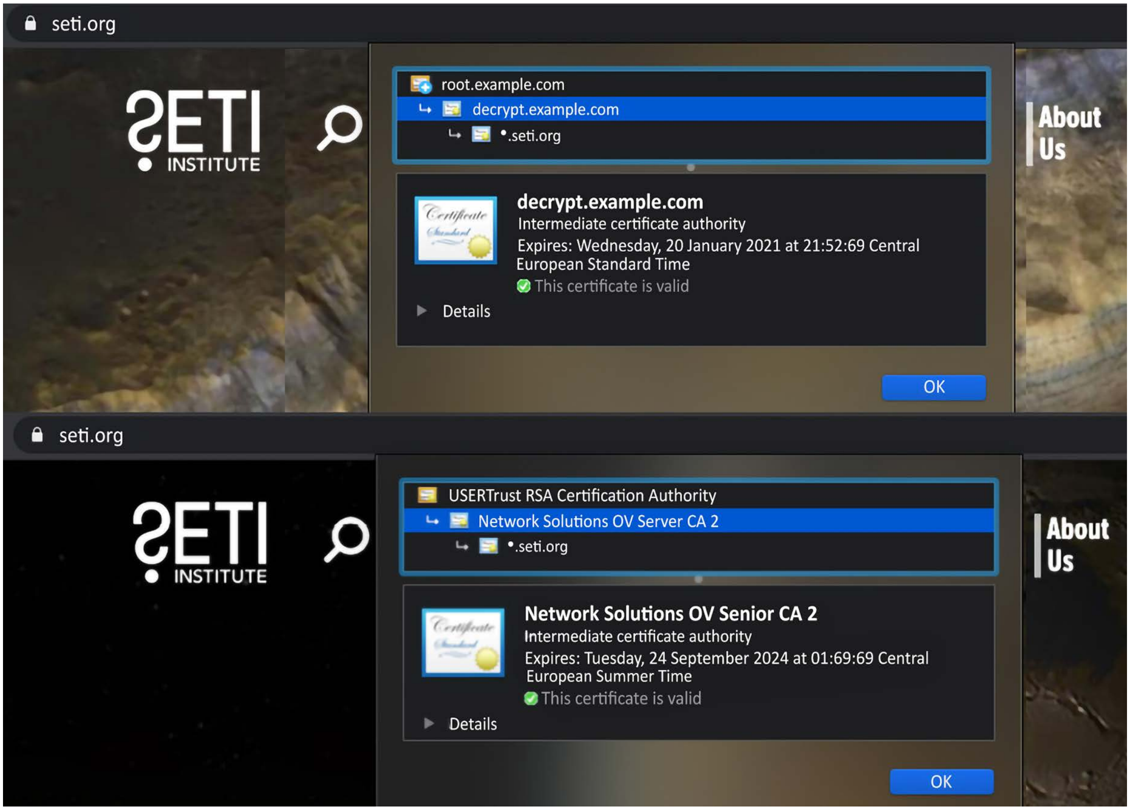


Figure 4.15: Decryption certificate chain versus the original certificate chain

There is one caveat to decrypting outbound connections, and that is that the encrypted session must use regular SSL/TLS as its encryption protocol.

Google developed the Quic transport layer network protocol that could derail decryption efforts as it uses proprietary encryption and relies on UDP over port 443 to transport sessions. As more sites and browsers start to support this protocol, this may prevent visibility in user activities and an inability to block threats. Therefore it is recommended and best practice to set a **Deny** rule at the top of your rule base to prevent the use of Quic and also block **UDP/80** and **UDP/443**, as illustrated in the following screenshot:

	Source		Destination							
NAME	ZONE	ADDRESS	ZONE	ADDRESS	APPLICATION	SERVICE	URL CATEGORY	ACTION	PROFILE	OPTIONS
block quic ports	trust	~~any~~	untrust	any	any	udp80 udp443	any	Deny		
block quic	trust	any	untrust	any	quic	application-default	any	Deny		

Figure 4.16: Blocking the Quic application

You're now able to set up the certificates needed for SSL decryption and build a decryption policy. In the next section, we'll set up inbound decryption for sites hosted in your environment.

SSL Inbound Inspection

SSL Inbound Inspection is used when the web application is hosted locally and you have access to the server certificate including its private key.

You will need to import the server certificate, including its private key, the certificate authority's intermediary, and the root certificate (you don't need the private keys of these last two; they simply serve to complete the chain). You can import certificates by navigating to **Device** | **Certificate management** | **Certificates** and clicking the import button. As the following screenshot shows, you need to import the certificate and the private key files:

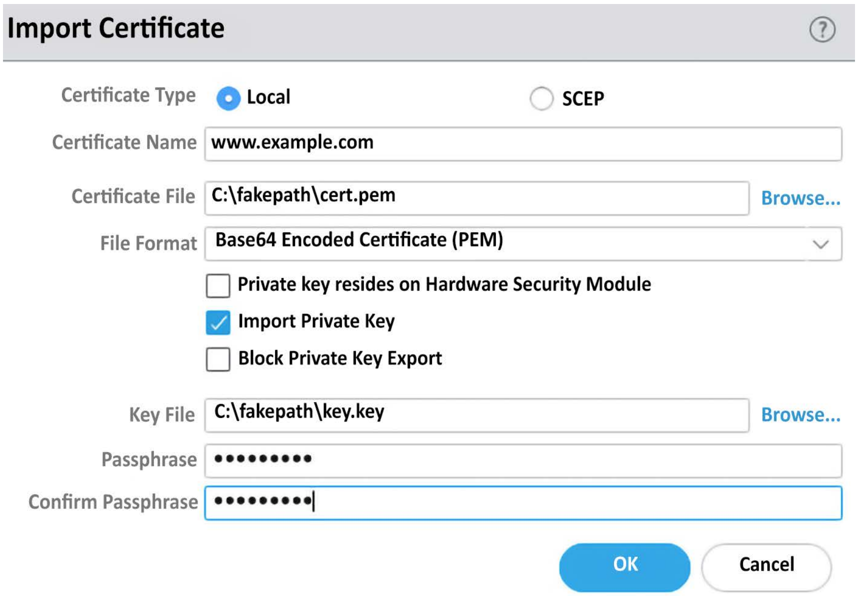


Figure 4.17: Importing a server certificate with a private key

When the chain has been imported, your certificate page should look similar to this:

Figure 4.18: Full certificate chain for your server certificate

Once you have imported the certificate chain, you can create the following policy:

1. Create a new decrypt rule and name it after your domain name or server.
2. Set the source zone to `Untrust-L3`.
3. Set the destination zone to `DMZ-L3` and the destination IP to your server `public` (pre-NAT) IP.

4. Leave the service and URL category as **any** for now.
5. Set the action to `decrypt`, type in `SSL Inbound Inspection`, set the certificate to your server certificate, and enable the decryption profile:

```
reaper@pa-440# set rulebase decryption rules examplecom from Untrust-L3
to DMZ-L3 destination 198.51.100.5 category any profile "decryption
profile"  action decrypt type ssl-inbound-inspection exampledotcom
```

Because the firewall has the server certificate and its private key, it can decrypt in real time; no proxying is required.

Forwarding sessions to an external device

It may be interesting to forward decrypted session information to another scanning device for additional security, forensics gathering, or compliance requirements. There are two licenses available that will allow decrypted sessions to be forwarded to another security device. Both licenses are free but come with different requirements:

- **Decryption Port Mirror** can be applied on all firewalls and will simply send decrypted sessions out of the interface (we saw the Decryption Port Mirror at the end of *Chapter 2*, *Setting up a New Device*) if the decryption profile is set to forward sessions matching the profile. The drawback is that the receiving device(s) will only be able to detect, not prevent, any threats. Once the license is installed and the firewall rebooted, this setting can be enabled in the decryption profile in **Objects** | **Decryption** | **Decryption Profile** by selecting an interface that I set to Type **Decryption Mirror**. As shown in the following screenshot, the **Forwarded Only** option will, when enabled, only mirror decrypted traffic after traffic is forwarded through the firewall. This can be useful in cases when traffic is also forwarded to external threat inspection. The default setting will mirror traffic even when the final security policy is a drop action.

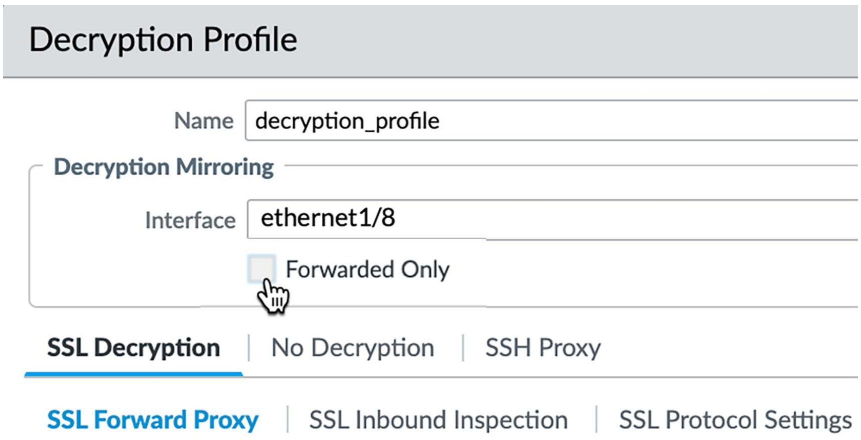


Figure 4.19: Decryption Mirroring interface

- **Decryption Broker** is also a free license that can be activated the same way as the Decryption Port Mirror, but can only be enabled on the PA-7000, PA-5400, PA-5200, PA-3400, PA-3200, VM700, and VM300 series firewalls. If this license is active, two interfaces will be used to communicate with a security chain. The firewall will decrypt the traffic and inspect the content, and then forward the clear text packets to the next device in the security chain. The last device in the security chain will send the clear text packet back to the firewall that will re-encrypt and forward it to the final destination. There are two modes the Decryption Broker can be deployed in:
    - **Layer 3 security chain**: Each device has an assigned IP and static routing is used to redirect packets to the next member.
    - **Transparent bridge security chain**: Devices are serially connected to pass packets from one to the next.

You can now set up SSL decryption for both your users and your hosted environment and choose which categories to exclude or include. You are also able to prevent (or allow) tunneling over SSH sessions. In the next section, we'll learn about changing how sessions are sent from the firewall.

Redirecting sessions over different paths using policy-based forwarding

Policy-Based Forwarding (**PBF**) allows you to set up rules that let certain sessions bypass routing entirely. In the first stage of packet processing, a session can be sent over a different interface than what the routing table would normally dictate. This could be handy if you want to send certain sessions over a secondary ISP link (or leased line) or if you need to ensure packets go out on a specific VLAN, tunnel, or SD-WAN interface. We'll be going over a few common use cases in the following sections as PBF can be applied in several different ways.

Redirecting critical traffic

A common scenario is a small office with a cheap but unreliable DSL or cable uplink with high bandwidth for internet traffic and a reliable but expensive link for business-critical applications. While the default route in the virtual router directs all traffic out of the DSL or cable model, a PBF rule could redirect critical protocols, such as SAP and SQL, over your leased line. This will ensure your important applications are using a stable connection while less important applications use the less reliable uplink.

To create such a rule, follow these steps:

1. Go to **Policies** | **Policy Based Forwarding** to create a new rule, and call it `redirect critical apps to ISP2`.
2. For the source, set your `Trust-L3` network and subnet.
3. For the destination, set the destination address/subnet or the FQDN that hosts critical applications. Don't set applications if you don't have to; use service ports if appropriate.

4. From **Forwarding**, select the new egress interface and the next hop. The next hop could be a router IP, or none if you simply want to put traffic onto a VLAN or into a tunnel interface. If you are adding a next hop, add a monitoring profile and set it to **failover**, then check **Disable this rule if nexthop/monitor ip is unreachable** so that your critical applications are routed over the regular link if your dedicated line goes down.
5. The resulting rule will look like the following screenshot:

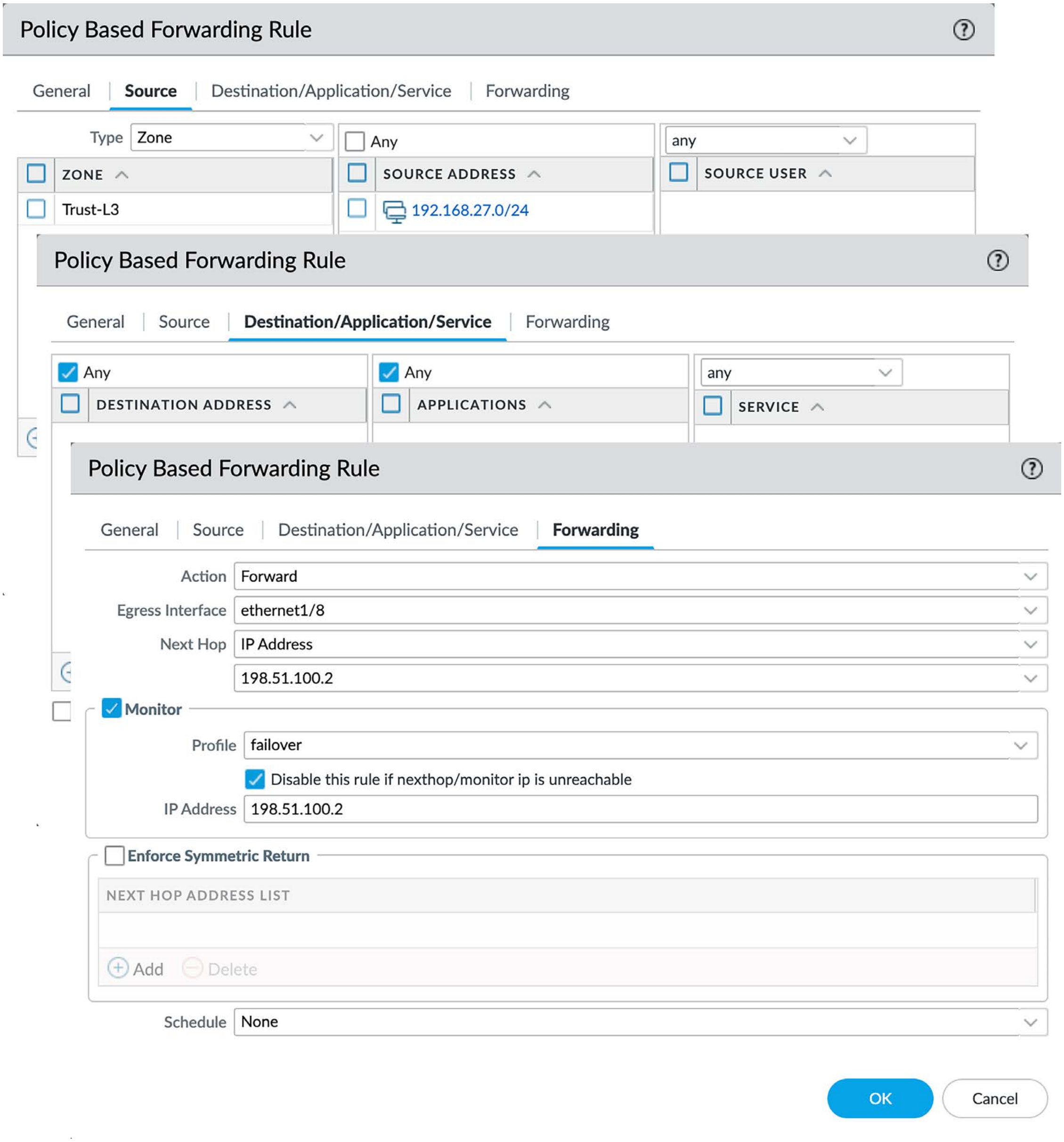


Figure 4.20: Policy Based Forwarding Rule

The rule can also be created using the following CLI commands:

```
reaper@pa-440# set network profiles monitor-profile failover action fail-over
interval 2 threshold 3
reaper@pa-440# set rulebase pbf rules "redirect critical apps to ISP2" from
zone Trust-L3
reaper@pa-440# set rulebase pbf rules "redirect critical apps to ISP2" source
192.168.27.0/24 destination any service service-https action forward monitor
profile failover disable-if-unreachable yes ip-address 198.51.100.2
reaper@pa-440# set rulebase pbf rules "redirect critical apps to ISP2" action
forward egress-interface ethernet1/8 nexthop ip-address 198.51.100.2
```

It is preferable to *not set an application* in the **Destination/Application/Service** tabs for uncommon sessions (for example, web browsing to different destinations). Stick to service ports and destination IPs instead as identifying an application takes a few packets; the first few packets cannot go through an app-based PBF rule and will follow the routing table instead. Recurring connections will be stored in `app-cache` and can hit the PBF on the first packet. The caveat here is that the first session must be able to go through regular routing before the App-ID and associated tuples can be cached in `app-cache`.

You can now redirect important outbound sessions out of a different interface than the default route. In the next section, we will learn how to leverage multiple uplinks for inbound connections.

Load balancing

There may be cases where multiple ISP uplinks are available and can be used for different purposes. In the following sections, we look at a few technologies and designs to help you understand the available options.

Policy based forwarding

Let's look at a common scenario first – say there are two or more uplinks and both are used to provide services (such as an internally hosted website or email server) to internet users. The default route could cause return packets to leave out of a different interface than the interface that they came in through, causing asymmetric routing and failed sessions for the client.

This is illustrated in the following diagram:

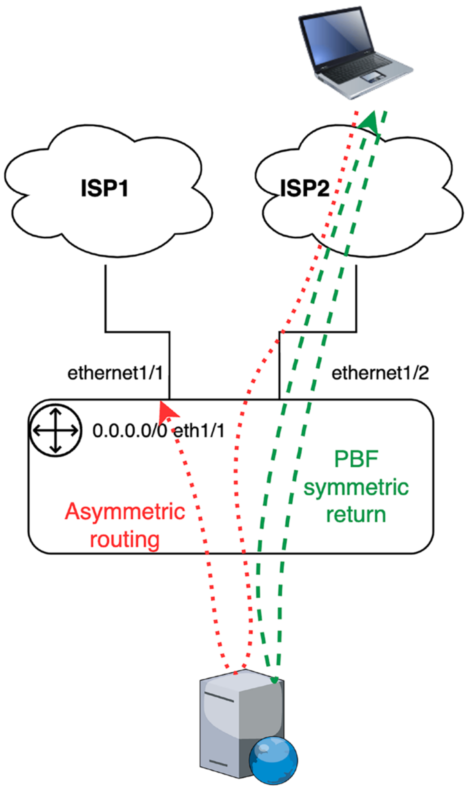


Figure 4.21: Regular routing versus policy based forwarding

PBF can be used to enforce symmetric return, redirecting reply packets to the original interface they came in through, even if the routing table would have sent them elsewhere. These are the steps to set this up:

1. Set the source zone to the ISP sessions that they come in from.
2. Set the destination IP to your server and the appropriate application and service port.
3. The **Forward** action sends packets out of the DMZ interface directly to the mail server, which is what regular routing would achieve.
4. If additional paths are available, monitoring can be enabled and set to disable this PBF rule in case the monitored IP becomes available. The firewall will then start using the next best match in the PBF policies, or default to regular routing.

5. Enabling the **Enforce Symmetric Return** option ensures that reply packets are sent out to the secondary ISP's router instead of using the default route (to `ISP1`). The next hop address will be the IP for ISP2's router.

The PBF rule should now look similar to the following screenshot:

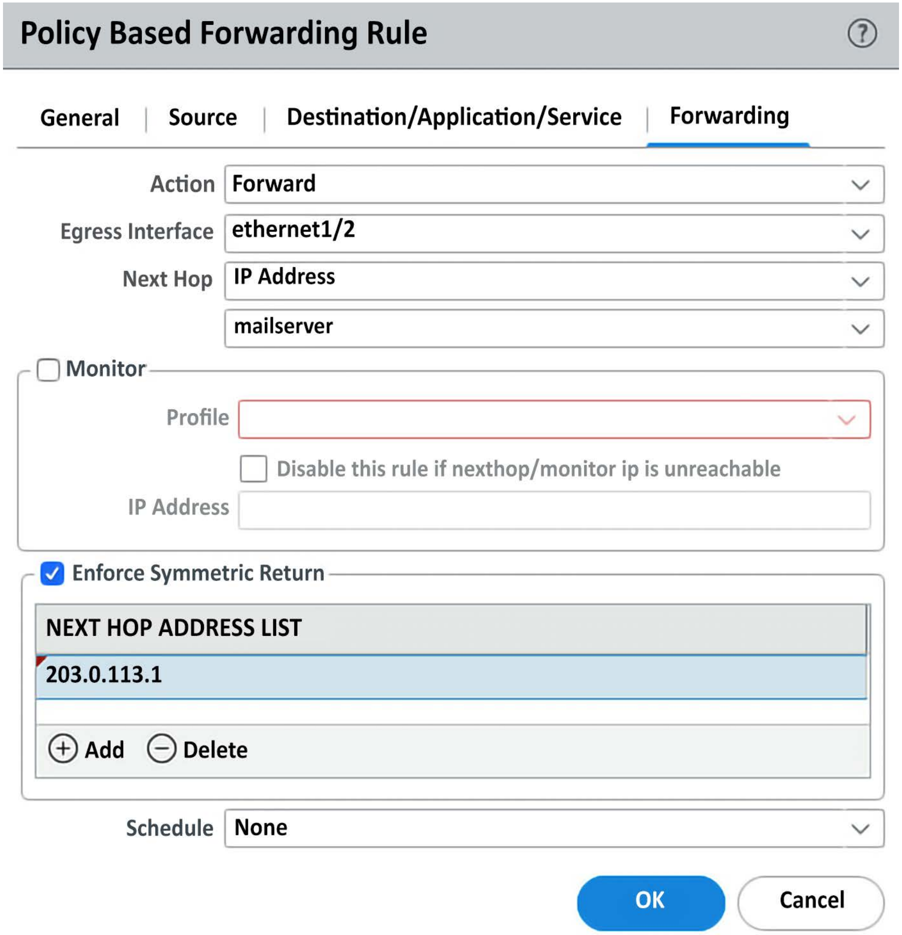


Figure 4.22: PBF rule set for Enforce Symmetric Return

Important note

Since the app's cache creates entries based on the destination IP address, destination port, and protocol ID, inbound PBF sessions to the same server are easily identified by their application in `app-cache`.

IPSec redundancy via virtual routers

Another common use case is to set up two virtual routers and connect a different ISP to each one. Then, configure a VPN tunnel on each virtual router so that there are two simultaneous uplinks to the remote site. PBF can then be used to route user sessions to the remote end over the primary link, and if this ISP were to fail, you can revert to the default route and use the backup link, as illustrated in the following diagram:

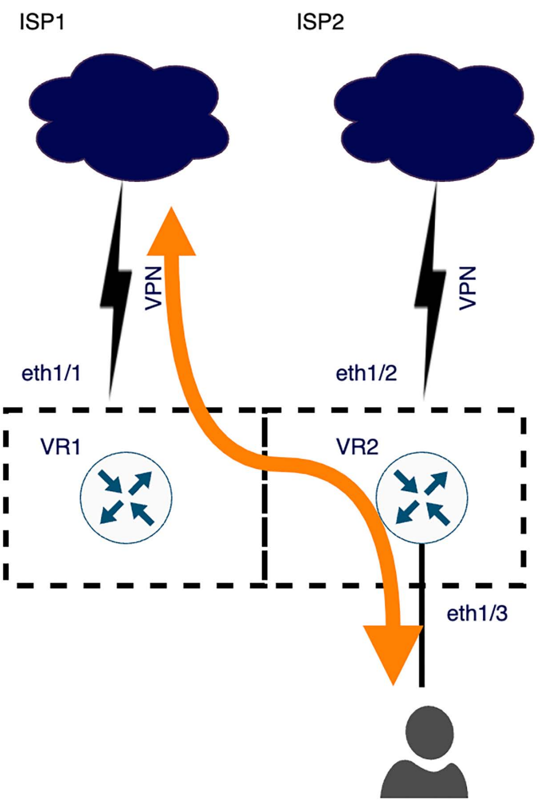


Figure 4.23: VPN redundancy using PBF

You can now receive inbound connections on an interface that does not have (the dominant) default route and ensure return packets flow back through the original interface. In some cases, the fancy way is not always the best way. We will take a look at simplified link balancing in the next section.

Equal cost multipath as an alternative

As an alternative solution to the previous use case, **Equal Cost Multi-Path** (**ECMP**) routing can be enabled on the virtual router where the ISPs are connected. ECMP enables link balancing over multiple paths so that you can combine several smaller-bandwidth ISP connections for increased performance. Whereas PBF requires rules to direct sessions over an alternative link with more control over which destinations or ports to send over each link, ECMP simply spreads sessions over multiple links, making it more of a true load balancing solution.

To set up paths (each "path" is an ISP uplink), first enable ECMP by going to **Network** | **Virtual Routers** | **VR**, which holds your ISP uplinks. ECMP supports up to four paths, and I recommend you consider these three in particular:

- Set **Symmetric return** if you want packets to go back out through the same interface that they came in through. This is useful if you host services on one or both ISP subnets.
- Enabling **Strict Source Path** ensures firewall-sourced traffic (IKE/IPSec) is not subject to ECMP and will be bound to the interface the IPSec tunnel is configured on and use regular routing to determine the route path. This setting should only be enabled if you require sticky VPN connections that only use a particular outbound interface and route.
- **Max Path** tells ECMP how many interfaces can participate. This number should correspond to the number of uplinks you intend to balance over.

As you can see in the following screenshot, there are several load balancing methods that you can choose from. Pick a method that best suits your needs as each will result in a unique behavior:

- **IP Modulo** uses a hash of the source and destination IP to determine which ECMP route to take.
- **IP Hash** uses the source IP or source IP-destination port to calculate which ECMP route to take.
- **Weighted Round Robin** lets you decide which interface gets more or fewer sessions assigned to it based on weight; a higher weight assigns a higher preference, as shown.
- **Balanced Round Robin** balances ECMP sessions equally across all uplinks.

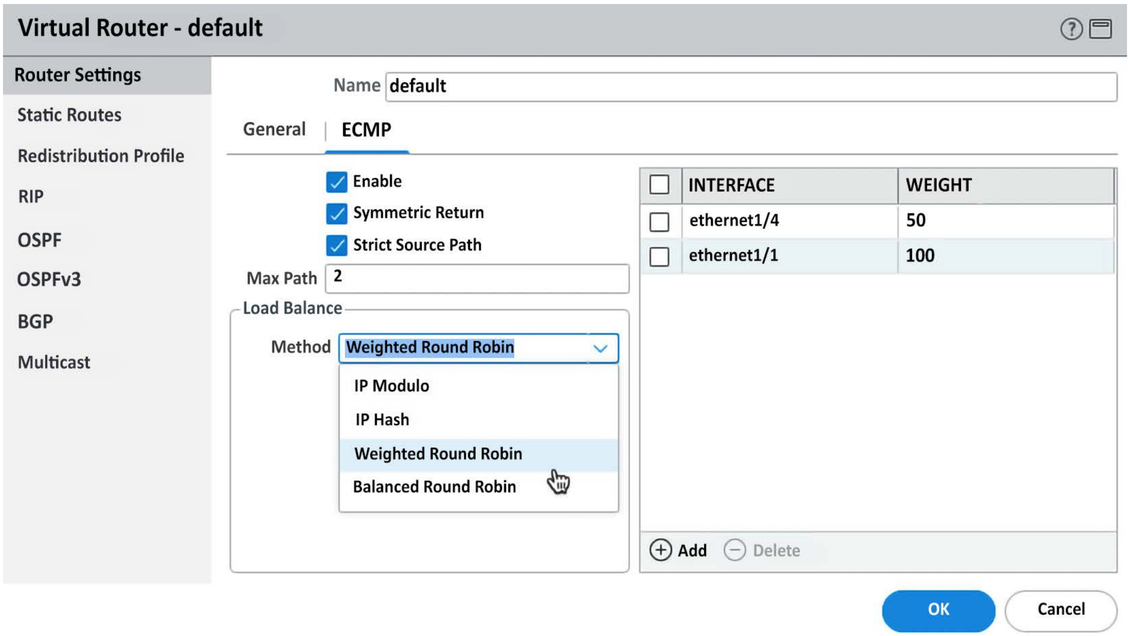


Figure 4.24: ECMP routing

In this section, you learned how to use PBF and symmetric return to manipulate how sessions are egressed out of the firewall, as well as how ECMP can help bundle ISP uplinks.

Summary

In this chapter, you learned how to shape sessions to prevent your internet uplink from getting flooded while guaranteeing business-critical applications always have bandwidth available. You can now implement decryption so that TLS sessions can be inspected for App-ID and threats, and you can leverage PBF and ECMP to control how sessions flow, regardless of routing. You are able to implement QoS rules and profiles to efficiently limit bandwidth for chatty applications and ensure your important applications have a guaranteed bandwidth so that even at the busiest times, they will never encounter any bandwidth issues.

If you're studying for the PCNSE, take note that QoS is achieved by setting rules that assign a class to sessions that match the rule and that profiles are added to interfaces to define which guarantees and maximum throughput are assigned per class; Class 4 is the default class. Remember that SSL decryption works best if a decryption profile is assigned, even for no-decrypt rules. Take note that there are two types of interface that will forward decrypted traffic off-device.

In the next chapter, we will enable services on the firewall that are traditionally hosted on servers in the network and we will learn about setting the firewall in high-availability mode and adding virtual systems.

5

Services and Operational Modes

Most networks have some supporting services to ensure users don't need to configure their laptop, mobile, or workstation to get access to corporate resources or the internet. **Dynamic Host Configuration Protocol** (**DHCP**) helps users connect to a network by assigning them an IP address and several other settings. The **Domain Name System** (**DNS**) allows them to visit websites with a friendly name. Rather than needing to stand up a server at each location, we will be configuring the firewall to provide these services.

High-availability clustering and virtualization make deployments more resilient to failure and ensure that businesses can go on, even if something breaks. We will be setting up **High Availability** (**HA**) and reviewing implications for using both Active/Passive mode and Active/Active mode. We will also take a deeper look at using virtual systems on a chassis to logically split up network segments.

In this chapter, we're going to cover the following main topics:

- Applying a DHCP client and DHCP server
- Configuring a DNS proxy
- Setting up high availability
- Enabling virtual systems
- Managing certificates

These will enable you to deploy a firewall in a location where there are no other services available and take on some of the responsibilities that would otherwise have been addressed by the services. HA will help you create a robust deployment that will survive a failure and virtual systems will help you segregate multiple environments without needing to acquire more hardware.

Technical requirements

This chapter covers basic networking protocols like DHCP and DNS, and you should be comfortable configuring these in an enterprise environment. Prior experience with clustering and multi-tenant systems is recommended.

Applying a DHCP client and DHCP server

In most offices, DHCP is the norm when it comes to setting clients up on the network, but for smaller offices, it can be difficult or expensive to set up a dedicated server to provide IP addresses, or your local ISP may require you to connect a DHCP client to their network before they're able to assign you an IP address and let you on the internet. Luckily, a firewall can also perform these duties. We will start by setting up a firewall as a DHCP client to a dynamic ISP.

DHCP client

To set a data plane interface up as a DHCP client, follow the same steps as you would to configure a regular Layer 3 interface (or configure a Layer 3 VLAN interface), but set **IPv4** to **DHCP Client:**

1. Edit the interface.
2. Set the mode to **Layer3**.
3. Select an appropriate zone and virtual router.
4. Set **IPv4** to **DHCP Client**.

You can choose to accept the default route from the ISP or set your own in the virtual router, and if you want, send a hostname upstream (some ISPs may require you to set a specific hostname, which you can set here without changing the actual system hostname):

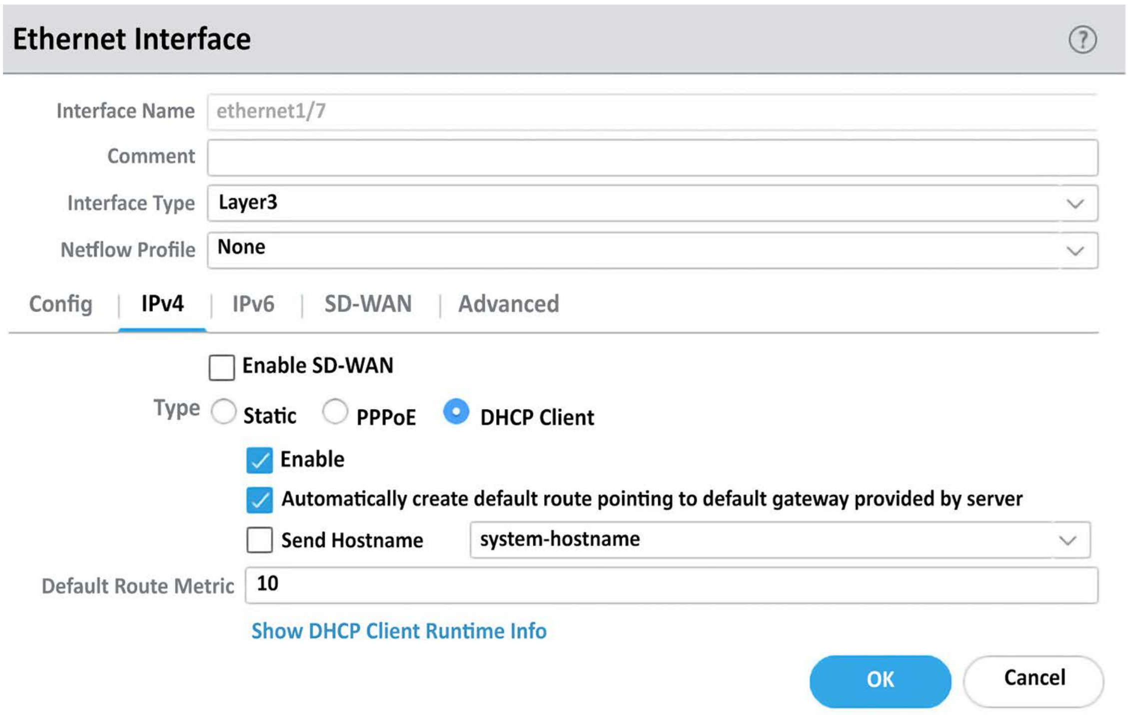


Figure 5.1: Interface in DHCP Client mode

Once the change has been committed, you can view the runtime info and use the **renew** or **release** options in the interface configuration or the interface overview at **Network** | **Interfaces** | **Ethernet**. The **renew** option will request a refresh on the interface IP information from the DHCP server and **release** will clear the interface of its IP address, which may be handy when there's an IP conflict.

Some useful CLI commands include the following:

```
> show dhcp client state all
> request dhcp client renew all
> request dhcp client release all
```

You are now able to configure the firewall as a DHCP client and receive an IP address on an interface.

Next, we can extend this service into the local zones, providing IP addresses to internal clients.

DHCP server and relay

On the inside of your network, the firewall can function as a DHCP server and hand out IP addresses, DNS and **Network Time Protocol** (**NTP**) settings, and many other options. The DHCP server component needs to be attached to the interface that is in the same broadcast domain as the IP subnet or range it will be handing out. Do the following in **Network** | **DHCP** | **DHCP Server**:

1. Create a new DHCP server profile.
2. Select the interface your clients are connected to.
3. Select the appropriate mode:
    - **Auto** polls the network for another DHCP server and deactivates itself if one is found.
    - **Enabled** sets the DHCP server to always on; this could conflict with an existing DHCP server on the network.
    - **Disabled** sets the DHCP server as inactive.
4. Enabling **Ping IP when allocating new IP** makes the firewall ping an IP before assigning it to a new host. This prevents IP conflicts.
5. Choose **Unlimited** or **Timeout** lease time:
    - **Timeout** lease time will time out and remove a lease after the set amount of time, forcing the client to renew its lease or lose it if the client is no longer online at the time the lease expires.
    - **Unlimited** lease time will keep the lease permanently. If the IP pool is depleted, the next new client will not be able to receive an IP.
6. Add the IP pool subnet or range, and add reservations as needed:
    - A reservation without a MAC entry will simply withhold the IP address from being assigned. The host using this IP needs to be configured manually.
    - A reservation with a MAC address will only assign the IP to the host with the matching MAC address on its interface.
7. In the **Options** tab, you can inherit DHCP options from an upstream (ISP) DHCP server if you like. This could be useful for sharing the ISP DNS with downstream clients.

8. The gateway and subnet mask need to be set to the firewall interface IP and subnet mask.
9. DNS, NTP, and other options can be manually configured or set to the inheritance of the upstream DHCP server.
10. Custom DHCP options can be added in the range of *1-254*.
11. Don't forget to add an intrazone security rule that allows the application `dhcp` if a general drop rule has been configured to supersede the default intrazone allow rule.

The following screenshot illustrates what the DHCP server configuration may look like:

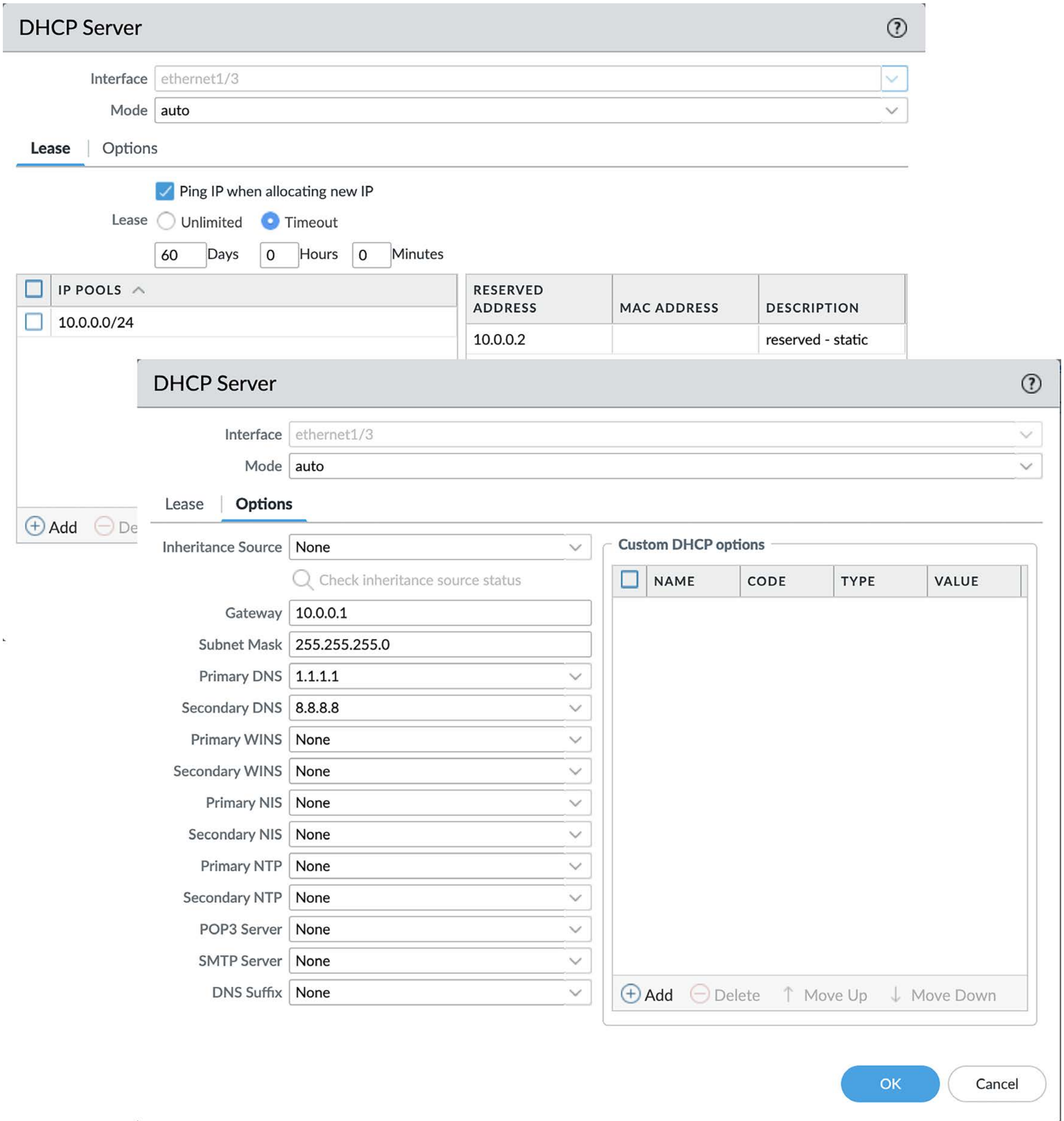


Figure 5.2: DHCP Server configuration

As shown in the following screenshot, the DHCP relay only needs to be assigned to the interface the clients will be active in, and the IP where the DHCP requests need to be forwarded to:

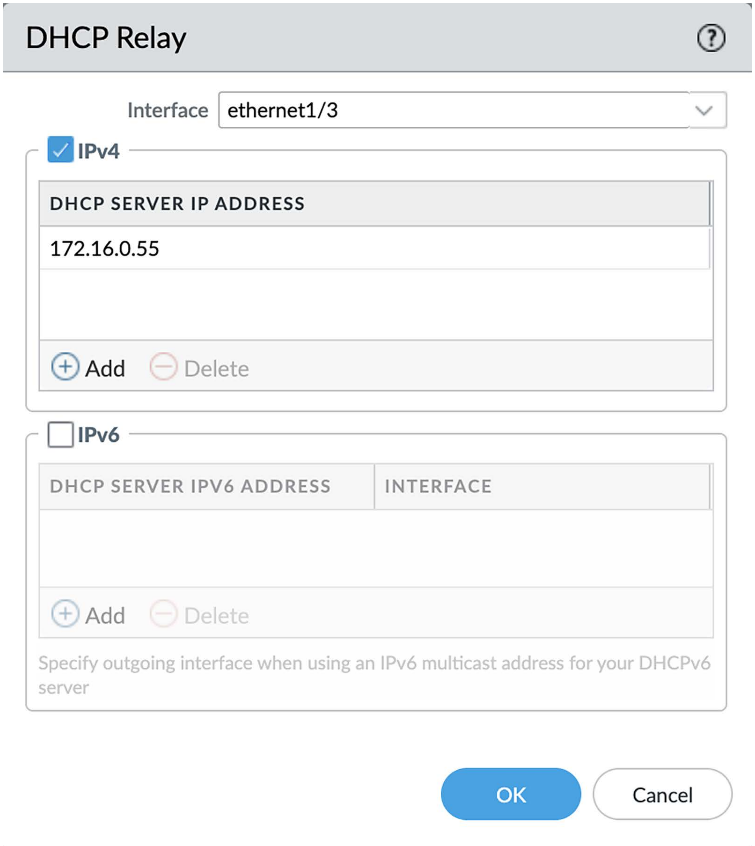


Figure 5.3: DHCP Relay configuration

The firewall will listen for DHCP requests on the interface and forward all DHCP packets to the DHCP server that is located in a network that is connected to a different interface.

Important note

DHCP relay requires that the application `dhcp` is allowed from the client to the firewall interface, and also from the firewall interface to the DHCP server.

Some areas may not have an ISP available that can provide a static IP address, which makes having the ability to set the firewall as a DHCP client a very nifty tool in your arsenal.

Providing DHCP leases, or relaying DHCP requests for internal clients, also takes away the need to have local infrastructure. This knowledge can help you quickly deploy a small office, but you may also need DNS services, which we'll cover next.

Configuring a DNS proxy

A DNS proxy helps control how internal clients connect to DNS servers and where they get domain information from, or which information they receive.

> **Important note**
>
> Clients must be configured with the firewall's interface IP set as the DNS server. This can be forced via the DNS attribute in the DHCP server or may need to be set manually.
>
> The firewall may need a security rule that allows DNS connections to the firewall interface from the clients, and a second one that allows DNS from the firewall interface out to the internet.

Configure the DNS proxy by following these steps:

1. Create a new DNS proxy object in **Network** | **DNS Proxy**.
2. Add a name and, if you want to inherit DNS configuration from an upstream DHCP server (ISP), set **Inheritance Source** to the DHCP interface.
3. Set the primary and secondary DNS server for outgoing DNS requests to servers of your choice, or select **Inherit** if you want to use your ISP's DNS servers for generic lookups (and if you selected an **Inheritance Source** in step 2).
4. Add the interfaces that the firewall will be accepting DNS queries on.
5. In the **DNS Proxy Rules** tab, add redirect rules. Requests for these **fully qualified domain names** (**FQDNs**) are redirected to different DNS servers, which can be internal DNS servers, serving up internal records with a private IP. This could be useful for internal clients to receive the private IP of internally hosted servers.
6. In the static entries, add the FQDNs that the firewall will reply to with the IPs you configure here. These queries will not be forwarded to any DNS server.
7. In PAN-OS 11.2 and above, there is also the **Encrypted DNS** tab. Encryption can be enabled for clients connecting to the firewall. For **Server Settings**, you can select which **Connection Type** the firewall uses to communicate with the upstream DNS server:
    - **DoH** (DNS over HTTPS) uses port 443 to make DNS queries so no other services using port 443 (like a management profile) can be set on the interface that acts as a DNS proxy.
    - **DoT** (DNS over TLS) uses port 853 to perform queries.
    - **Origin** will send DNS queries to the primary DNS server using the same type as it originally received from the client. If the primary DNS server times out, the secondary will be queried.
    - **Cleartext** will use regular DNS lookups on the upstream DNS server.
    - For **Client Settings**, you can select **DoT**, **DoH**, and **Cleartext** as allowed DNS connection types the firewall will accept from the client.
    - Next, set a **SSL/TLS Service Profile.** This is the certificate the DNS proxy will present to clients making TLS-enabled DNS requests, so it needs to be signed by a trusted root certificate, like the organization's PKI or a distributed self-signed CA, and must match the interface IP in the SAN field.

8. In the **Advanced** tab, you can configure the following:
   - The maximum concurrent pending TCP DNS requests (between `64` and `256`).
   - The interval and maximum attempts for unanswered UDP queries.
   - The **Time To Live** (**TTL**) can be enabled to set the maximum time (between `60` and `86400` seconds) a record can be cached before the firewall is forced to refresh the entry. By default, a record is not deleted until the firewall runs out of cache memory, or the record's own TTL expires.

   An extension mechanism for DNS can be cached if the option for EDNS is checked. This enables the caching of partial DNS responses that are greater than `512` bytes.

The following screenshot shows a fully configured DNS proxy object:

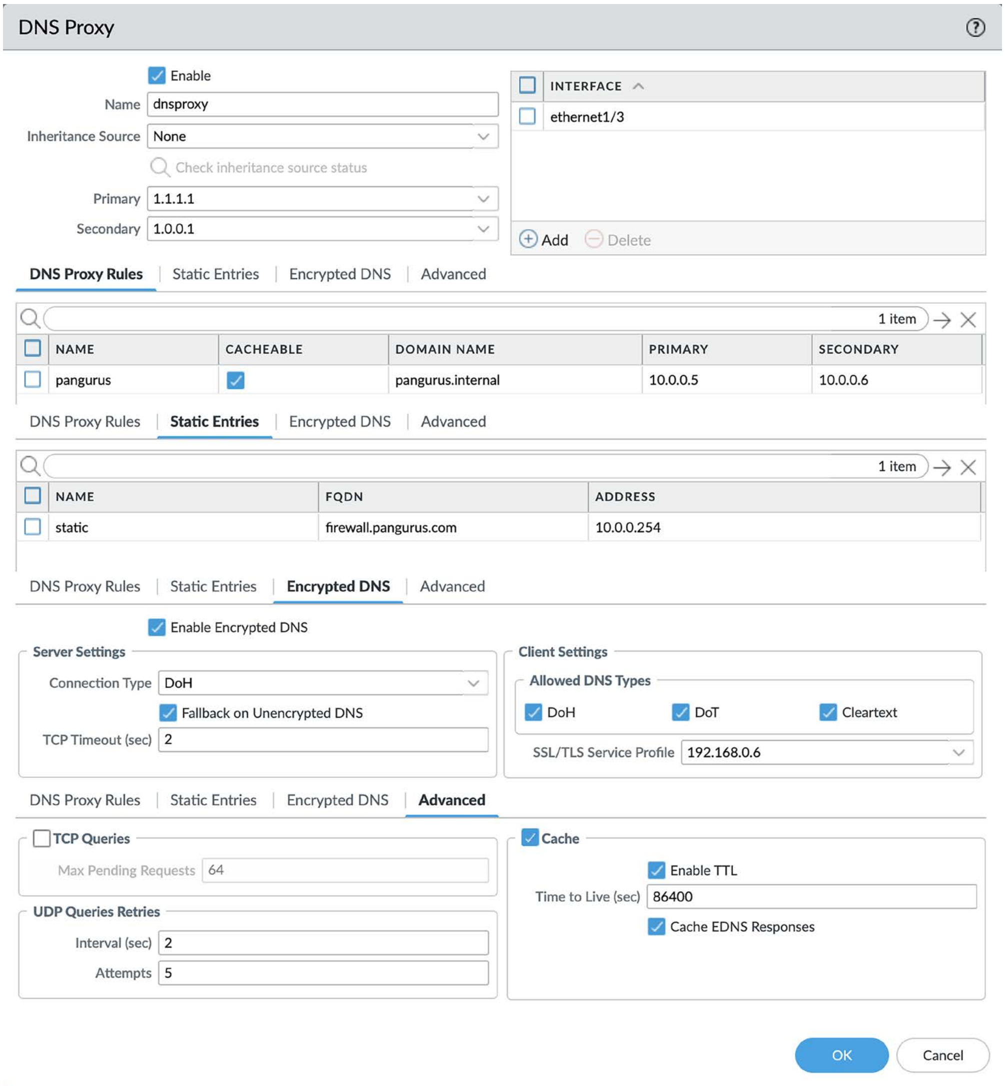


Figure 5.4: DNS proxy object

You are now able to configure a DNS proxy object that can control which servers your clients are able to connect to and perform some rewriting where needed. In the next section, we'll learn how to set up HA.

Setting up high availability

HA is a configuration where two identical (the same chassis or VM version) firewalls are connected to form a cluster. When clustering is enabled, both systems will form a single entity to the outside and will handle failover for certain problems so the service remains available to users. The following types of monitoring are, or can be, performed in a cluster member to ensure its own and its peers' health:

- **Link monitoring**: If an interface goes down, the member fails
- **Path monitoring**: If an IP becomes unavailable, the member fails
- **Heartbeat monitoring:** The peers periodically send heartbeat packages and hello messages to verify they are up and running
- **Hardware monitoring:** The member continually performs packet path health monitoring on its own hardware and fails if a malfunction is detected

When you enable HA, you need to select a **Group ID**, as shown in *Figure 5.5*. This ID needs to be identical for both members. The **Group ID** will also have an impact on the **Media Access Control** (**MAC**) addresses associated with each interface as they switch to a virtual MAC that both firewalls will be able to claim via gratuitous **Address Resolution Protocol** (**ARP**) in case one member fails.

Important note

Any Layer 3 interface that is already active in the network will receive a new MAC address once HA is enabled (and committed), which could cause connectivity issues while switches and clients learn the new MAC associated with the firewall IPs. Some ARP tables may need to be cleared and static entries updated.

As seen in the following screenshot, there is a checkbox that allows you to disable **Enable Config Sync** between members. Use extreme caution if you disable this option as it will have far-reaching consequences (for one, each interface, zone, and object has a unique identifier that is normally synced between peers for session consistency; disabling this could prevent sessions from failing over).

It should only be used on rare occasions where the configuration must be different and as a rule of thumb should always be enabled unless there is a use case to disable it:

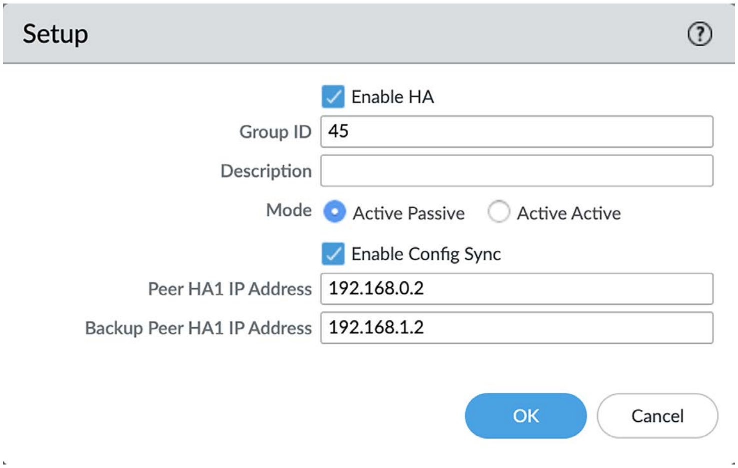


Figure 5.5: Enabling HA

There are several modes in which the cluster can be configured, which will be covered in the following sections. We'll first cover the modes and then go deeper into the two main modes of **Active/Passive** and **Active/Active** after we've laid out all the concepts that make up HA.

Active/Passive mode

In Active/Passive mode, one member (the primary member) processes all traffic while the secondary peer does not participate.

By default, the passive device will have its interfaces in a **shutdown** state, meaning any connected devices will also see the link as being down. Depending on your environment, this could prevent other clusters from functioning properly, in which case you will need to set these to **Auto** (up but not accepting packets).

Monitor Fail Hold Down Time keeps the firewall in a failed state (non-functional: see the *Firewall states* section) for the specified amount of time after an error was detected before setting the member to the passive state:

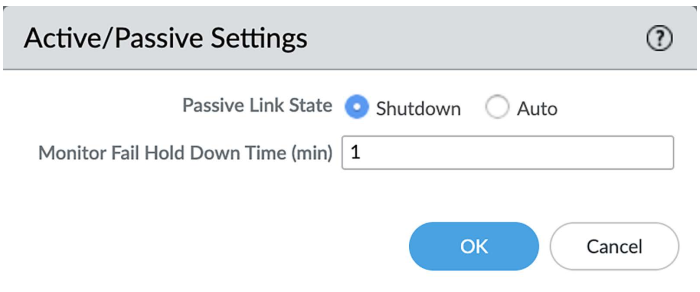


Figure 5.6: Passive Link State

If you set **Passive Link State** to **Auto** and you want even faster link negotiation, you can enable **Link Layer Discovery Protocol** (**LLDP**) and **Link Aggregation Control Protocol** (**LACP**) in passive mode by accessing the interface's **LACP** tab where these protocols have been enabled and checking **Enable in HA Passive State** as shown here:

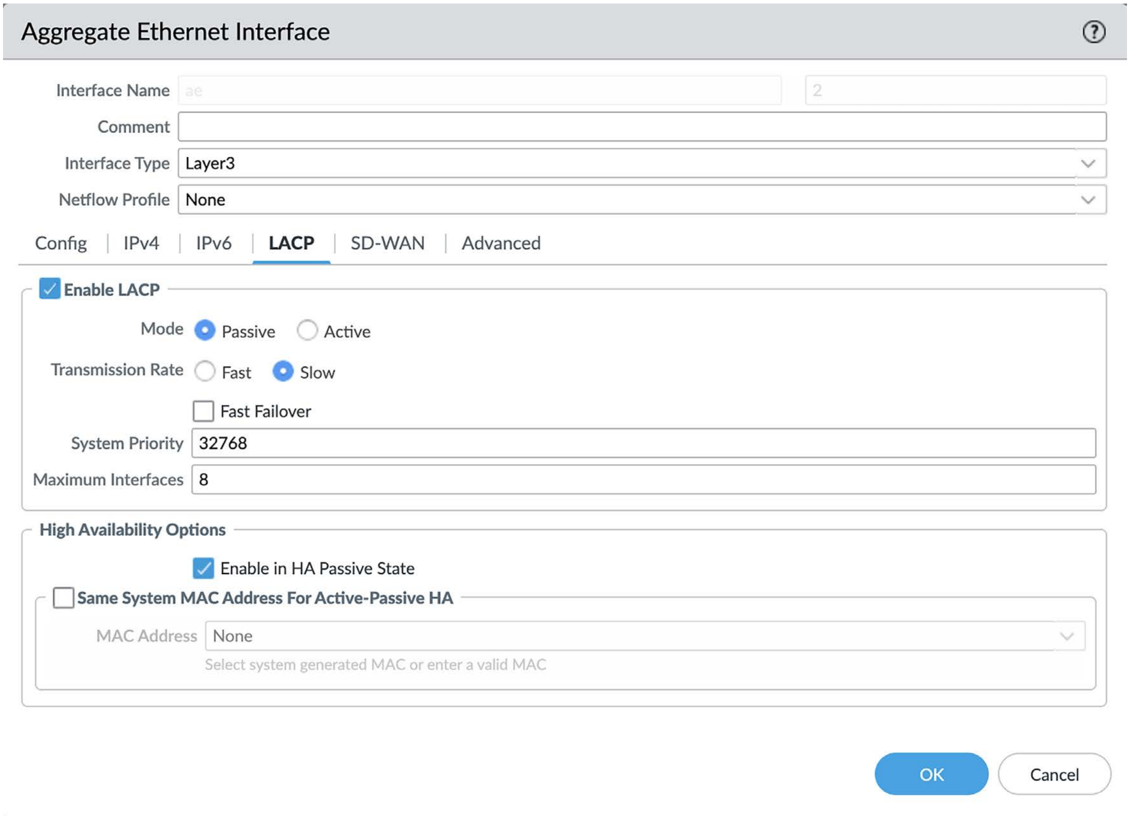


Figure 5.7: LACP HA passive state

The next clustering mode has both members participating in an active capacity.

Active/Active mode

In **Active/Active**, both firewalls actively take sessions and maintain their own session table. Session tables are synchronized with the peer. In Active/Active mode, both peers can individually process their own sessions, one peer can be assigned as master and process all sessions, or a load balancing/sharing mechanism (IP modulo or IP hash) can be used to distribute scanning among both peers.

This mode only supports Layer 3 and Virtual Wire interfaces and can't run as a DHCP client, and only the active-primary member can act as a DHCP relay.

> It is important to realize Active/Active is not a load balancing configuration. The main issue Active/Active is intended to tackle is asymmetric flows or a requirement for faster failover in a dynamic routing environment. An Active/Active cluster will be able to handle peak traffic bursts better than an Active/Passive cluster due to the availability of an additional active member, but the average load may be slightly higher for regular traffic as both peers will have more overhead synchronizing sessions.
>
> Active/Active introduces far more complexity than Active/Passive, so please consider the trade-off.

Clustering

A third type of HA is **clustering**. In this setup, multiple HA pairs and standalone devices can be combined into a geographical cluster. This can be a useful redundancy measure if, for example, there are multiple large datacenters each having its own HA pair. In case the entire site were to go down, another HA pair could resume the established sessions as each pair's state table is synchronized to all members of the cluster. Up to 16 devices can be part of a cluster. Not all members of the cluster need to be the same form factor (i.e., VM-300, PA-3200, and PA-5200 can all be part of the same cluster).

Each form factor supports the following number of clustering peers:

- PA-3200: 6
- PA-5200: 16
- PA-5450: 8
- PA-7050: 6
- PA-7080: 4
- VM-300, VM-500: 6
- VM-700: 16

Clustering is established by configuring **Enable Cluster Participation** in **Device** | **High Availability** | **General** | **Clustering Settings:**

1. Select the **Enable Cluster Participation** checkbox.
2. Set a **Cluster ID**. This ID needs to be identical among all members.
3. **Cluster Synchronization Timeout** (0–30 min) is the time a cluster member will wait before going into an active state if a cluster peer is preventing the cluster from fully syncing (e.g., when it is in an inactive or defective state).
4. **Monitor Fail Holddown Time** (1–60 min) is the amount of time the firewall waits before retesting a link that was previously down.
5. Next, the HA4 and HA4 backup links should be configured in **Device** | **High Availability** | **HA Communications** | **Clustering Links**. These are dedicated links to synchronize the state tables among all cluster members.

Important note

The cluster state table is not added to the local firewall's active state table and is stored and maintained separately until a cluster member is required to take over traffic from a downed peer.

6. Once the HA4 links have been configured, the cluster members need to be added in **Device** | **High Availability** | **Cluster Config**.
7. Each member is added individually by adding its **Device Serial Number**, **HA4 IP Address**, and **HA4 Backup IP Address**. **Session Synchronization** should be enabled to synchronize the local session table to the peer.

A typical active/passive configuration will look similar to the following screenshot:

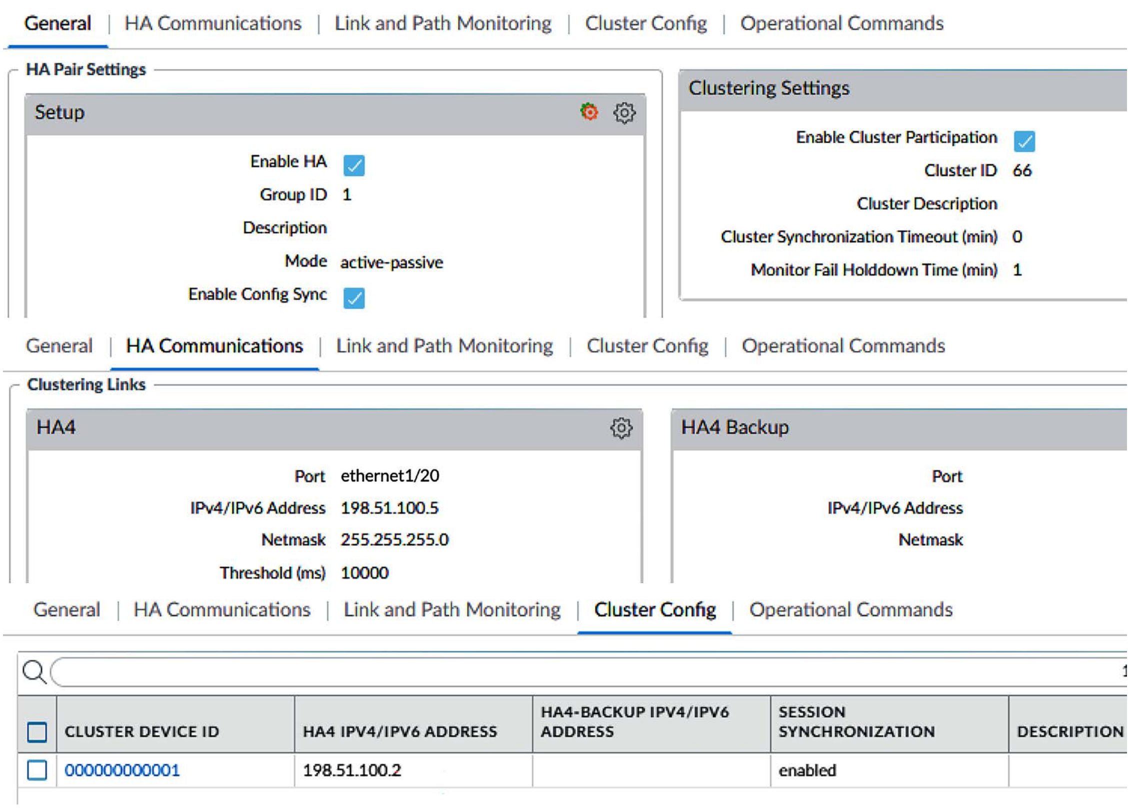


Figure 5.8: Cluster Config

The cluster status can be checked from the **Dashboard** HA widget.

Firewall states

The firewall can be in one of eight states while it is a cluster member:

- **Initial**: The firewall assumes this state after it boots up, at which time it will start looking for a peer. If none is found after the timeout expires, the firewall becomes active.
- **Active**: The firewall is accepting and processing packets.
- **Passive**: The firewall is in a standby state; it receives state table and runtime object updates from the primary member while it monitors the active member with hello and heartbeat messages to ensure it does not need to take over.
- **Non-functional**: The firewall has encountered a failure condition, which could be due to an interface being down or data plane error but could also be caused by a configuration mismatch or PAN-OS mismatch (the member with the highest version of PAN-OS will go into a non-functional state).
- **Suspended**: The firewall still receives update information from the active member, but an administrator has temporarily made this device incapable of taking an active role. This could be useful for troubleshooting or during an upgrade.
- **Active-primary**: In Active/Active mode, DHCP servers, User-ID agents, NAT, and PBF rules can be assigned to one or both members.
- **Active-secondary**: All of the above, except that the active-secondary can't be a DHCP relay.
- **Tentative**: In Active/Active, if the firewall leaves the suspended or non-functional state, it will first become tentative while it synchronizes sessions. It will forward all received packets to its peer over the HA3 link for processing and then send them out over its egress interface until it leaves this state and starts processing packets itself.

To ensure both cluster members are able to synchronize configuration and share session tables, special interfaces are needed.

HA interfaces

HA requires several interfaces to perform certain tasks: **HA1**, **HA1 backup**, **HA2**, **HA3**, and **HA4**.

HA1 is the primary management link that is used to synchronize configuration and perform monitoring (hello messages) of the remote peer. **HA1** can be enabled on the management interface, a dedicated interface (visibly marked **HA1** interface on the chassis), or a data plane interface set to interface type **HA**. If the **HA1** link goes down, the passive member will assume the primary member is down and assume the **Active** state.

HA1 is a Layer 3 interface, so an IP address needs to be set for the local and remote **HA1** interface (see *Figure 5.10*) and uses ports `28260` and `28769` for cleartext or `28` for encrypted communication.

Due to the sensitivity of the information traversing **HA1**, the sessions can be encrypted: **HA1** syncs all configuration except the management parameters (basically, everything under the **Device** tab is considered local). To allow encryption, both peers' HA keys need to be exported and imported on the other peer.

You can find the export/import option in **Device** | **Certificate Management** | **Certificates**:

Figure 5.9: Import and export of the HA key

HA1 synchronizes this runtime information:

- User to IP/group mapping
- DHCP lease
- IKE keys (Phase 2)
- **Forwarding information base** (**FIB**)
- URL cache
- PPPoE
- SSL VPN logged-in users

Because the **HA1** link is so crucial, it is best practice to have an **HA1 backup** interface configured to prevent a *split-brain* if **HA1** ever were to get disconnected. A split-brain is when both HA members think the other peer is down, and both take ownership of the floating IP addresses at the same time, which will cause all kinds of havoc and mayhem on the network.

If **HA1** is set on a dedicated interface, an **HA1** backup can be enabled on the management interface, a dedicated **HA1** backup interface, or a data plane interface set to interface type **HA**. An **HA1** backup uses ports `28260` and `28770`. The peer IP configuration is set in the **General** tab, while the local IP addresses are set in the **HA Communications** tab:

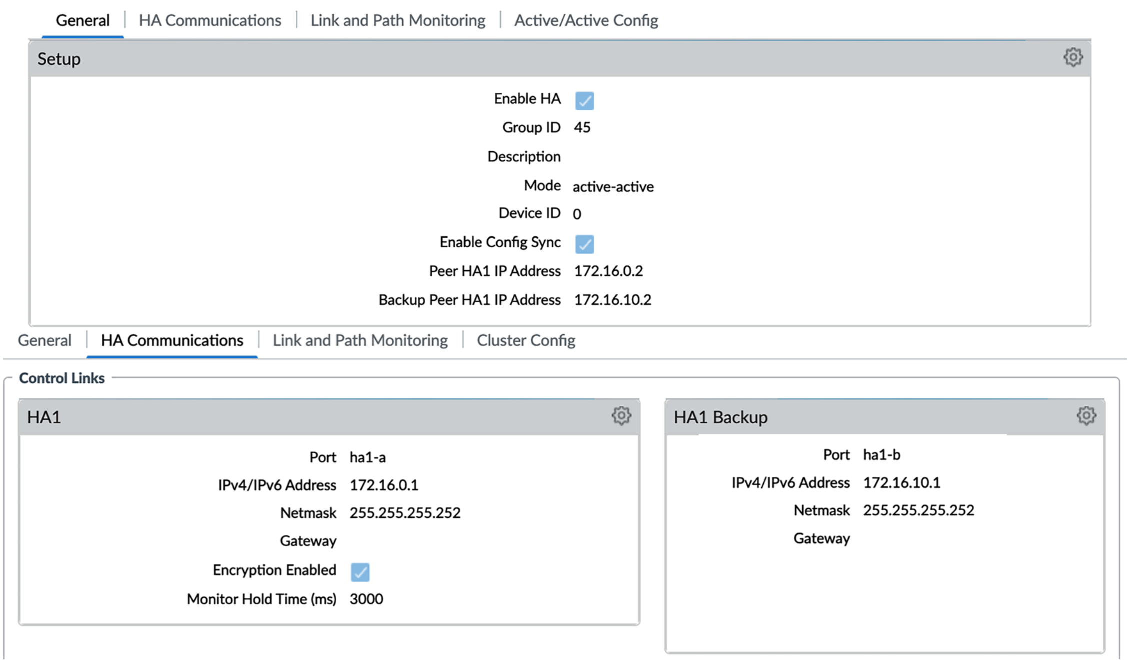


Figure 5.10: HA1 configuration

HA2 takes care of the session table being synced over to the peer. By default, the transport mode for **HA2** is **Ethernet** (**Ethertype 0x7261**), which means it has a very low overhead as it doesn't use IP headers, which is ideal if both devices are directly connected. If some sort of transport is required, you can use the following:

- IP (IP protocol 99) mode, which uses very basic IP headers.
- UDP (UDP port 29281) mode, which uses UDP to transport the session state information over a routed network.
- **HA2 Keep-alive** can be configured to monitor and maintain the **HA2** connection. A log will be written in the event of a failure, or in Active/Active mode, the action can be set to **Split Datapath** to instruct both peers to keep processing traffic while only maintaining a local state table until **HA2** returns:

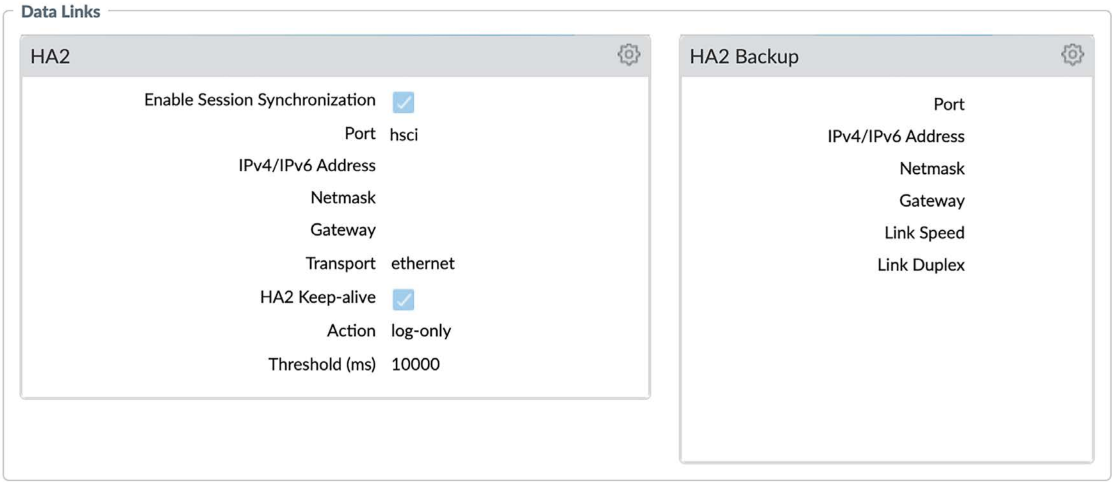


Figure 5.11: HA2 configuration

HA2 synchronizes this runtime information:

- Session table
- ARP table
- **Neighbor Discovery** (**ND**) table
- MAC table
- IPSec sequence number
- Virtual MAC
- **Stream Control Transmission Protocol** (**SCTP**) associations

An **HA2** backup can be configured on a dedicated interface like the **HSCI** (**High Speed Chassis Interconnect**), a Layer 1 SFP+ direct connection interface), or a data plane interface set to interface type **HA** to serve as a backup in case HA2 fails.

HA3 is used exclusively in Active/Active deployments and is used to forward (whole) packets to the peer for packet inspection. It uses MAC-in-MAC encapsulation to transmit packets between peers, with the entire packet as the payload. The HA3 link therefore needs to support jumbo frames as frames will be larger than the data packets. This may be needed when the primary device is set as the session owner, when the session setup is IP modulo or IP hash and the remote peer is selected for session setup, or when asymmetric packets are received on the member that does not own the session. The packets are sent over for session completeness on the Session Owner device, and then returned to the recipient so it can egress the packet out of its data plane interface (asymmetry is maintained but the session is fully scanned by one session owner).

HA4 is used exclusively in a cluster configuration where multiple standalone or HA pairs share their state table for geo-redundancy.

Now that you understand which HA modes are available, we can go ahead and set them up.

Setting up Active/Passive mode

Follow these steps to configure Active/Passive mode, starting with the primary member:

1. In **Device** | **High Availability** | **Setup**, enable **High Availability**.
2. Pick a **Group ID**. Go with `50` if you don't have a clear preference.
3. Leave the mode as **active-passive**.
4. Make sure **Enable Config Sync is enabled.**
5. Peer `HA1` IP: Use a private IP (in a `/30` subnet) that does not overlap with your existing internal subnets (for example, `172.16.0.2`). If you have a smaller device without dedicated HA interfaces and need to use the management interface as the `HA1` interface, set the peer's management IP.
6. Backup peer `HA1`: If you are able to sacrifice a data plane interface as a backup `HA1` interface, add another non-overlapping private IP (for example, `172.16.1.2`), or the peer's management IP if you intend to use the management interface as a backup `HA1` link.
7. Click **OK**.

If you need to change the passive link state interface's behavior, open **Active/Passive Settings** and change **Passive Link State** to **Auto** (this will bring the interface up when the device is in a passive state). **Monitor Fail Hold Down Time** is used to leave the device in a non-functional state for the specified amount of time after a monitor failure before it is allowed to transition to the passive state.

Next, open **Election Settings**:

1. Set **Device Priority** to `50`.
2. Enable **Heartbeat Backup** if you do not have a HA1 backup link configured. This will use the management interface to send a simple heartbeat to the remote peer.
3. **Preemptive** will ensure the device with the lowest priority is always the active member and will, after the primary member has failed, preemptively fail back to the primary member (see HA timers below) after a set amount of time. The drawback is that, if the failure condition is still present, the primary member will then need to fail over again. This process can repeat until the failure condition is fixed or until the primary member reaches its maximum number of flaps.
4. HA timers are set to **Recommended per platform**, but you can choose **Aggressive** for faster failover (but at a cost of overhead), or choose **Advanced** to manually change timers and counters. These are a few interesting counters:

    - **Promotion Hold Time** is the amount of time the secondary member will wait before becoming active after the connection with the primary one has been lost.
    - **Hello Interval** is the number of milliseconds between hello messages.
    - **Heartbeat Interval** is the amount of time between ICMP heartbeat packets.
    - **Flap max & Preemption Hold Time:** If you enable **Preemptive**, the firewall will blindly *flap* back to the active state after the preemption hold timer expires. If the original error that caused it to fail still exists, it will fail again. The **flap max** counter will prevent the firewall from repeating this scenario more than the specified number of times, at which time the firewall will go into a *permanently* failed state that can only be recovered via manual intervention.
    - **Monitor Fail Hold Up Time** is the amount of time the firewall will wait to fail over once a monitor (path, interface, and so on) has been detected, in the case of an extremely short interruption.
    - **Additional Master Hold Time** is used to add even more hold time to **Monitor Fail Hold Up Time**.

5. Click **OK**.

Your configuration should look similar to the following screenshot:

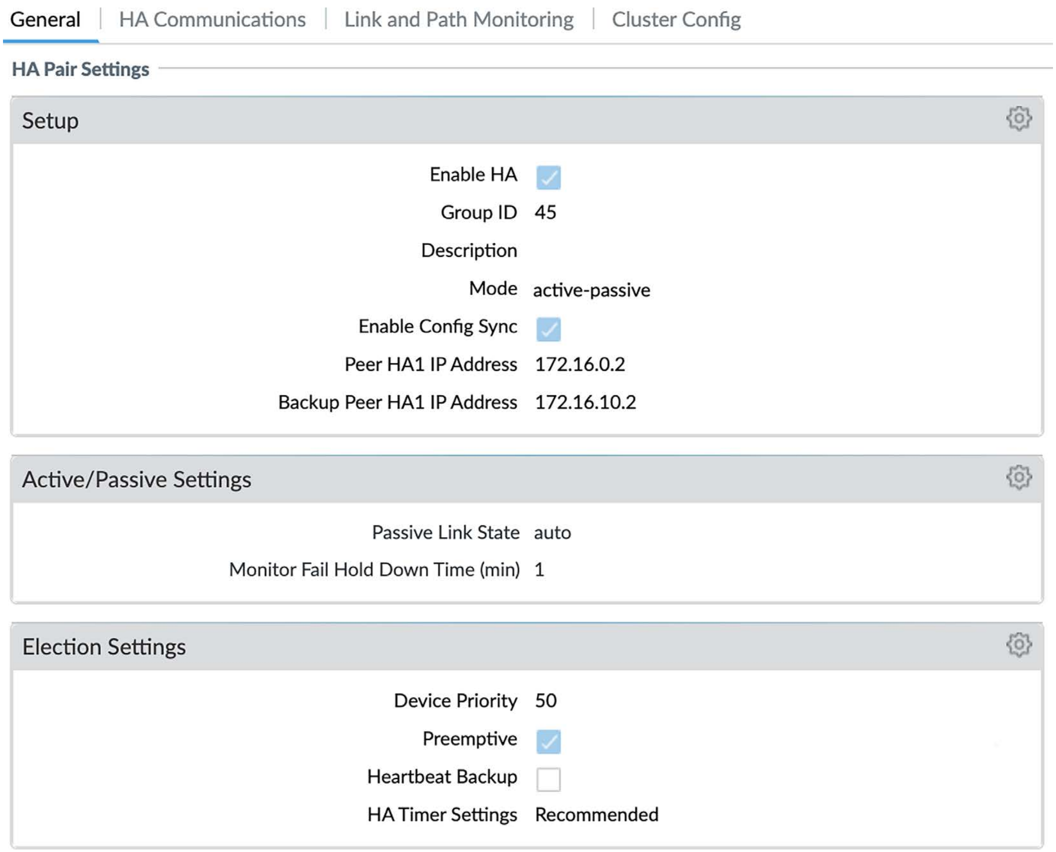


Figure 5.12: Active-Passive configuration

Next, we need to configure the HA links that enable both peers to communicate. In **the HA Communications** tab, open **Control Link**:

1. Set the interface to the dedicated `ha1-a` link if possible, or select a data plane interface (if you have not prepared a data plane interface yet, go to **Network** | **Interfaces** and select an interface. Set it to type **HA** so it can be used for high-availability communication). Fill in the IP address `172.16.0.1` and subnet mask `255.255.255.252`. Add a gateway if needed and enable encryption (make sure you exported/imported the HA keys on both peers). Alternatively, you can set the management interface instead of a dedicated or data plane interface.
2. **Monitor hold time** is the amount of time to wait before declaring a failure of the peer when **HA1** connectivity is lost. With heartbeat backup and **HA1** backup in place, this number can be lowered significantly. If neither backup option is available to you, do not lower this number as a short interruption could lead to a *split-brain*, where both peers become active, which is not fun for anyone.
3. Repeat *step 1* for **HA1** backup, using the second dedicated interface, `ha1-b`, a second data plane interface, using the second IP range from *step 6: Backup peer HA1...*(`172.16.1.1` to `255.255.255.252`), or the management interface.

4. Click **OK**.

Next up is the data link that will be used to synchronize the session state table:

1. Open the **HA2** settings and enable session synchronization.
2. If available, use the HSCI interface; otherwise, set a data plane interface (you can create an aggregate interface and set it to type **HA**, and use the aggregate here as well).
3. If you are able to use the **ethernet** transport mode, there's no need for IP addresses. If you need to use the IP or UDP transport mode, use a third non-overlapping subnet (for example, `172.16.3.1` and subnet mask `255.255.255.252`).
4. Enable **HA2 Keep-alive** and leave it as **Log Only** (**Split Datapath** is an Active/Active feature). It is best practice to have HA2 Keep-alive enabled as it will generate logs whenever HA2 connectivity is lost or restored.
5. If you are able to sacrifice another data plane interface, you could add it as the **HA2** backup interface. The **HA2** backup link is only used if the main **HA2** link goes down or if the keep-alive messages exceed the threshold.

The **HA Communications** tab should now look similar to the following screenshot:

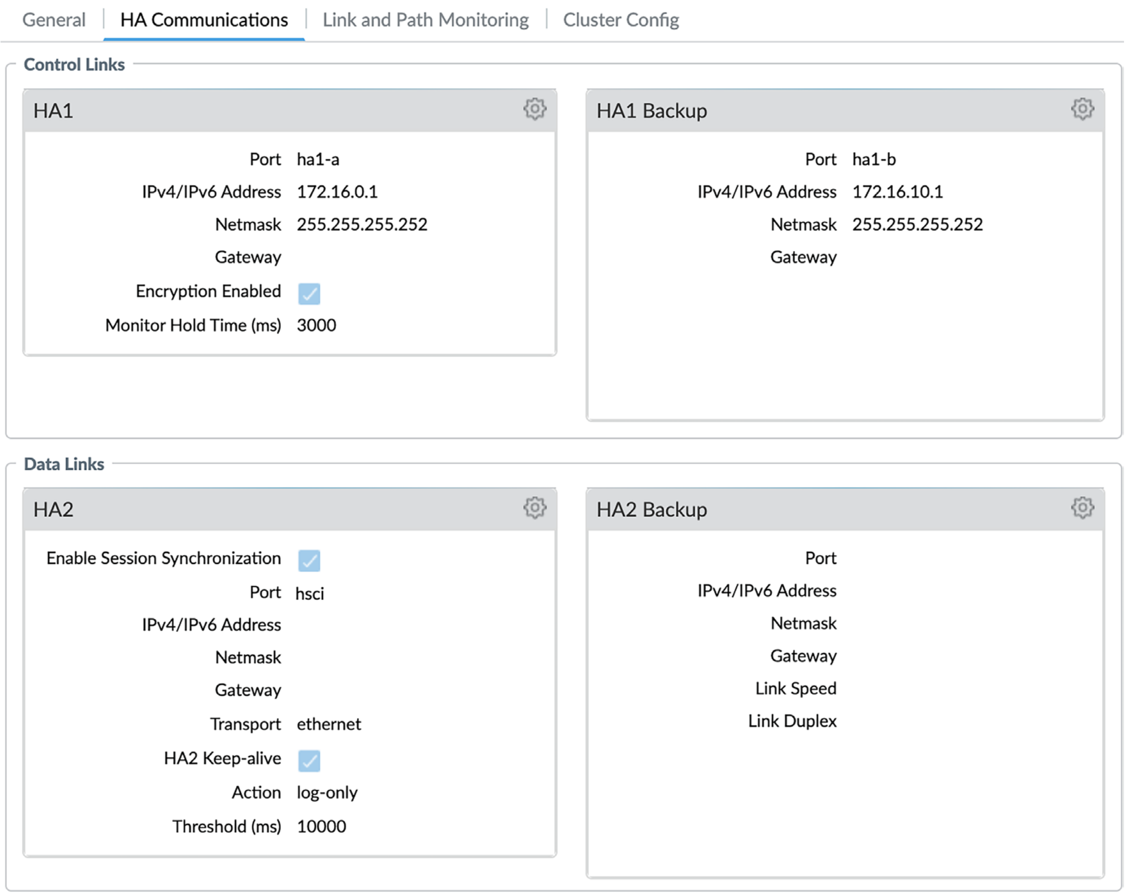


Figure 5.13: HA Communications configuration

Link state should be monitored to ensure the member fails over when an interface goes down. Path monitoring can be added in addition to ensure a remote router is available to pass traffic. Access the **Link and Path Monitoring** tab:

1. Enable link monitoring and create a **Link** group.
2. In the **Link** group, add all the interfaces that need to be monitored and set the fail condition to `any`. A group could be created where `all` interfaces need to be down for the chassis to fail, which could be helpful if you have redundant links and don't need an HA failover if just one or part of the links is down.
3. If path monitoring needs to be enabled, create a path group: you can add a VWire, VLAN, or virtual router path monitor. For VWire and VLAN, you must specify a source IP the monitor will use to spoof its source. The monitored router must know a route back to the VWire or VLAN. For virtual router path monitoring, the source will be the egress interface closest to the monitored next-hop.

For the secondary member, repeat all of the preceding steps with the following differences:

1. The peer's HA1 IP will be `172.16.0.1`.
2. The peer's HA1 backup IP will be `172.16.1.1`.
3. Set **Device Priority** to `100`.
4. Make sure that if preemption is enabled on the primary, it is also enabled on the secondary.
5. Also make sure the timers are identical: if you changed timers on the primary, set the timers to match on the secondary.
6. In the local control link (**HA1**), use the same interface as the primary, but set the IP to `172.16.0.2`.
7. For the HA1 backup, set the same configuration as the primary but set the IP to `172.16.1.2`.
8. For **HA2**, also use the same interface, and if you need to use an alternate transport mode, use `172.16.0.2`.
9. Make sure all other settings, including path and link monitoring, are identical to the primary member.

You are now able to configure an Active/Passive high-availability pair, the most common form of HA. The next HA mode is more complex but also a little more versatile.

Setting up Active/Active mode

Before embarking on the wonderful journey that is Active/Active, make sure you're taking it for the appropriate reasons:

- To "fix" asymmetric traffic flows
- Because of a requirement to have (floating) IPs active on specific devices unless there is a failover situation (like having a double Active/Passive setup)
- Because Active/Active has a very low tolerance for dynamic routing renegotiation latency when a failure occurs (with both devices up, dynamic routing can renegotiate faster than when a passive device first needs to come online)

While Active/Active is better at handling burst traffic due to the availability of two firewalls, it should be considered as having a lower average throughput than an Active/Passive cluster due to the overhead introduced by remote inspection, where the local device needs to forward entire packets to the remote peer for inspection, rather than performing this operation locally.

To configure Active/Active, follow these steps:

1. In **Device | High Availability**, edit **Setup** and enable **HA**.
2. Set **Group ID**. The actual ID is only important if you need to avoid MAC address conflicts with other firewall clusters in the same broadcast domain.
3. Set the mode as **Active/Active**.
4. Select **Device ID** `0` or `1` (typically, **active-primary** is `0` and **active-secondary** is `1`).
5. Enable **config sync**. Leaving config sync disabled allows for a different configuration on both peers (i.e., different interface IPs, etc.), but requires religious upkeep of shared configuration items as those will not be synchronized either.
6. Set **Peer HA1 IP address** and **Backup peer A1 IP address** (you'll need two small subnets that do not overlap with any used internally, for example, `172.16.0.2` and `172.16.1.2` with subnets `255.255.255.252`).

The election settings are similar to an Active/Passive cluster but serve to determine which member is the active-primary, rather than the Active:

1. In **Election settings**, set a device priority: **active-primary** should have the lowest priority, so set it to `50`.
2. Enable **preemptive** if you require floating IPs to be *sticky* to either cluster member. Keep it disabled to prevent floating IPs from moving back and forth if a cluster member encounters issues.
3. Don't enable **Heartbeat Backup** unless HA1 backup can't be set up. **Heartbeat Backup** uses simple pings to check liveness via the management interface. If HA1 is configured to use the management interface, don't enable **Heartbeat Backup**.
4. HA timers are set to **recommended per platform**, but you can choose **aggressive** for faster failover (but at a cost of overhead), or choose **advanced** to manually change timers and counters. A few interesting counters:
   - **Promotion hold time** is the amount of time the secondary will wait before becoming active after the connection with the primary has been lost.
   - **Hello interval** is the number of milliseconds between hello messages.
   - **Heartbeat interval** is the amount of time between ICMP heartbeat packets.
   - **Flap max & preemption hold timer**: If you enable **preempt**, the firewall will blindly *flap* back to the active state after the preemption hold timer expires. If the original error that caused it to fail still exists, it will fail again. The **flap max** counter will prevent the firewall from repeating this scenario more than the specified number of times, at which time the firewall will go into a "permanently" failed state that can only be recovered via manual intervention.

- **Monitor Fail Hold Up Time** is the amount of time the firewall will wait to fail over once a monitor (path, interface, and so on) has been detected, in the case of an extremely short interruption.
- **Additional master hold time** is used to add even more hold time to **Monitor Fail Hold Up Time.**

5. Click **OK.**

We need to configure the control link so the cluster is able to synchronize configuration and routing **FIB** (**Forwarding Information Base**):

1. Set the interface to the dedicated **ha1-a** link if possible, or the data plane interface you set to type **HA** to be used as the control link, and fill in IP address `172.16.0.1` and subnet mask `255.255.255.252`. Add a gateway if needed, enable encryption (make sure you exported/imported the **HA** keys on both peers), or set the management interface.
2. **Monitor Hold Time** is the amount of time to wait before declaring a failure of the peer when **HA1** connectivity is lost. With Heartbeat backup and **HA1** backup in place, this number can be lowered significantly. If neither backup option is available to you, do not lower this number as a short interruption could lead to a *split-brain* where both peers become active, which is not fun for anyone.
3. Set the interface to **ha1-b**, a dedicated interface, and set IP address `172.16.1.1` and subnet mask `255.255.255.252`, or set the management interface if no alternative interfaces are available.
4. Click **OK.**

The data links need to be configured to synchronize the session, ARP, and MAC tables:

1. Open the **HA2** settings and enable session synchronization.
2. If available, use the HSCI interface; otherwise, set a data plane interface (you can create an aggregate interface and set it to type **HA**, and use the aggregate here as well).
3. If you are able to use the **ethernet** transport mode, there's no need for IP addresses. If you need to use the IP or UDP transport mode, use a third non-overlapping subnet (for example, `172.16.3.1` and subnet mask `255.255.255.252`).
4. Enable HA keep-alive and set it as **split-datapath**. The split datapath option lets both peers take control of their local session and state table if the **HA2** link is disrupted, so they can keep processing local sessions.
5. If you are able to sacrifice another data plane interface, it is recommended to add it as the **HA2** backup interface. The **HA2** backup link is only used if the main **HA2** link goes down or if the keep-alive messages exceed the threshold, and helps prevent **split-datapath** if the main **HA2** link is interrupted.
6. Click **OK.**

The link state should be monitored to ensure the member fails over when an interface goes down. Path monitoring can be added in addition, to ensure a remote router is available to pass traffic.

Access the **Link and Path Monitoring** tab:

1. Enable link monitoring and create a **Link** group.
2. In the **Link** group, add all the interfaces that need to be monitored and set the fail condition to **Any**. A group could be created where *all* interfaces need to be down for the chassis to fail, which could be helpful if you have redundant links and don't need an **HA** failover if just one or part of a link is down.
3. If path monitoring needs to be enabled, create a path group: you can add a VWire, VLAN, or virtual router path monitor. For VWire and VLAN, you must specify a source IP the monitor will use to spoof its source. The monitored router must know a route back to the VWire or VLAN. For virtual router path monitoring, the source will be the egress interface closest to the monitored next-hop.

In Active/Active mode, the **HA3** interface also needs to be enabled to pass along packets for session setup or session owner forwarding, and to synchronize the routing and QoS configuration:

1. Access the **Active/Active Configuration** or **HA Communications** tab.
2. In **Packet Forwarding**, select the HSCI interface if your chassis has one available. Otherwise, you'll want to set up an **AE** (**Aggregate Ethernet**) group of interfaces to carry the **HA3** sessions. The number of interfaces should be scaled to accommodate the expected amount of traffic flowing through a member where the remote peer is assigned the session owner role.
3. Check the boxes next to VR and QoS sync to ensure the routing table and QoS profile selection information are synced:
    - If you intend to run both peers as individual dynamic routing nodes (through dynamic routing such as OSPF or BGP), *disable* **VR Sync / LR Sync**
    - If both peers have different bandwidth available, disable **QoS Sync** and set up individual QoS profiles per member
4. **Tentative hold time** is the time granted to the peer after it recovers from a failure for it to rebuild its dynamic routing table before assuming its normal active role. If no dynamic routing is used, you can disable this timer.

5. **Session Owner Selection** will have an enormous impact on your device load depending on which type of deployment you choose:
    - If you intend to have the primary firewall be the master device of all sessions and only need the secondary online for dynamic routing, or as an asymmetric routing solution, you can set the session owner to **Primary**: the primary device will perform all Layer 7 session scanning while the secondary will simply receive packets and hand them over to the primary for processing, and participate in dynamic routing.
    - If both members are intended to take an active role, select **first packet** as the packet processing setting.
6. With **Session Setup**, you can also select which member is responsible for all Layer 2 through Layer 4 (routing, NAT translation, and so on) operations by selecting **Primary Device**, **First Packet**, or a load balancing algorithm like **IP Modulo** or **IP Hash**:
    - **IP Modulo** distributes the sessions based on the parity of the source IP address.
    - **IP Hash** distributes the sessions based on a hash of the source IP address, or the source and destination IP addresses. A hash seed can be added to increase randomization.
7. Click **OK**.

You can also add Active/Active virtual addresses. These are floating addresses that can be configured to stick to a specific member or float about, or be shared, between the two peers:

- A floating IP with a priority set to either member will stick to one member unless that member encounters a failure, at which time it will fail over, similar to the Active/Passive setup.
- A floating IP that is bound to the active master also acts similarly to the Active/Passive configuration as it will only transfer to the secondary member if the active master goes offline or non-functional.
- ARP load sharing will leverage ARP in such a way that depending on the source IP (IP Modulo or IP Hash), a client will receive ARP replies from either member 0 or member 1 for a gateway IP, effectively loading balancing sessions over both members. The firewall needs to be in the same broadcast domain as the client for this option to work (for example, a downstream router and hosts behind it will always talk to the same peer).
- A floating IP with priorities set will be active on both peers but only the peer with the highest (available) priority will respond to ARP requests, while a floating IP bound to the Active Primary will not “exist” on the active secondary. In other words, if there are priorities set, the lowest priority member can still accept packets for the floating IP if external factors force a packet to the peer.

You determine the behavior of each virtual address, as you can see in the following screenshot:

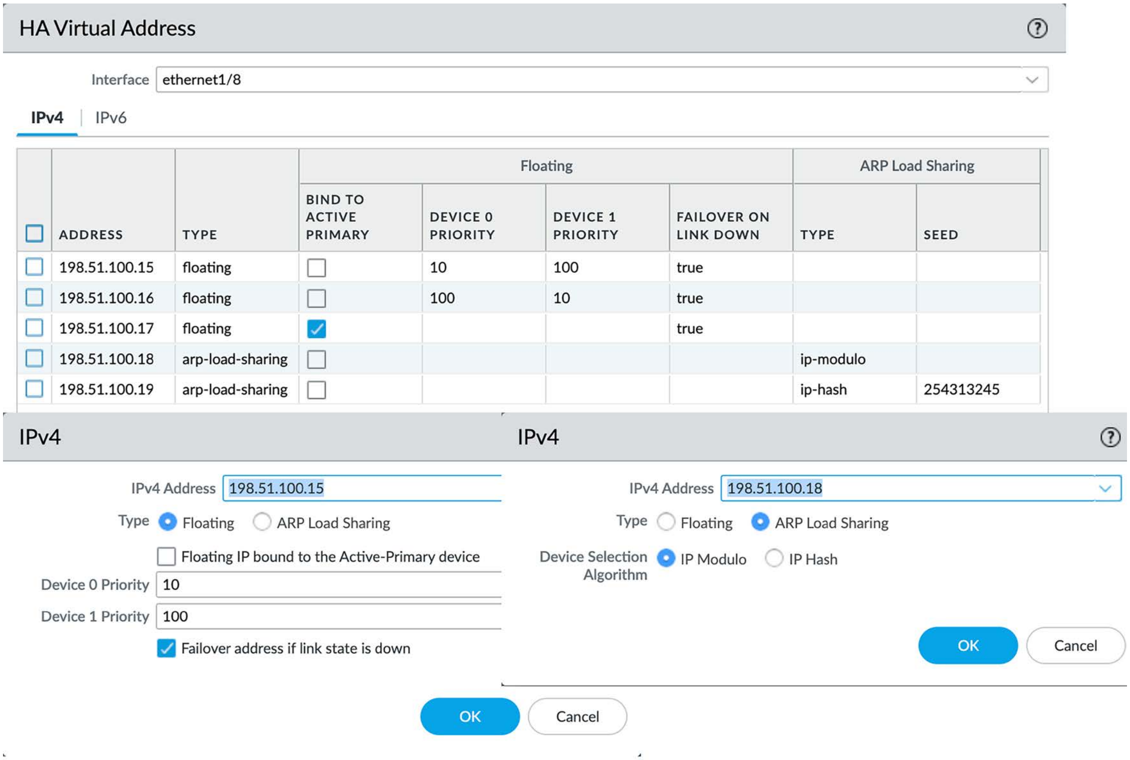


Figure 5.14: Active/Active virtual addresses

NAT rules in Active/Active configuration have an additional tab called **Active/Active HA Binding** where you need to decide which member a NAT policy sticks to, as you can see in the following screenshot. This needs to correspond to the virtual IP configuration in the HA configuration to ensure NAT is applied to the appropriate member that owns an IP address. The **primary** option is used when the primary member is chosen for the session setup. If either member has a lower priority for a certain IP, or if **dynamic IP and port** (**DIPP**) is set up, select that member's ID (0 or 1). When using ARP load sharing, select **both**:

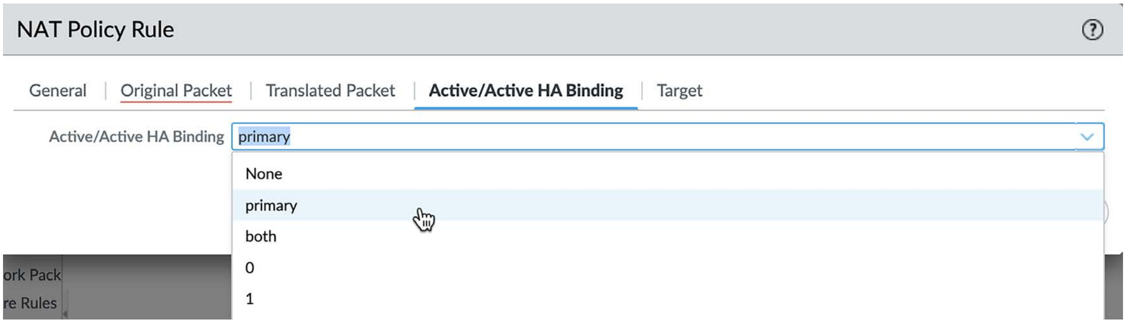


Figure 5.15: NAT in an Active/Active configuration

Make sure you duplicate the configuration on both peers, as Active/Active requires both sides to be configured individually for floating IP ownership.

It is best practice that the HA1 communication be encrypted as this can prevent sensitive data from being exposed: the HA1 link shares configuration, User-ID, and routing information.

HA1 encryption

Because the HA1 interface shares very sensitive information with the cluster peer, it is recommended to encrypt all traffic flowing between the two firewalls. Before enabling this feature, the HA keys of both peers first need to be exported and imported on the peer device. The export and import options are available from **Device** | **Certificate Management** | **Certificates**, as illustrated in the following screenshot:

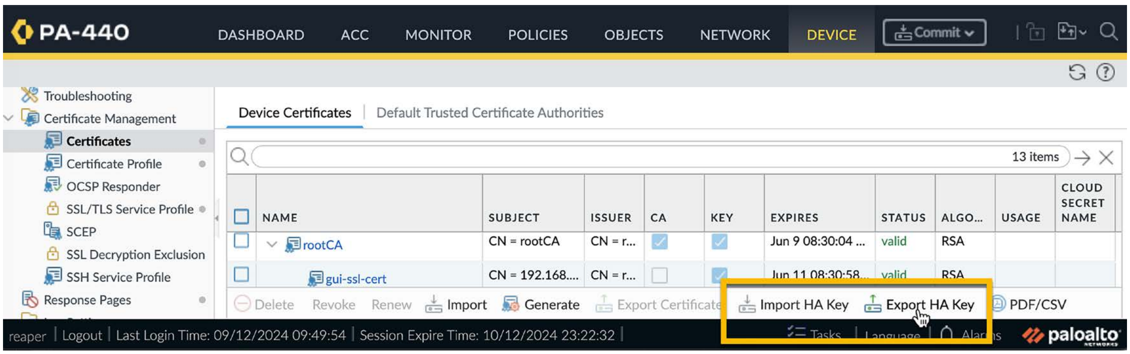


Figure 5.16: Import and Export HA Key

The last step is to enable encryption on the HA1 configuration. Go to **Device** | **High Availability** | **HA Communications** and in the HA1 configuration, check the box for **Encryption Enabled**, as illustrated in the following screenshot:

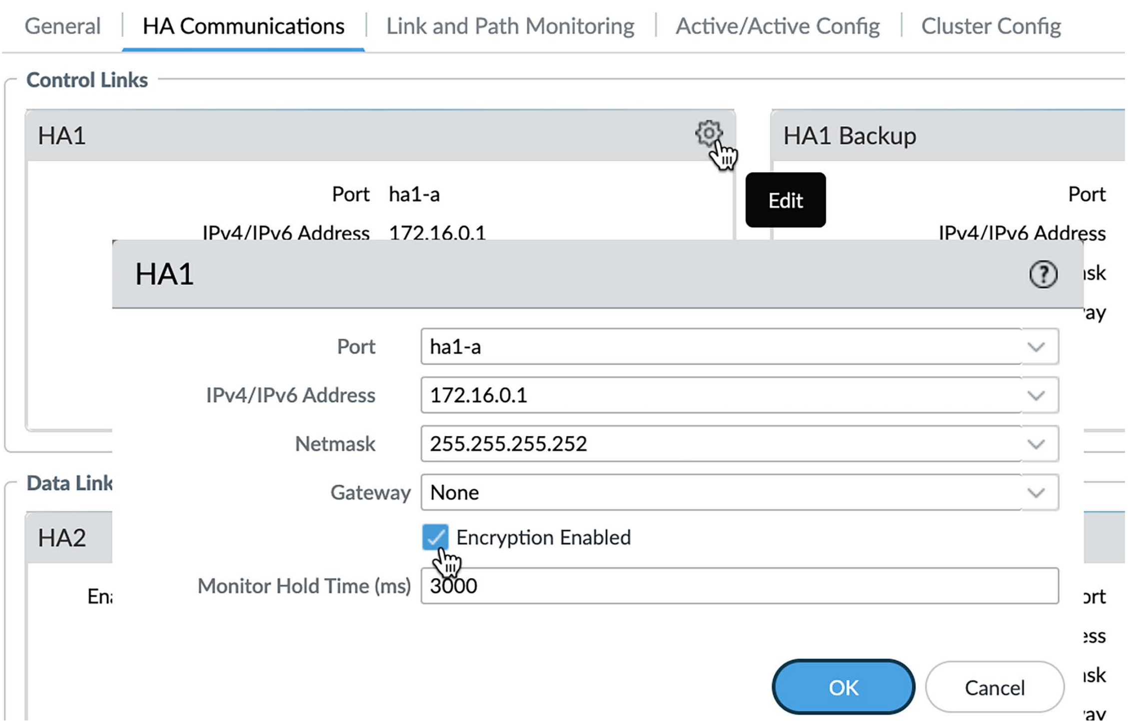


Figure 5.17: Enable HA1 encryption

When this configuration is committed, it will disconnect the HA1 link as one side will use encryption while the other doesn't. Ensure this change is committed during a maintenance window or while the passive member is in a suspended state.

You are now able to set up a cluster and decide whether you want a regular Active/Passive deployment or need the more complex Active/Active flavor. You're also able to implement an often forgotten but very critical aspect of finalizing the HA configuration by encrypting the very sensitive HA1 link.

In the next section, you will learn how to set up virtual systems so you can segregate networks, or customers, into a logical firewall instance.

Enabling virtual systems

Enabling **virtual systems** (**VSYS**) on a firewall makes it into a multi-tenant system. Each VSYS represents a virtual firewall instance that can operate independently while sharing the resources available on the host system. The host system still retains control over all networking functions (interfaces and their configurations, routing tables, IPSec and GRE tunnels, DHCP, DNS proxy, and so on) and the management configuration.

Each VSYS can be assigned its own (sub) interfaces and routing can either be taken care of at the system level or by creating virtual routers and assigning them to each VSYS.

> **Important note**
>
> By default, each firewall creates its objects in `vsys1`. This is the native VSYS even for devices that do not support multi-VSYS. Objects created in `vsys1` or any other VSYS will not be visible to other VSYSes unless their location is set as `shared`.

Starting from PAN-OS 11.1.3, multi-vsys is supported on PA-400, PA-1400, PA-3200, PA-3400, PA-5200, PA-5400, and PA-7000 platforms and VM series. On earlier PAN-OS versions, only the larger physical platforms (PA-3220 and up) support multi-VSYS mode. The number of virtual systems supported also varies per device, with the largest platform supporting up to 225 virtual systems.

To enable multi-VSYS, you will first need to activate a VSYS license and import it onto the device. Then, in **Device** | **Setup** | **Management** | **General settings**, you can enable **Multi Virtual System Capability**. Enabling the option and clicking **OK** will pop up a warning that this action will cause the system to commit as shown here:

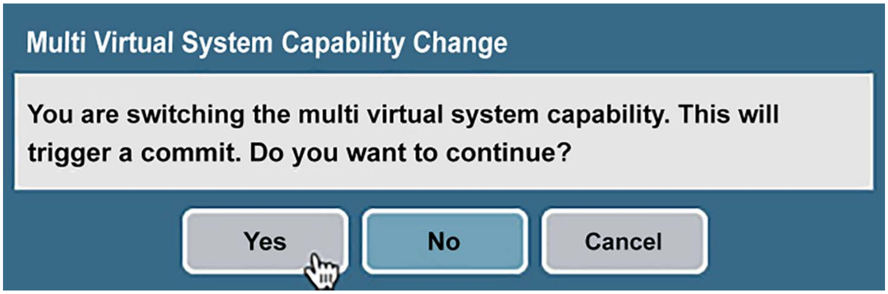


Figure 5.18: Multi Virtual System Capability Change commit warning

Once the feature is enabled, two new menu items will appear under **Device**:

- **Virtual Systems**: Where you add a new VSYS
- **Shared Gateways:** This is an aggregation zone in case multiple VSYSes need to use the same ISP uplink (commonly used in a shared services environment)

After enabling the capability, the first thing to do is to create a new VSYS.

Creating a new VSYS

When you create a new VSYS, there's not a lot you can configure yet as the interfaces, VLANs, VWire, and virtual routers will most likely still need to be created. But you can enable a "visible VSYS."

A **Visible Virtual System** allows you to select which VSYS can be reached by another VSYS. This can be useful if you need to segregate some network segments but need to allow some routing. Keeping visibility disabled will enforce segregation.

It is important to note that each VSYS can have its resources limited so it doesn't flood out other VSYSes by overconsuming the system's total available resources. As seen in the next screenshot, in the **Resource** tab of the VSYS, the total amount of sessions can be limited, the number of VPN tunnels can be limited, and the number of rules this VSYS can hold can be limited. Each physical host has a finite number of rules and sessions it can maintain, so setting limitations helps maintain order when different administrators are put in charge of setting up their own rule bases:

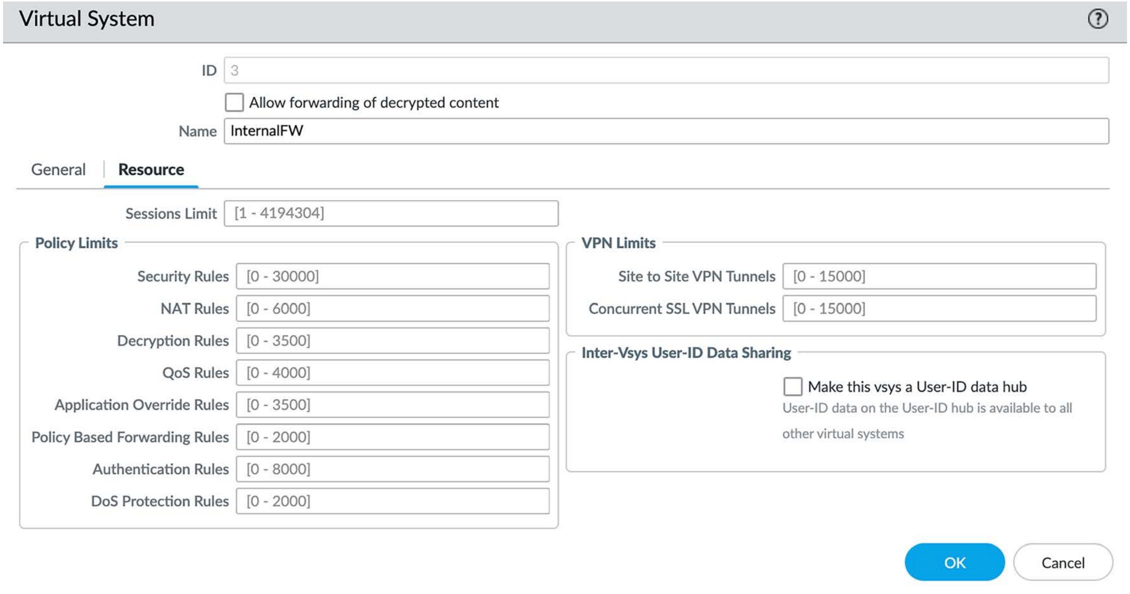


Figure 5.19: VSYS resource limitation

Next up, you will need to configure all the interfaces, zones, and the virtual router(s) as if setting up a factory-new device:

1. In **Network** | **Zone**, create new (internal, external, DMZ, and so on) zones and set the new VSYS as **Location**.
2. In **Network** | **Virtual Router**, create a new VR and add the appropriate routing configuration you will be using in the new VSYS. Click **OK** and then add it to the appropriate VSYS by clicking the hyperlinked **none** next to the VSYS on the main page, as seen here:

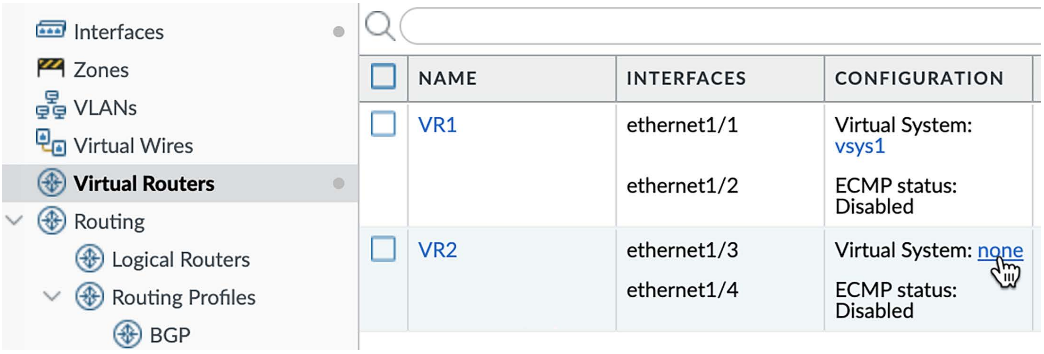
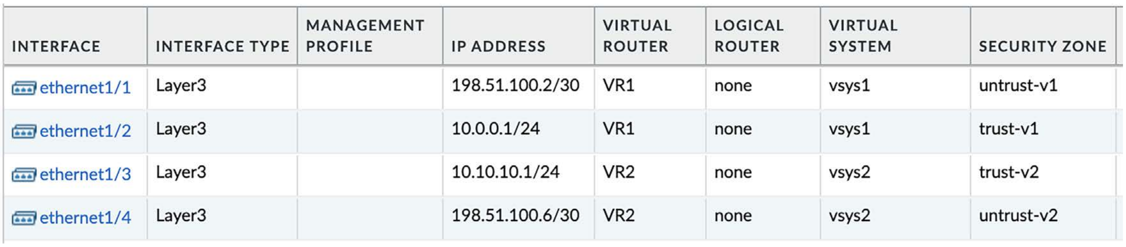


Figure 5.20: Adding a new virtual router to a VSYS

3. If you need a VWire in the new VSYS, create it in **Network | Virtual Wires**.
4. In **Network | Interfaces**, configure the interfaces you will add to the VSYS so they are themselves set to the right VSYS and are using the VSYS VR and zones.

You should now have interfaces set similarly to the following screenshot, with **ethernet1/1** and **1/2** set to **vsys1**, using the VR in **vsys1**, and zones in **vsys1**, while **ethernet1/7** and **1/8** are configured in **vsys2** with the VR and zones in **vsys2**:



INTERFACE	INTERFACE TYPE	MANAGEMENT PROFILE	IP ADDRESS	VIRTUAL ROUTER	LOGICAL ROUTER	VIRTUAL SYSTEM	SECURITY ZONE
ethernet1/1	Layer3		198.51.100.2/30	VR1	none	vsys1	untrust-v1
ethernet1/2	Layer3		10.0.0.1/24	VR1	none	vsys1	trust-v1
ethernet1/3	Layer3		10.10.10.1/24	VR2	none	vsys2	trust-v2
ethernet1/4	Layer3		198.51.100.6/30	VR2	none	vsys2	untrust-v2

Figure 5.21: Interfaces configured on two different VSYSes

Each interface can be set to its own VSYS even if it is a subinterface. One physical interface can have multiple subinterfaces all assigned to a different VSYS.

Important note

All VSYSes can have the same zone names (e.g., "trust" in both vsys1 and vsys2) as each system is segregated from the others, but this could lead to administrator confusion, so it is recommended to use a different naming convention for each VSYS. For a shared hosting environment where each customer only has access to their own VSYS, it could help to set every customer up with a `trust`, `untrust`, and `dmz` zone for ease of use.

When hosting multiple logical firewalls, there may also be administrators that only need access to a specific VSYS, rather than the whole system.

Administrators in a multi-VSYS environment

With the activation of a multi-VSYS, new administrator types become available that are restricted to the confines of the VSYS: **Virtual System Administrator** and **Virtual System Administrator (read only)** are limited to accessing only a specific VSYS and share the following abilities:

- They are able to see but not edit all the device configuration, except anything that relates directly to other VSYSes
- They can only see logs and ACC data related to their own VSYS
- They can create, edit, and delete rules but only for their own VSYS
- They are not able to see any interface-related configuration (interfaces, VWires, VRs, VLANs, and so on) except the zones attached to their VSYS, and certain menu items are removed, as shown in the following screenshot:

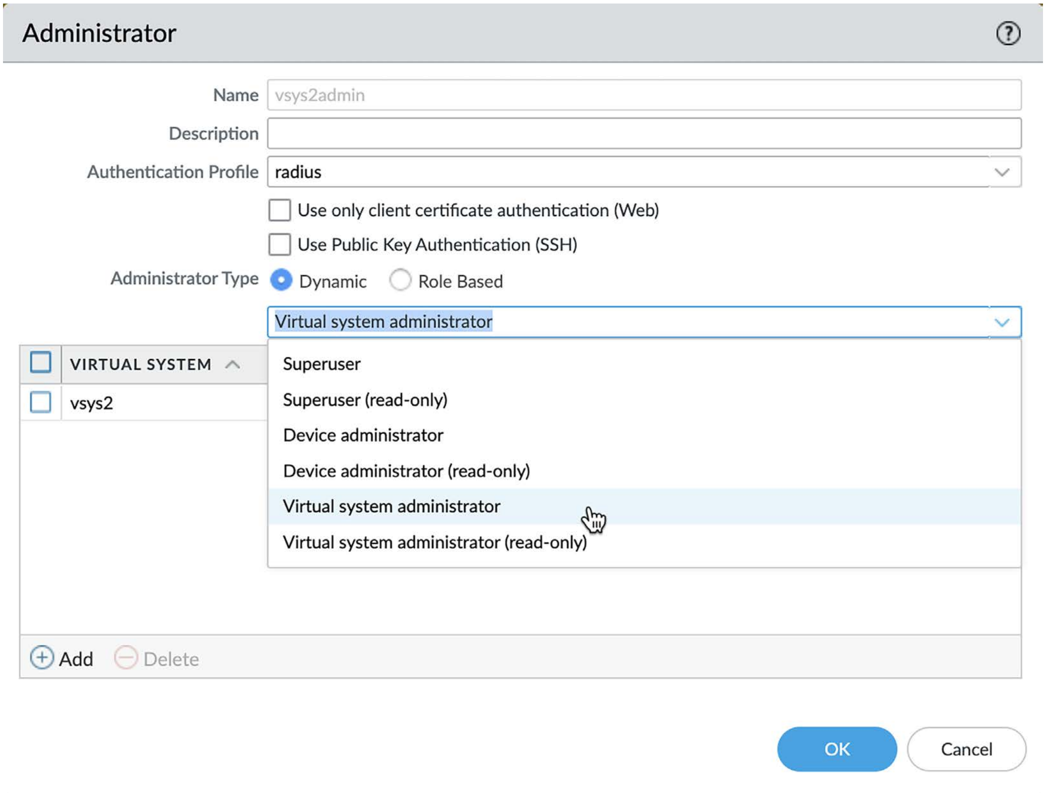


Figure 5.22: Restricting view of a VSYS administrator

Access can be restricted even further by setting up a VSYS restricted admin role that limits the access of the administrator to the VSYS but you can also remove tabs and menu items, and allow administrators read-only or edit privileges in individual menu options. In the following example, you can see the **Dashboard**, **ACC**, and **Device** tabs have been removed. The admin is unable to see logs because the log view is restricted to **vsys2** only:

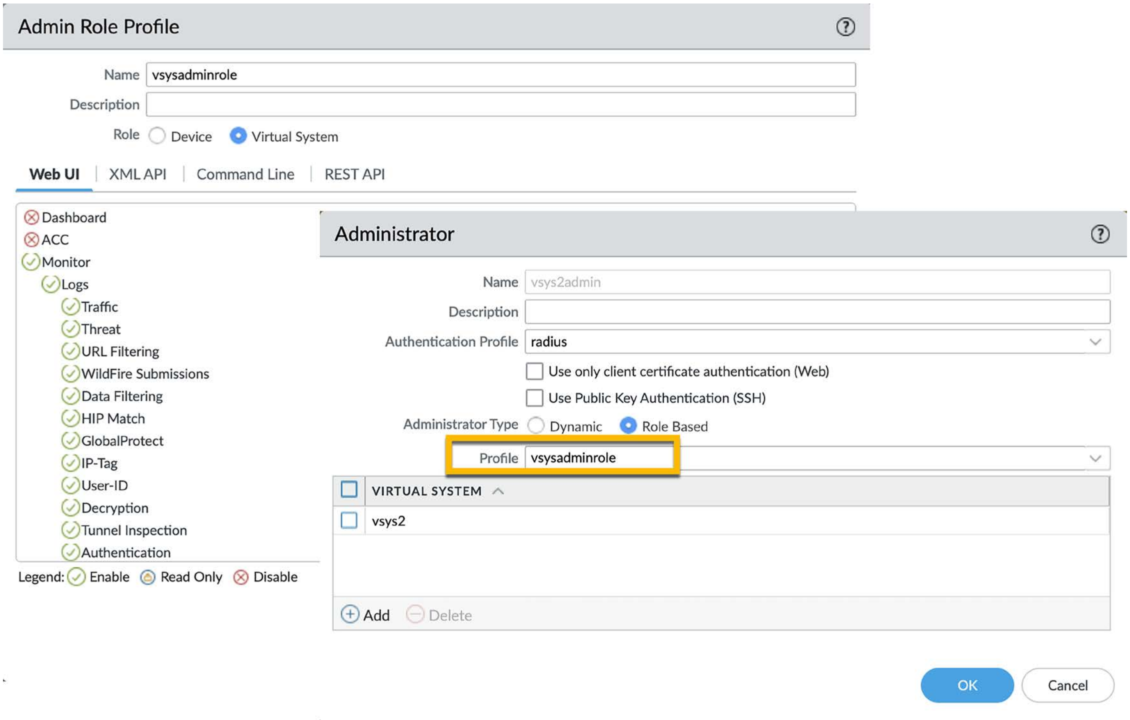


Figure 5.23: Virtual System admin role

After you set up two or more fully segregated logical firewall instances, the need may arise to have certain hosts or subnets communicate with each other even though they belong to a different VSYS.

Inter-VSYS routing

Because VSYSes are not aware of each other's existence, some steps are needed before sessions can be set up between VSYSes. Each VSYS will see the other VSYS as existing in the **External** zone, which is a special area for inter-VSYS routing:

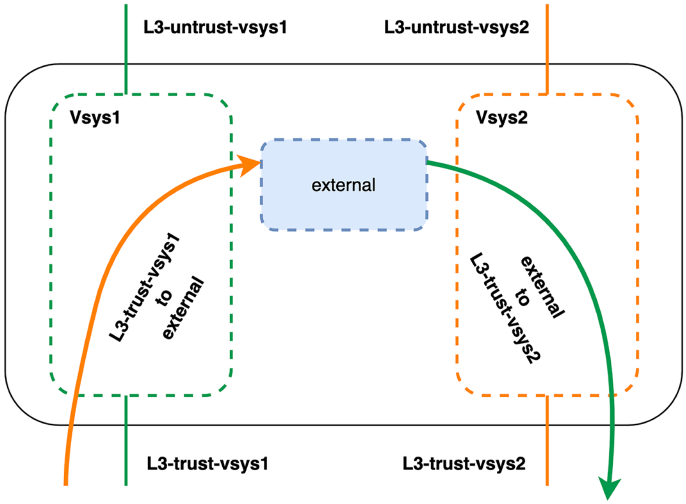


Figure 5.24: Inter-VSYS routing

You need to follow these steps:

1. Enable the visibility of the other VSYS in *each* VSYS profile by checking the box of the appropriate VSYS in the general tab of every VSYS (e.g., check **vsys2** in the vsys1 **General** tab, and check **vsys1** in the vsys2 **General** tab).

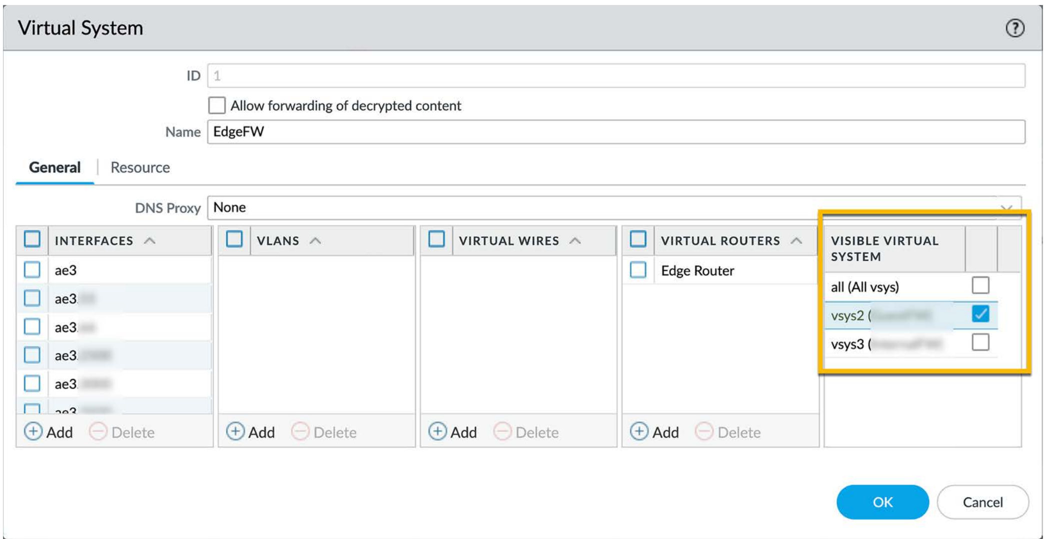


Figure 5.25: Inter-VSYS visibility

2. Create a new zone called `out-to-vsys2`:

   - Set it to location **vsys1**
   - Set it to type **External**
   - Add **vsys2** to the VSYS selection

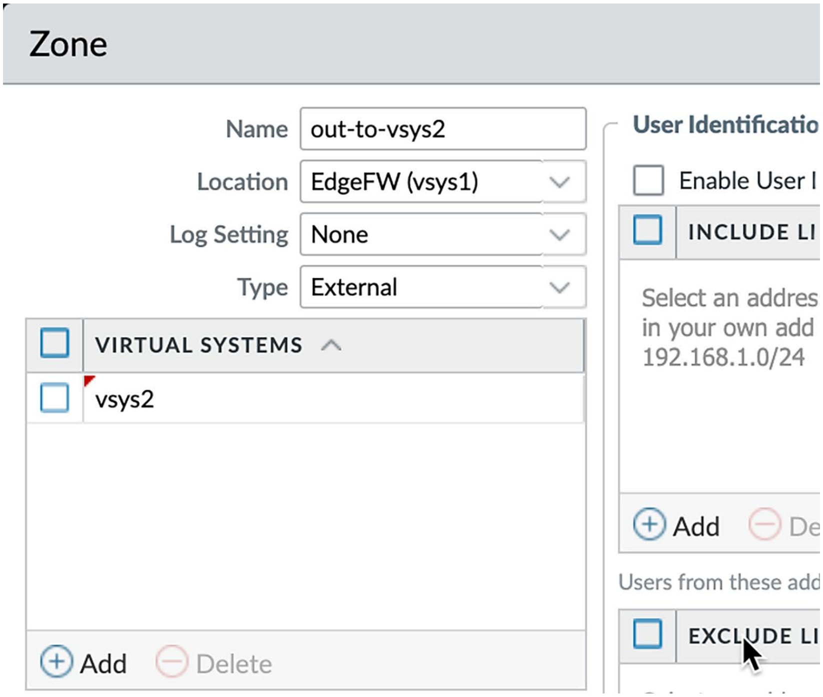


Figure 5.26: Inter-VSYS zone

3. Create a new zone called `out-to-vsys1`:

   - Set it to location **vsys2**
   - Set it to type **External**
   - Add **vsys1** to the VSYS selection

4. On the virtual router in **vsys1**, create a new static route:

   - Set the name to `vsys2-subnet`
   - Add the destination subnet of **vsys2** (`10.1.0.0/24`)
   - Leave the interface as **none**
   - Set **Next Hop** to `Next VR` and assign the **VR** in **vsys2**
   - Click **Ok**

5. On the **VR** in **vsys2**, create a new static route:
   - Set the name to `vsys1-subnet`
   - Add the destination subnet of **vsys1** (`10.0.0.0/24`)
   - Leave the interface as **none**
   - Set **Next Hop** to **Next VR** and assign the **VR** in **vsys1**
   - Click **Ok**
6. In **Policies** | **Security**, create a security rule for each direction and the applications that need to be able to be used in the session, plus security profiles.

   For sessions flowing from **vsys1** to **vsys2**, do the following:
   - In **vsys1**, create a security rule from **L3-trust-V1** to **out-to-vsys2**
   - In **vsys2**, create a security rule from **out-to-vsys1** to **L3-trust-V2**

   For sessions flowing from **vsys2** to **vsys1**, do the following:
   - In **vsys2**, create a security rule from **L3-trust-V2** to **out-to-vsys1**
   - In **vsys1**, create a security rule from **out-to-vsys2** to **L3-trust-V1**

While enabling inter-VSYS routing can help solve some interesting challenges, it also prevents said traffic from being offloaded (hardware offloading on the larger platforms), so will cause additional load on the system. Take this into account when considering routing externally versus inter-VSYS.

You are now able to create completely separate environments on the same hardware and even enable traffic to flow between these instances, but in some cases, there may be restrictions on how many external interfaces or IP addresses are available with the ISP. This can be overcome by enabling a shared gateway.

Creating a shared gateway

Similar to inter-VSYS routing, a shared gateway is a VSYS that is intended to provide internet access to multiple VSYSes. This allows you to keep each VSYS separate while still using the same internet connection. There are several steps involved in this process. First, create a new shared gateway in **Device** | **Shared gateways**:

1. Assign ID `1`.
2. Provide an easy-to-identify name.
3. If a DNS proxy configuration is needed, set one.

Next, in **Network | Zones**, configure the zones that will be used on the egress interface:

1. Create a new zone and name it `SG-untrust`.
2. Set it to type **Layer3.**
3. Set the location to **Shared gateway** (**sg1**).
4. Create another zone and name it `SG-to-vsys1`.
5. Set it to type **External.**
6. Set the location as **Shared Gateways** (**sg1**).
7. Add **vsys1** to the **Virtual Systems**.
8. Repeat *steps 4-7* for the additional VSYS.
9. In each **VSYS**, also make a new zone set to type `External` that has **sg1** (**SharedGW**) as the VSYS. Call this `external-SG1`.

Then you will need a virtual router. Go to **Network** | **Virtual Routers**:

1. Create a new virtual router and call it `SharedVR`.
2. If you will use a static IP ISP link, create the static route for the default route (`0.0.0.0/0` out of the egress interface to the ISP router).
3. Add routes to the other VSYS by setting the destination subnet, setting **Next Hop** to **Next VR,** and assigning the appropriate VSYS virtual router (for example, `10.0.0.0/24` set to **Next VR** to `v1-default`).
4. Do *not* set the VSYS assignment; leave it as **none**.
5. In the other VSYS virtual routers, create a default route that points to `SharedVR` (for example, `0.0.0.0/0` set to **Next VR** equal to `SharedVR`).

Then, configure the interface in **Network | Interfaces**:

1. Open the interface you will use for the shared gateway.
2. Set it to interface type **Layer3.**
3. Assign VSYS **SharedGW** (**sg1**).
4. Assign zone **SG-untrust.**
5. Assign VR **SharedVR.**
6. Access the **Ipv4** tab and set the IP configuration (static IP or dynamic configuration).

Lastly, we need to create policies:

1. Security policies are created on the individual VSYS and will look as follows:

   `L3-trust-V1` to `external-SG1` with the desired applications, services set to **application-default**, and a security profile group.

2. NAT is set up on the shared gateway; you can use the individual `SG-to_vsysX` to create individual NAT rules if you want to assign each VSYS its own NAT address or put all the zones in the source of a single hide-NAT rule.

   An inbound NAT will be configured as follows:

   - From SG-untrust to SG-untrust, with the public IP as **Destination**, translate to the appropriate vsysX IP. Routing will take care of delivery to the appropriate VSYS. On the VSYS, a security policy will need to be configured.
   - From **external-SG1** to **L3-dmz-V1** to the pre-NAT destination IP, allowing the appropriate applications, and using a security profile group.

Important note

If an individual VSYS does not need its own routing table, you can run the entire system on a single VR that is set to `none` in the VSYS selection.

You are now able to create logical firewall instances and leverage a shared gateway to provide internet access via a single ISP uplink. In the next section, we'll learn about managing certificates on the firewall.

Managing certificates

Certificates are used for all kinds of useful things like decrypting TLS/SSL traffic, authenticating users, and ensuring an SSL VPN is secure. When performing SSL decryption, the firewall needs to have access to a certificate the client will trust so it doesn't cause a certificate warning in the browser. The firewall will also need to know which root certificate authorities are trustworthy and which ones *should* cause red flags to pop up. It will need to provide a valid certificate when a VPN client connects to the portal or gateway and the administrator should ideally also be greeted by a friendly lock in the address bar rather than a warning page. All these certificates can be managed from the **Device** | **Certificate management** | **Certificates** menu. As you can see from *Figure 5.27*, certificates in a chain are automatically sorted so you have immediate visibility of what their relationship is. Several certificates also have a **usage**.

A **Trusted Root CA Certificate** is an imported or externally available root **certificate authority** (**CA**) that the firewall should treat as trusted. This could be, for example, an internal CA that is not an internet root CA that has signed internal server certificates that the firewall might encounter while performing forward decryption:

- **Forward Trust Certificate** is the certificate used in SSL decryption and will act as the intermediary for any website visited by the client.
- **Forward Untrust Certificate** is a faulty certificate on purpose (this one should *not* be installed on the clients as a trusted root CA) as it is intended to cause a certificate warning on the client side while still decrypting the session. This certificate is triggered whenever the visited site's root or intermediary CA is not in the **Trusted Root CAs**, has expired, or has some other defect that makes it untrustworthy.
- **Certificate for secure Syslog** can be used to secure syslog forwarding.

Other certificates may include GlobalProtect portal and gateway certificates, web server certificates (with the private key) so the firewall can perform inbound SSL decryption, and a certificate for the firewall web interface:

Device Certificates | Default Trusted Certificate Authorities

13 it

NAME	SUBJECT	ISSUER	CA	KEY	EXPIRES	STATUS	ALGORIT...	USAGE
saasedl-root	C = US, O = Google ...	C = US, O...	✓		Jun 22 00:00:00 203...	valid	RSA	
saasedl	C = US, O = Google ...	C = US, O...	✓		Sep 30 00:00:42 202...	valid	RSA	
	C = BE, O = ICT, CN...	palo-inter...			Jun 3 06:12:44 2023 ...	expired	Elliptic Cu...	
noip-CA	C = US, O = DigiCer...	issuer=C ...	✓		Nov 6 12:23:45 2027 ...	valid	RSA	
rootCA	CN = rootCA	CN = root...	✓	✓	Jun 9 08:30:04 2033 ...	valid	RSA	
gui-ssl-cert	CN = 192.168.0.2	CN = root...		✓	Jun 11 08:30:58 202...	valid	RSA	
PANgurusCA	CN = PANgurusCA	CN = PA...	✓	✓	Jun 28 10:55:36 204...	valid	RSA	Forward Trust...
19216806	CN = 192.168.0.6	CN = PA...		✓	May 3 11:55:09 2027...	valid	RSA	
pangurus-ddns	CN =	CN = PA...		✓	May 3 11:54:33 2027...	valid	Elliptic Cu...	

Figure 5.27: Common certificates on a firewall

The **Default Trusted Certificate Authorities** tab contains about 400 of the most common public root certificates the firewall implicitly trusts. This list is updated with content updates but if any are missing, these can be imported to the **Device Certificates** tab and set as **Trusted Root Certificate**.

As part of User-ID and GlobalProtect, certificate profiles (**Device** | **Certificate management** | **Certificates Profiles**) can be leveraged to identify users. As you can see in the following screenshot, in **Certificate Profile**, you can indicate which CA certificate should have been used to sign the received client certificates, which field to use to identify the user, and the (NetBIOS) domain to map the user to, whether **OCSP** (**Online Certificate Status Protocol** host) will be used and which host to poll, and if certain certificate conditions should lead to a block action:

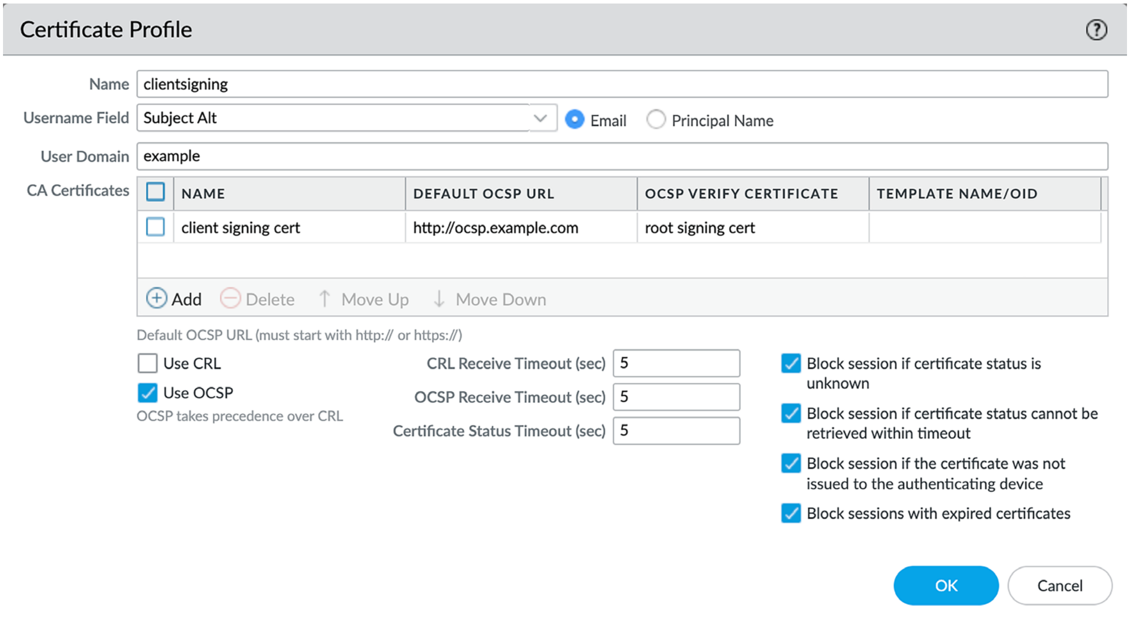


Figure 5.28: Certificate Profile

The SSL/TLS service profiles are used for all web interfaces (the GlobalProtect portal, gateway, captive portal, and the firewall management interface) to set the minimum and maximum TLS version, and to restrict which key exchange, encryption, and authentication algorithms are supported. As shown in the following example, always set **TLSv1.2** as the minimum version and disable SHA1:

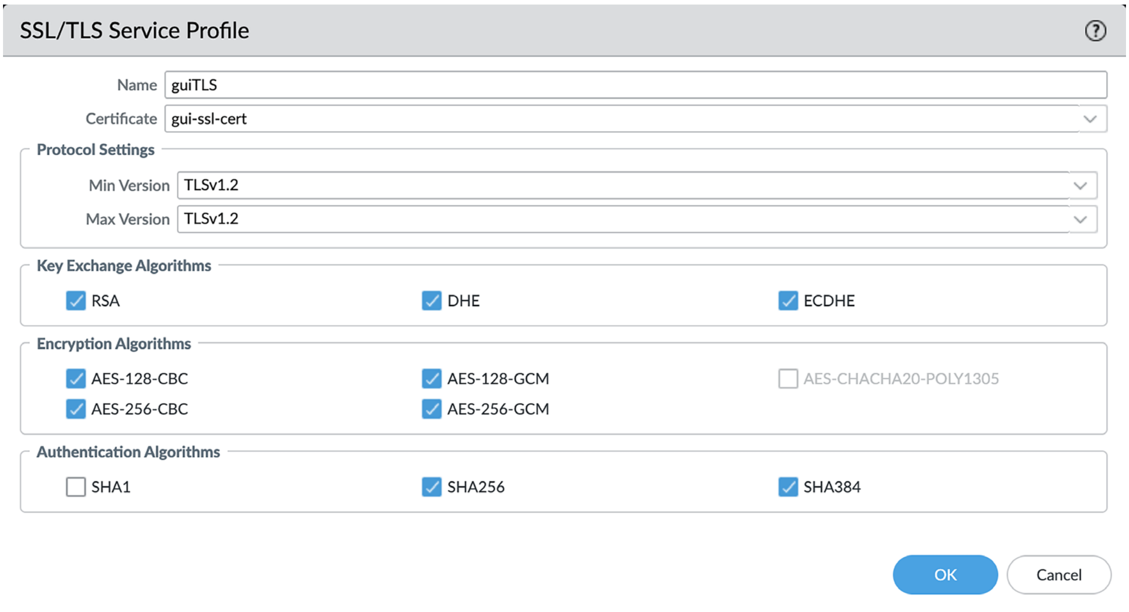


Figure 5.29: SSL/TLS Service Profile

A **Simple Certificate Enrollment Protocol** (**SCEP**) profile can be created if your external CA supports SCEP. This makes generating new client, portal, and other certificates much easier as you simply create a new request for a certificate and the SCEP does all the work for you. The CA server will return a certificate with all the bells and whistles with little input from you:

- If your SCEP enrollment server requires it, you can select **fixed** for a simple password or **dynamic** for an OTP deployment (the OTP is handled between the firewall and CA and doesn't require interaction from you).
- You need to set the **Server URL**, **CA-IDENT Name**, and which type of certificate this SCEP profile will be used for. The variables are `$USERNAME`, `$EMAILADDRESS`, `$HOSTID`, and `$UDID`.
- Set the cryptographic preferences and the SSL authentication certificates if the CA is on HTTPS (not required if the CA is still on HTTP).

If you create a SCEP profile, it should look similar to this:

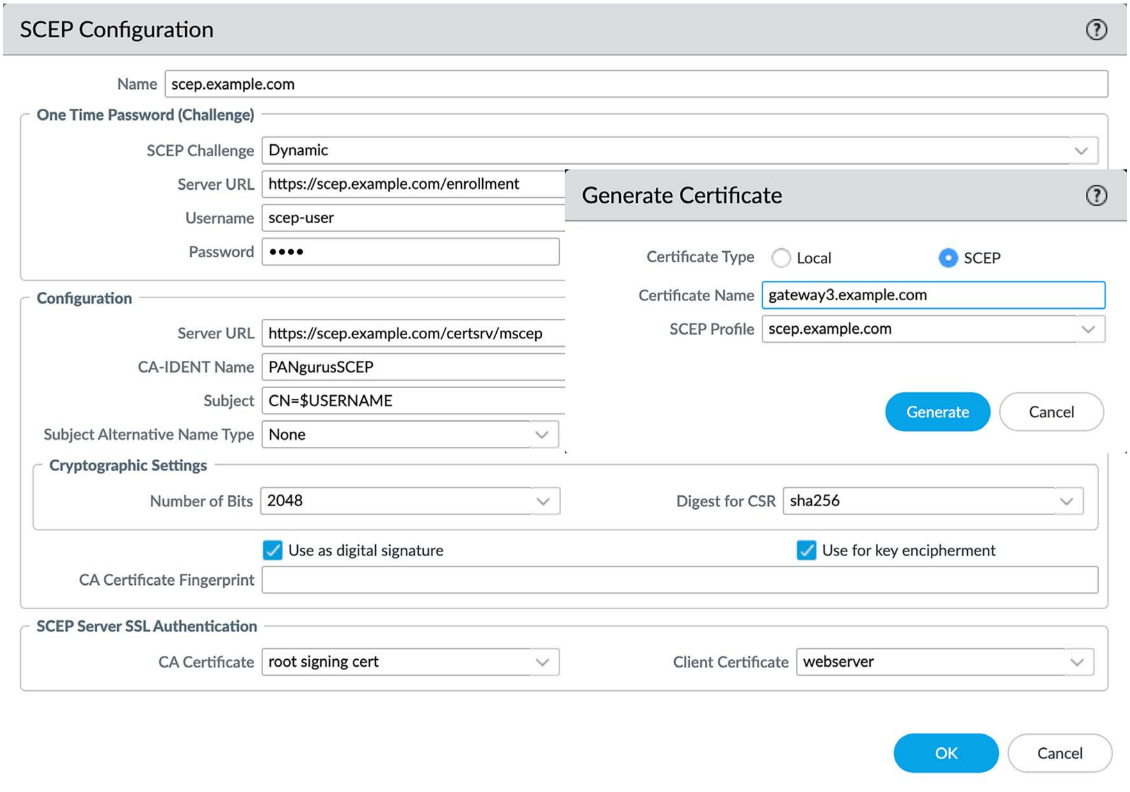


Figure 5.30: SCEP profile and certificate generation using SCEP

To generate a **Certificate Signing Request** (**CSR**) to have a certificate signed by an external authority, simply generate a new certificate in **Device** | **Certificate Management** | **Certificates** and select **External Authority (CSR)** in the **Signed By** field, as illustrated here:

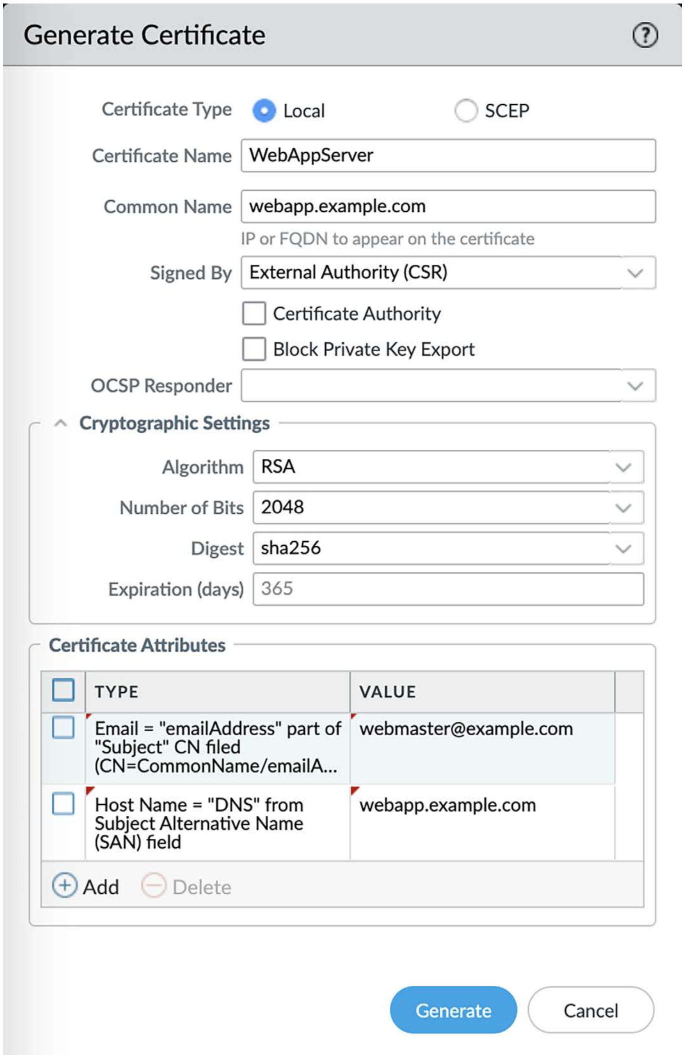


Figure 5.31: Certificate signing request

This will result in a CSR request file that can be exported and uploaded to the certificate signing authority or PKI server. When the CSR response file is generated, make sure to rename it to exactly match the CSR name (i.e., **WebAppServer** as illustrated in *Figure 5.32*) before importing it to the firewall:

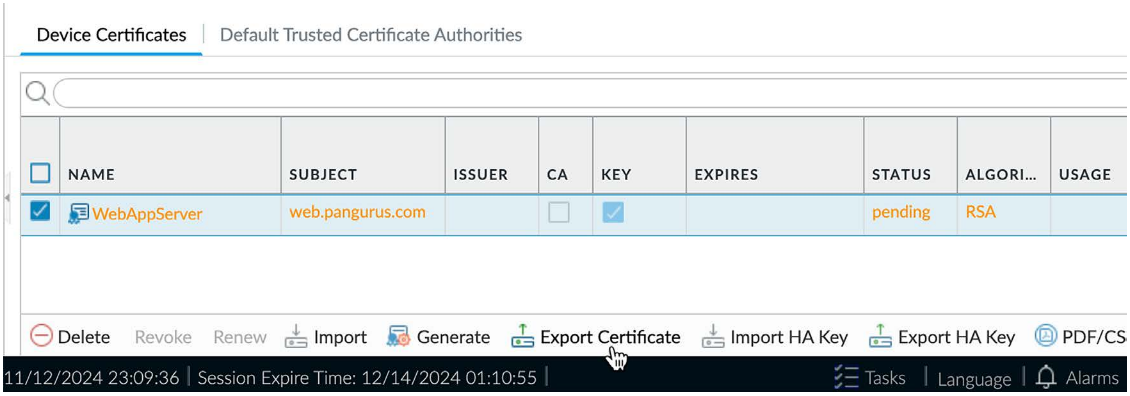


Figure 5.32: Certificate signing request exportable file

Lastly, in the **Certificate Management** menu, you can also set **SSL Decryption Exclusions**, in case you want to manually prevent a specific website from being decrypted:

- If a website is using an unsupported certificate (for example, a partner that still needs to replace a legacy certificate).
- If a pinned certificate is used: certificate pinning restricts which certificates are considered valid for a specific website, thwarting the man-in-the-middle certificate switch used by SSL decryption.
- If client certificate authentication is in place.

In the following screenshot, you can see that the exclusions list is already prepopulated with hostnames that are known to use certificates or methods that cannot be decrypted:

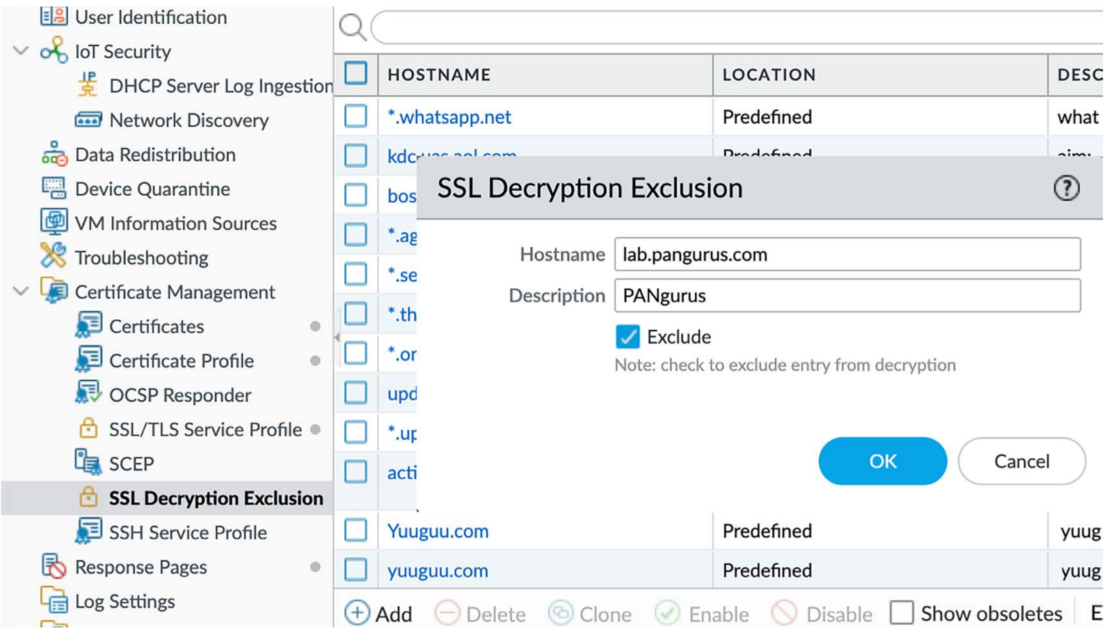


Figure 5.33: Decryption exclusions

SSL Decryption Exclusion will disable decryption for the hostname at the system level. If more granular control is desirable, you can use no-decrypt rules with a URL list.

Additional information regarding decryption issues, including troubleshooting, can be found in the **Monitor** | **Decryption** log.

With the information you just learned, you should be able to ascertain which type of certificate (self-signed, private, or public) you will need to achieve any goal and how to properly store and manage them.

Summary

In this chapter, you learned how to configure the firewall so that it is able to work with DHCP-enabled ISPs and how to serve IP addresses to clients on local networks or relay DHCP for an internal server. You also learned how to set the firewall as a DNS proxy and ensure internal hosts resolve domain names efficiently and securely. You are now able to set up HA clusters in both Active/Passive and Active/Active modes, understand the differences and implications of both modes, and know how to manage and maintain certificates. In a multi-tenant or segregated environment, you can leverage virtual systems to create multiple instances on a single hardware platform.

If you're preparing for the PCNSE, remember how to manage certificates and how and why to create a TLS profile. Take note of what the key differences are between Active/Passive and Active/Active and what all the HA interfaces are used for.

In the next chapter, we'll take a closer look at the various methods to identify users and how group mapping can help build policies that enforce **role-based access control** (**RBAC**).

6

Identifying Users and Controlling Access

In this chapter, we will be learning about **User Identification** (**User-ID**) and the various ways in which we can intercept credentials or have users identify themselves. Once they're identified, their user-to-IP mapping can be leveraged to control which resources they can access. User-based reports can also be generated to keep track of users' habits or review incidents. In addition, we will link user-to-IP mappings to group membership so we can apply role-/group-based access control. This will help us to identify groups of users so they can access only the resources they need while roaming without the need for network segmentation or static IP addresses.

In this chapter, we're going to cover the following topics:

- User-ID basics
- Configuring group mapping
- The Cloud Identity Engine
- Setting up a captive portal
- Using APIs for User-ID
- User credential phishing prevention

By the end of this chapter, you'll be able to leverage and enforce identity-based access controls so security rules no longer depend on IP subnets, which are easily bypassed.

Technical requirements

This chapter requires a working knowledge of **Active Directory** (**AD**) and **Lightweight Directory Access Protocol** (**LDAP**), as we will be collecting information from, and making changes in, AD and setting up an LDAP connection to collect user group membership information.

User-ID basics

In this section, we will learn how to set up the basics needed to identify users by preparing AD and configuring the agent/agentless configuration to collect user-to-IP mappings.

One universal truth is that for User-ID to work, the interface that receives connections from the users that need to be identified needs to have User-ID enabled in its zone. Navigate to **Network** | **Zones** and click **Enable User Identification** in the appropriate zone, as you can see in the following screenshot:

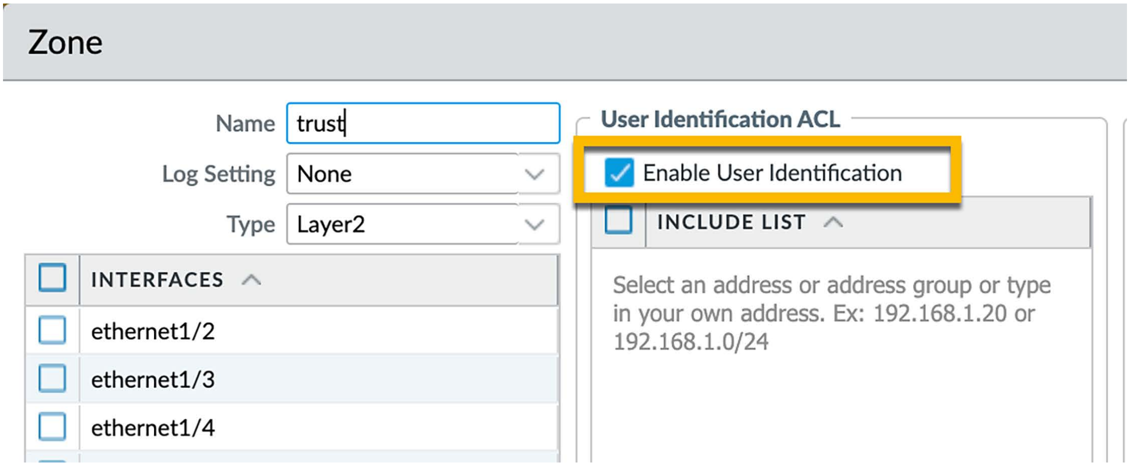


Figure 6.1: User-ID in a zone

This setting needs to be active in local zones, or remote zones (such as VPNs) that receive user sessions, but should not be enabled for untrusted zones such as internet uplinks. For each source IP in a User-ID-enabled zone, the firewall will try to create a record for user-to-IP mapping. If User-ID is enabled in an outside-facing zone, the available table space may get flooded with empty (random internet background noise) user-to-IP mappings and push out legitimate User-ID entries.

In the **Include List**, you can limit subnets to which User-ID is applied or exclude specific subnets by adding them to the exclude list. This can help limit the number of empty (without a valid User-ID to an IP) entries.

We first need to prepare AD before we can start the firewall configuration. One of the first steps we need to take is to enable audit logging in the AD local security policy as, by default, the logging we want to see is disabled. The User-ID agent (or the agentless deployment) needs to be able to capture at least one of four possible event IDs from AD: `4768` (Authentication Ticket Granted), `4769` (Service Ticket Granted), `4770` (Ticket Granted Renewed), and `4624` (Logon Success).

To do this, you will need to navigate to **Start** | **Windows Administrative Tools** | **Local Security Policy**. Then, in **Security Settings** | **Local Policy** | **Audit Policy**, set **Audit Logon Events** to **Success**, which will start logging all successful logon events that the User-ID agent can use to map the user to their workstation's IP.

You will also need to create a service account, which can be used to do the following:

- Run the service if an agent is being used.
- Connect remotely if an agentless deployment is being used.
- Perform **Windows Management Instrumentation** (**WMI**) probing. As WMI is somewhat outdated, it may no longer be relevant to your deployment for probing purposes. We will cover the topic briefly in the next section as probing is not a requirement. The service account will still need the appropriate privileges to communicate with AD.

If you are using an agent, do the following:

1. Create a new user in **Active Directory Users and Computers** | **Managed Service Accounts**. In the **Member Of** tab, add **Event Log Reader**. In the **Dial In** tab, set **Deny access**.
2. Then, in **Local Security Policy** | **Security Settings** | **Local Policy** | **User Rights Assignment**, add the service account to **Log on as a service.**
3. For security, you'll also want to add the service to **Deny log on as a batch job**, **Deny log on locally**, and **Deny log on through Remote Desktop Services**.
4. To add the user via **Group Policy Objects** (**GPOs**), if you intend to install multiple agents, do so via **Group Policy Management** | **<domain>** | **Default Domain Policy** and then right-click **Edit**. Then, select **Computer Configuration** | **Policies** | **Windows Settings** | **Security Settings** | **Local Policies** | **User Rights Assignment** and add the service account to **Log on as a service**, and the three **Deny log on** policies mentioned in *step 3*.

If you're going agentless, just follow these steps but also add the role of **Server Operator** to the **Member Of** tab in the service account.

With these settings, you will be able to reactively map user logon events to the source IP that initiated the logon, but there is also a way to actively poll who is logged on to a system, which we'll look at next.

Configuring WMI probes

One alternative method of collecting user information, or ensuring that a user is still logged on to their device, is having the agent send out periodical probes in the form of NetBIOS queries or WMI probes. NetBIOS does not require authentication but is most likely disabled in most modern networks as it is outdated and insecure. WMI uses authentication and is more secure (you may still need to allow it in the client firewall by adding **WMI** to **Windows Firewall Exceptions**). WMI probing may not yield the desired outcome if not all the devices that will be probed are Windows-based; consider disabling probing entirely if many of your users are using macOS or Linux machines.

Let's look at what you need to configure:

1. To enable WMI probing, add the **Distributed COM Users** role to the **Member Of** tab in the User-ID service account.
2. Next, you will need to set permissions for the service account to remotely probe systems: launch `smimgmt.msc`, right-click **WMI Control (local)**, and open **Properties**.

3. In the **Security** tab, select **CIMV2**, click the **Security** button, add the User-ID service account, and check the **Allow** boxes next to **Enable Account** and **Remote Enable**.

If User-ID is not set up properly, probing could generate a large amount of network traffic, so be sure to enable probing only when everything else is set up and operational.

Setting up a User-ID agent

The next step is to download the agent from `https://support.paloaltonetworks.com` > **Updates** | **Software Updates** and install it on AD. Make sure to download the `UaInstall*.msi` file (`UaCredInstall.msi` is used for user credential detection, which we will cover in the final section, *User credential detection*).

If your AD is not a suitable host to run the agent, you can run it from a different server in the same domain and read the logs remotely. This will require the service account to be added to the **Server Operator** role. Reading event logs remotely will generate some load on the network, so make sure the server is close to your AD.

You will need to run the installer as an administrator. If your Windows installer won't let you use the **Run as** option directly by right-clicking the file, a handy trick is to execute `command.exe` as an administrator and execute the installer from the command line.

Once the agent is installed, you will first need to make two more adjustments:

1. Right-click and open the properties of `C:\Program Files (x86)\Palo Alto Networks`, select **Security**, click **Edit**, and then add the User-ID service account and grant it full access to the directory.
2. Open `regedit` and add the service account with full control permissions to the Palo Alto Networks key:

   `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node` (for 64-bit systems)

   `HKEY_LOCAL_MACHINE\SOFTWARE\Palo Alto Networks` (for 32-bit systems)

3. From the **Start** menu (or from the `install` folder), run `UaController.exe` as an administrator.

Configuring the User-ID agent

In **User Identification** | **Setup**, you can configure the agent. The access control list at the bottom lets you control which systems have access to the agent. You can restrict access to your management network or individual firewall IP addresses.

The configuration section at the top lets you set all the parameters in individual tabs by clicking the **Edit** button, as shown in the following screenshot:

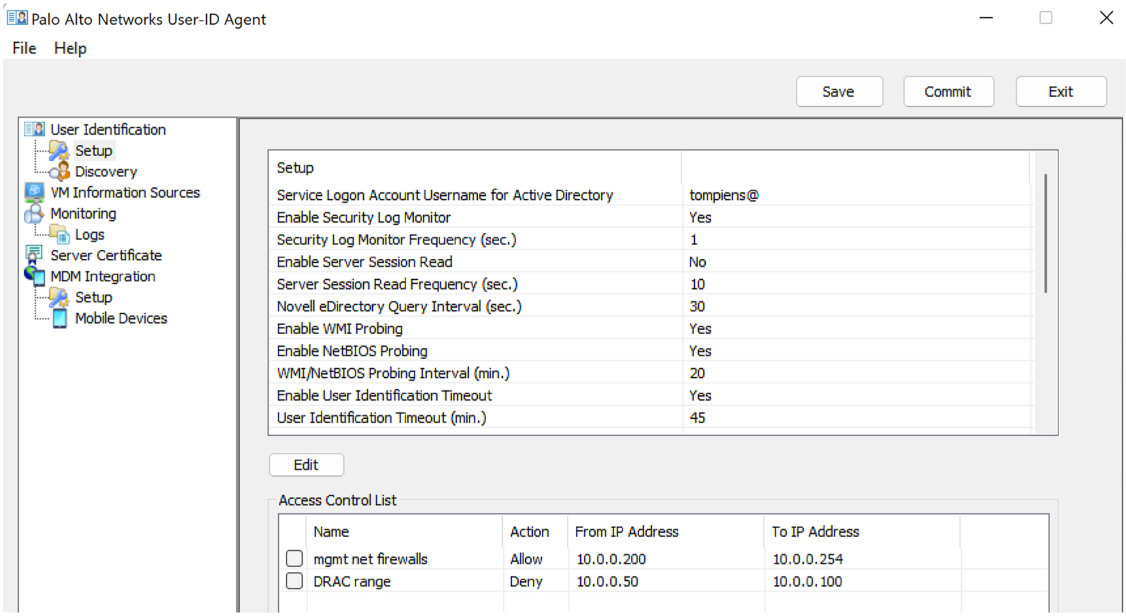


Figure 6.2: User-ID agent configuration

The following are the parameters you can set:

1. **Authentication** is where you need to fill in the service account used by the service and its password.

2. In the **Server Monitor** tab, **Enable Security Log Monitor** is enabled by default and set to `1` second. This is the process that reads the AD event logs. In the following case, it connects each second and reads the logs that have been created since the last read. You can optionally select the **Enable Server Session Read** option, which is a process that keeps track of users who have mapped a drive or directory on the local system:

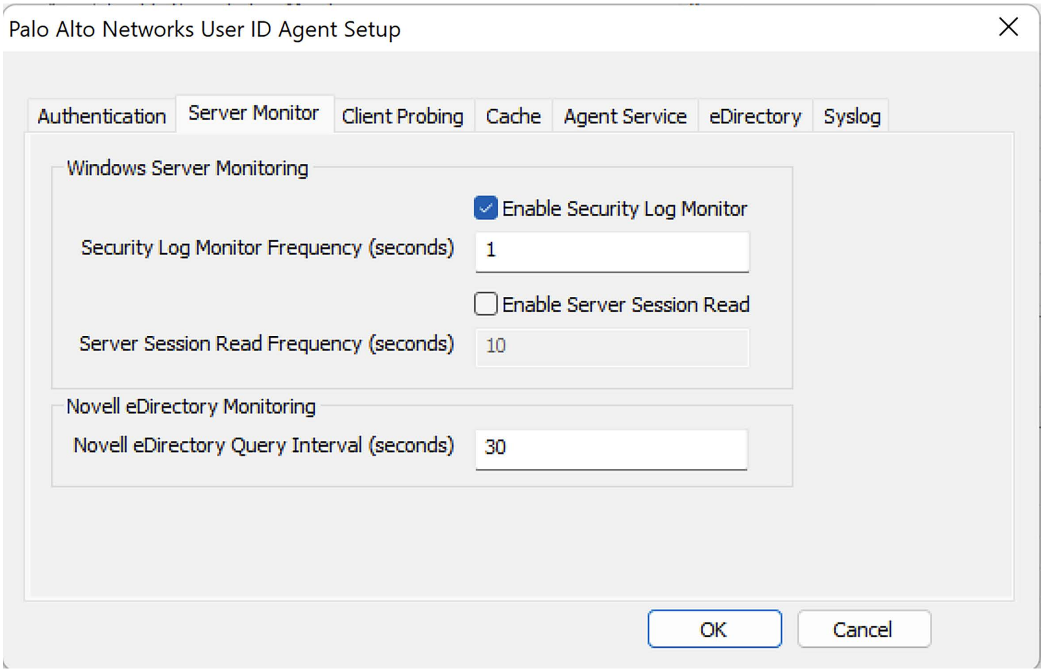


Figure 6.3: The Server Monitor tab

3. In the **Client Probing** tab, you can select whether you want to use WMI and/or NetBIOS probing, and the frequency of the probes. Mind the caveats mentioned in the *Configuring WMI probes* section.
4. In the **Cache** tab, you can control how long user credentials are cached. By default, this is enabled and set to `45` minutes. This is a hard timer, which means the user mapping is removed after the amount of time indicated and needs to be refreshed by a new logon or authentication event.

> In a fairly static office environment, my recommendation is to set this timeout to `9` or `10` hours, which is the length of a normal workday (and the default length of a Kerberos ticket is `600` minutes), as users tend to come in, log in, and then sit at their desks most of the day, possibly not generating any more logon or authentication events. Adjust the timeout to how dynamic you anticipate your environment will be.

5. In the **Agent Service** tab, you can set the port that will be used by firewalls to connect; the default is `5007`. You can also enable **User-ID XML API** (default port `5006`) if you want to use the API to inject user mappings directly into the agent.
6. In the **eDirectory** tab, you can poll a Novell eDirectory server for user information.

7. In the **Syslog** tab, you can decide to receive syslogs from an external system, such as a Cisco ISE. You'll need to define filters using regexes to scrape the logs for relevant information. These filters will vary depending on your syslog forwarder:

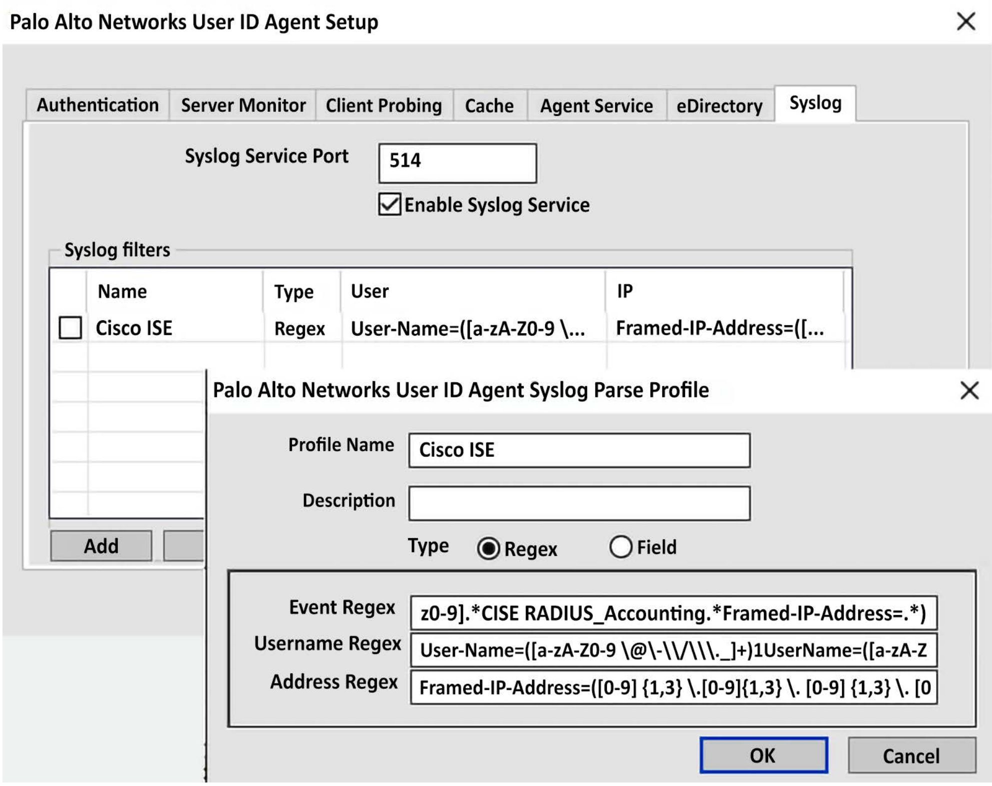


Figure 6.4: User-ID Agent syslog service

Here's an example for Cisco ISE 2.2; your instance may be different, so some tuning may be required:

- Event regex:

```
([A-Za-z0-9].*CISE_Guest.*NADAddress=.*)|([A-Za-z0-9].*CISE_
Guest.*GuestUserName=.*)
```

- Username regex:

```
User-Name=([a-zA-Z0-9\@\-\\/\\\._]+)|UserName=([a-
zA-Z0-9\@\-\\/\\\._]+)
```

- Address regex:

```
NADAddresss=([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3})
```

Here's an example for Cisco ISE 1.3:

- Event regex:

```
([A-Za-z0-9].*CISE_Passed_Authentications.*Framed-IP-
Address=.*)|([A-Za-z0-9].*CISE_RADIUS_Accounting.*Framed-IP-
Address=.*)
```

- Username regex:

```
User-Name=([a-zA-Z0-9\@\-\\/\\\._]+)|UserName=([a-
zA-Z0-9\@\-\\/\\\._]+)
```

- Address regex:

```
Framed-IP-Address=([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3})
```

8. Once you have completed the configuration, click **OK** to save the User-ID agent setup.

In the **User Identification** | **Discovery** menu, you can add the AD servers you want to poll. If the service account has been set up properly, AutoDiscover will discover and populate all of the AD servers associated with your forest (using the `_autodiscover._tcp SRV` record in your domain DNS). To remove servers, check the box and click **Delete**.

The include and exclude lists let you select which IP ranges are expected to contain known users and let you manually add exceptions. Typical exceptions include terminal servers, where multiple users are logged on at the same time (see the upcoming *Setting up a Terminal Server agent* section).

Important note

If you add an exclusion, you must also add included subnets.

Add your user subnets and add any excluded servers, and then click **Save** and **Commit**. Return to the User-ID main page. If, at the top, it is indicated that the service has stopped, click **Start**.

From this view, you will see which firewalls have made a successful connection to the User-ID agent and which AD servers are being connected to.

Once user events start being collected, new mappings will start appearing in **Monitoring**.

Now that you have configured the User-ID agent and it is collecting user information, the next step is to connect the firewall to the agent so it can benefit from the collected information and match users to security rules.

Adding the User-ID agent to the firewall

In **Device** | **Data Redistribution** | **Agents**, you can add a new entry for every User-ID agent you need to connect to.

There are a few important settings:

- The **Serial Number** radio button can be used if you have a Panorama management server that is set up for User-ID redistribution. Panorama can be set up to collect information from individual User-ID agents and then function as a distribution point. Firewalls will connect to Panorama for user-to-IP mappings instead of User-ID agents.
- **Host and Port** lets you set an IP and port for an agent so the firewall connects directly to User-ID agents to collect user-to-IP mappings. The default port for User-ID agents is `5007`.

  The agent can be set up to function as an **LDAP proxy**, in case the firewall needs to perform LDAP authentication (for VPN users or administrators) but doesn't have direct access to an LDAP server.

- **User-ID collector** information is used if the agent is another firewall configured in redistribution mode. Not only User-to-IP mappings can be shared, but also HIP reports, dynamic tags, and quarantined devices can be received from the other firewalls.
- In **Data Type**, you can select which data to collect. Some agents may serve useful tags, or you may be interested in quarantined devices.

A normal User-ID agent configuration will look like what you can see in the following screenshot. Add LDAP proxy functionality if needed, and add the User-ID collector name and the pre-shared key details if the agent is another firewall:

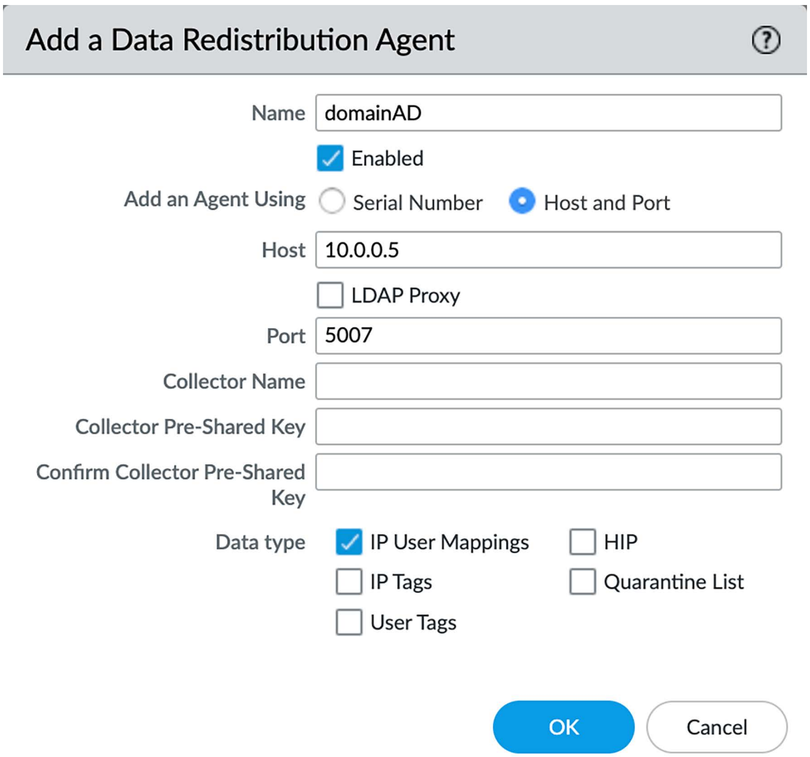


Figure 6.5: Adding a User-ID agent to the firewall

Here are a few important things worth noting about the User-ID agent:

- When the **User-ID agent** is started, it will go and read the last 50,000 log entries in the event log to build a user-to-IP mapping database.
- When the **User-ID agent** is stopped, it will retain its database for 10 minutes, after which the database is purged.
- If you need to exclude specific users, such as service accounts, you can create a file in the User-ID agent `install` directory containing all the usernames, one per line. The file must be named `ignore_user_list.txt`. You can use wildcards as a prefix in this file (for example, `*-adm`) but not as a suffix.
- You can use a certificate for authentication: create a certificate on your corporate **Certificate Authority** (CA), then import it into **Server Certificate** in the User-ID agent and create a certificate profile, and then add it to **Device** | **User Identification** | **Connection Security** on the firewall.

To enable a firewall to redistribute User-ID information, set a collector name and pre-shared secret in **Device** | **Data Redistribution** | **Collector settings** on the firewall that needs to redistribute its User-IP mapping. The firewall can now be added to other firewalls as a User-ID agent. IP connectivity needs to be available, and the collector name and pre-shared secret need to be set on all the clients.

You are now able to set up a User-ID agent that is able to match a unique source IP to a username. Next, we will learn how we can set up a Terminal Server agent for multiuser systems that host multiple unique users on the same source IP.

Setting up a Terminal Server agent

The **Terminal Server** (**TS**) agent is used to identify users who are all logged on to the same system. This means they will all have the same source IP, so to differentiate them, their source ports are adjusted to an assigned block of ports, so the firewall can identify which user is initiating a session just by looking at the source port of a session.

> **Important note**
>
> Some endpoint protection software will proxy sessions locally and randomize the source port, which interferes with the TS agent. You may need to configure the software to not touch the source port, or disable the proxy functionality altogether, for User-ID to work.

Install `TaInstall*.msi` as an administrator. Some environments may not let you open the executable as an administrator directly; as a workaround, you can launch a command prompt by right-clicking it and choosing **Run as administrator**, and then executing the installer from the command line.

Run `TaController.exe` as an administrator once the installation is complete and access the configuration.

Configuring the TS agent

On the TS agent, you will see whether any devices are connected, and you can configure an access control list to limit which devices are allowed to connect.

As shown in the following screenshot, in the **Configure** menu, you will see **System Source Port Allocation Range** and **System Reserved Source Ports**, which show the ranges of ports that are used for non-user sessions. These ranges are called ephemeral ports and are controlled by the host operating system (Windows).

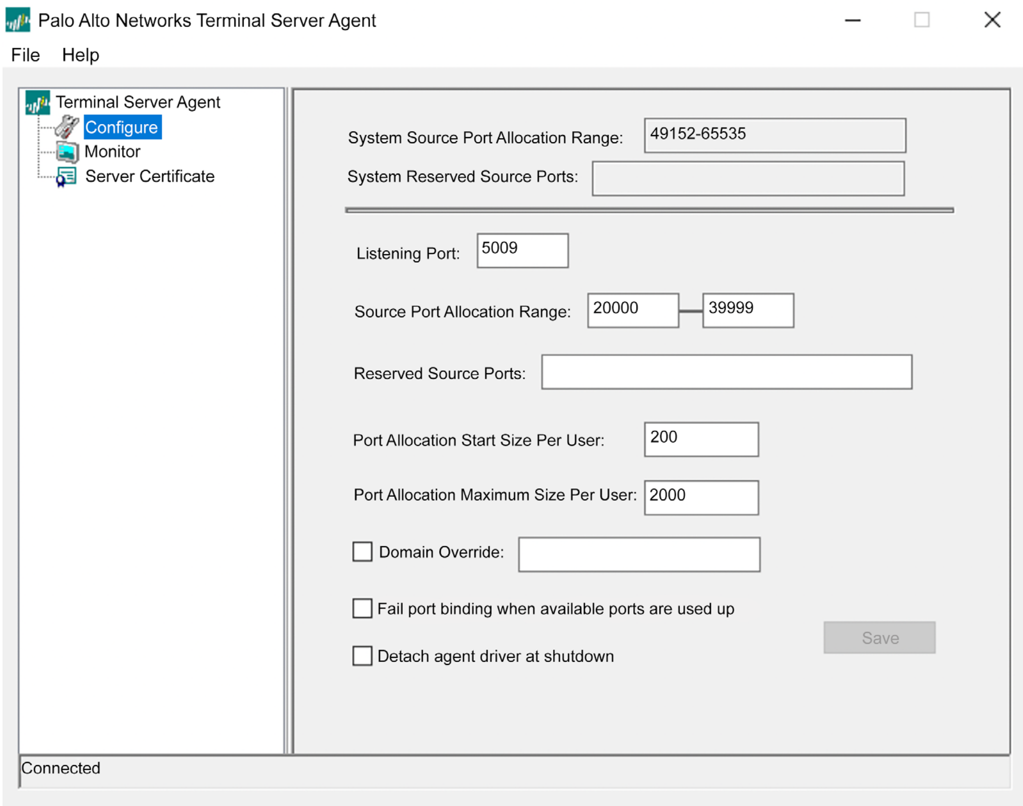


Figure 6.6: TS agent configuration

You can change this port range if you need to; the instructions are here: `https://support.microsoft.com/en-us/help/929851/the-default-dynamic-port-range-for-tcp-ip-has-changed-in-windows-vista`.

In the preceding screenshot, the following settings can be configured:

- **Listening Port** displays which port the firewall can use to receive source port information and associated usernames.
- The **Source Port Allocation Range** values determine the block of source ports that can be used by user sessions. This range can be increased as needed, as long as it doesn't overlap with the ephemeral ports.

- **Reserved Source Ports** lets you add an additional range of reserved source ports that the system can use exclusively.
- **Port Allocation Start Size Per User** is the range of ports a user can use for outgoing sessions.
- Once a user requires more source ports, a new block will be made available until the **Port Allocation Maximum Size Per User** value is reached or the total pool of available source ports is depleted.
- **Fail port binding when available ports are used up** prevents users from making any more connections once the available source ports are depleted. Disabling this option will allow users to still create sessions, but these sessions may no longer be identified.
- **Detach agent driver at shutdown** can be enabled if the TS agent becomes unresponsive when you try to shut it down.

There are a couple of cool Windows registry keys that can be found in `Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Palo Alto Networks\TS Agent\`:

- **Adv\HonorSrcPortRequest** (`0` or `1 - 0` by default) is used to allow applications to request a certain source port. This could prevent User-ID from matching the session because the source port may fall outside of the source port range used by User-ID. This setting is disabled (`0`) by default.
- **Conf\EnableTws** (`0` or `1 - 0` by default) enables polling on ports in `TimeWaitState`. This can be useful if users use applications that spawn many sessions and then leave open connections, starving new sessions of available source ports.

As you can see in the following screenshot, the **Monitor** menu keeps track of connected users:

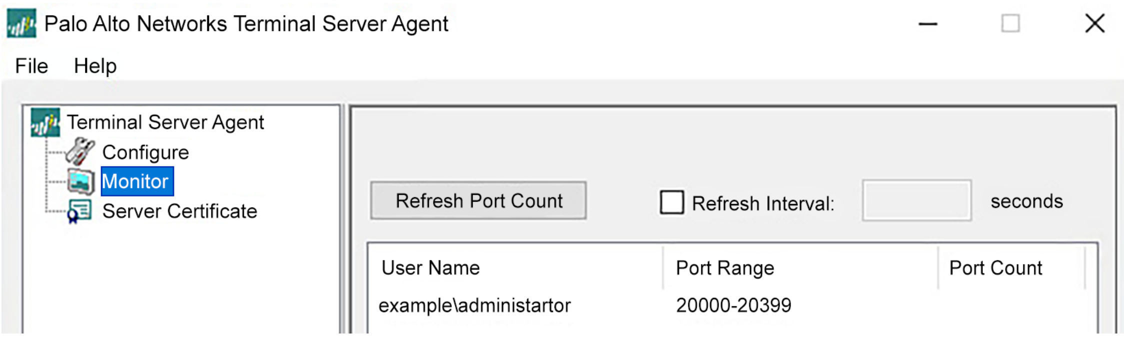


Figure 6.7: A detected user and the assigned source port range

Now that you have configured the TS agent, we can connect it to the firewall so users will start to get matched against security rules.

Adding the TS agent to the firewall

To add the TS agent via **Device** | **User Identification** | **Terminal Server Agents**, do the following:

1. Set a friendly name.
2. Set the main IP address or FQDN hostname in the **Host** field.
3. Change the port if the default port was changed on the agent.
4. Add any additional IP addresses the server may have – this is optional, of course.

The dialog box should look similar to the following screenshot:

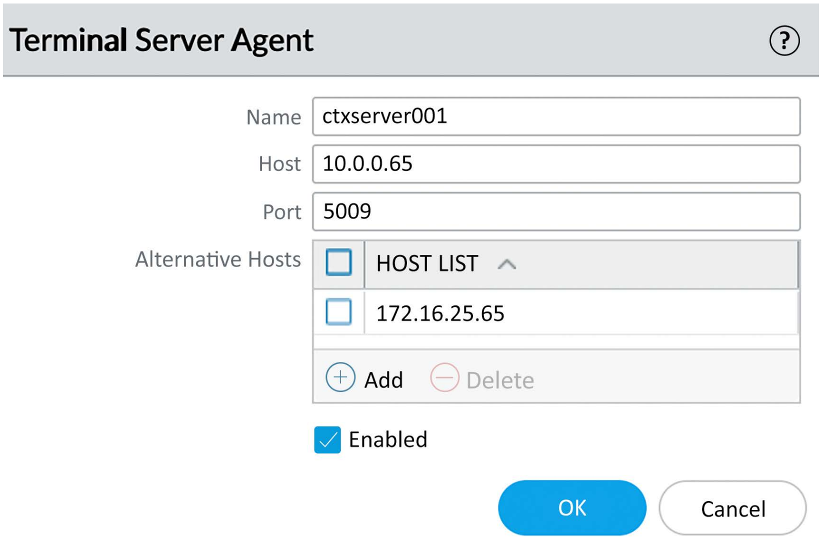


Figure 6.8: Adding a TS agent to the firewall

You are now able to configure both agents and connect them to the firewall, but the firewall can also function as an agent, which does not require the installation of a piece of software. In the next section, you'll learn how to set that up.

Agentless User-ID

The firewall also supports a clientless version, where the firewall acts as the agent. In **Device** | **User Identification** | **User Mapping**, you can define four types of servers that can be contacted to retrieve user information:

- **AD:** Reads event logs over WMI, WinRM-HTTP, or WinRM-HTTPS. WinRM is preferred as WMI is somewhat outdated
- **Exchange**: Monitors exchange connections over WMI, WinRM-HTTP, or WinRM-HTTPS
- **Novell eDirectory**: Accesses eDirectory user logins
- **Syslog Sender:** Sets the firewall as a syslog receiver and sets a filter (including Aerohive, Blue Coat, Juniper, Cisco, Citrix, and Squid predefined filters over SSL or UDP)

As illustrated in the following screenshots, the above attributes can be selected:

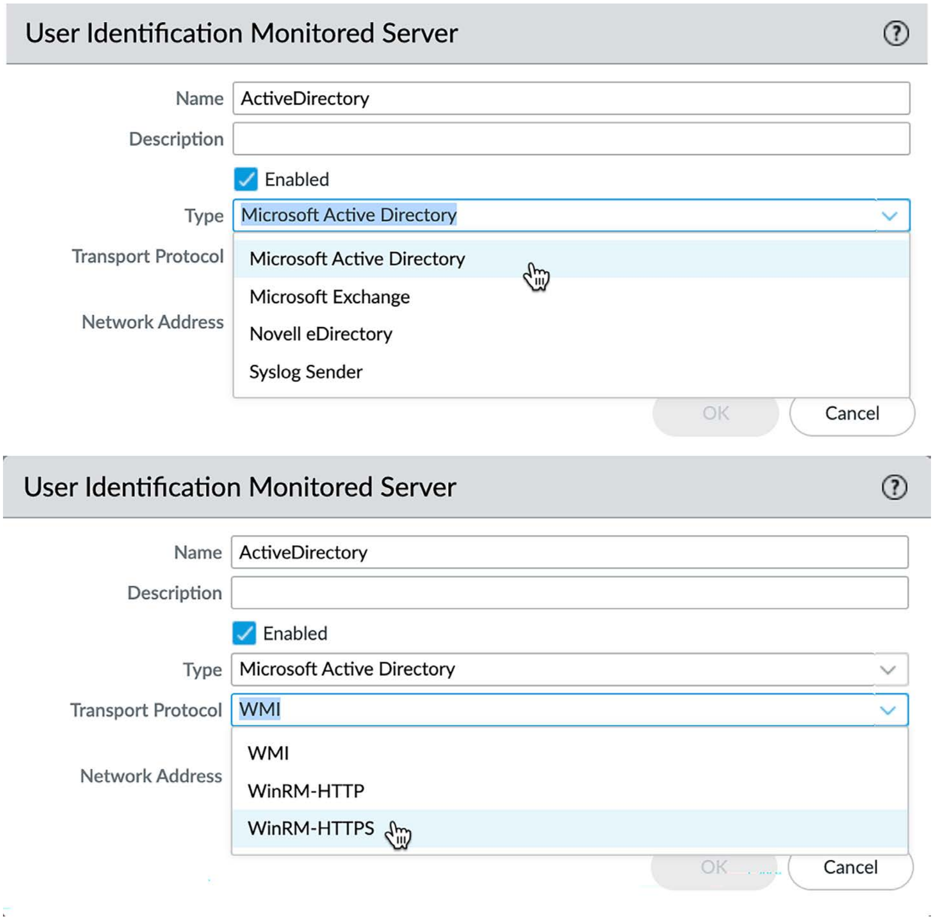


Figure 6.9: Adding server monitoring servers

You can also "autodiscover" available servers by clicking **Discover** under **Server Monitoring:** make sure the firewall is configured to use the internal DNS servers (to pick up on the `_autodiscover._tcp` SRV record) and has the domain set in **Device** | **Setup.**

Add include/exclude networks to limit the scope to your actual user subnets and exclude servers that may need a TS agent.

Configure the clientless agent with the following settings:

1. In **Server Monitor Account**, add a service account.
2. Make sure that, on the **ActiveDirectory** server, the account is set as a member of **Distributed COM Users** and **Event Log Readers**.
3. Enable the account for WMI probing.
4. Set the domain's full DNS name (`example.com`).

5. If you want to use **Windows Remote Management** (**WinRM**) to connect to servers, you need to add a Kerberos server profile (make sure that the firewall is set up with internal DNS servers, has the domain in **Device** | **Setup** | **Management** | **General Settings**, and has the NTP servers set).
6. To use WinRM-HTTPS, also add a User-ID certificate profile in **Device** | **User Identification** | **Connection Security**.

   Your config will look similar to the following screenshot:

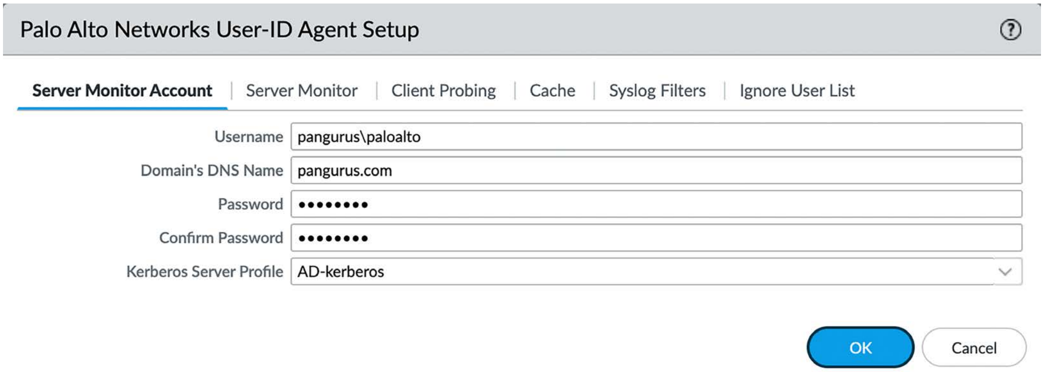


Figure 6.10: The Server Monitor Account tab

7. As shown in the following screenshot, in the **Server Monitor** tab (accessed through **Device** | **User Identification** | **Connection Security**), log reading is enabled by default, and a server monitor can be enabled, giving you control over the poll frequency in seconds. If the agent should listen for syslogs, an SSL/TLS profile can be added here if the connection is set to use SSL instead of UDP:

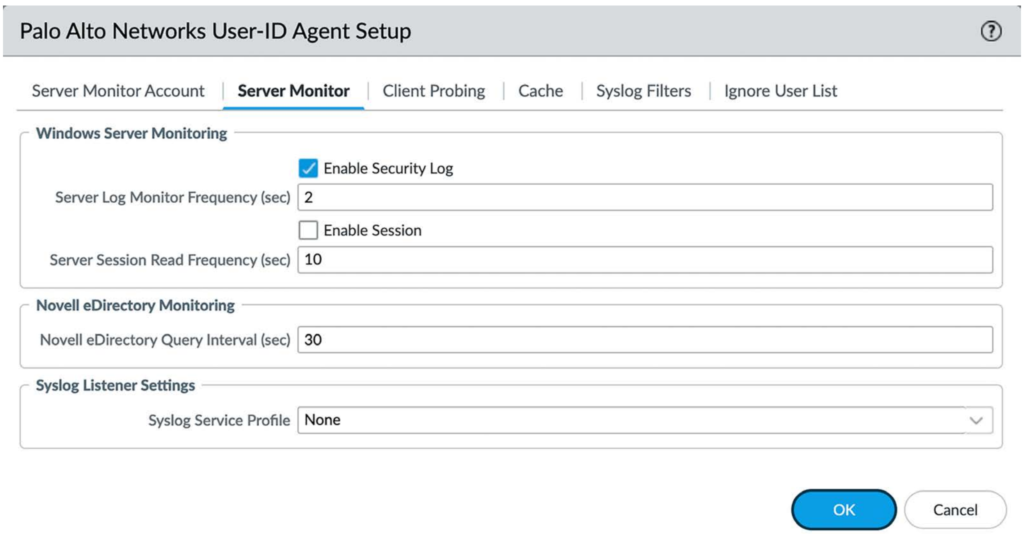


Figure 6.11: Server Monitor tab

8. In **Client Probing**, WMI probes can be enabled and their frequency can be set in minutes. Ensure Windows Firewall has been configured to accept WMI probes; otherwise, skip this step.
9. Unlike the client installed on a server, the clientless deployment does not support NetBIOS probing. If you intend to enable probing, make sure that the include and exclude networks have been set up so probes are not sent to inappropriate or high-security networks.
10. In **Cache**, you can choose whether user-to-IP mappings will live and how long they will live. Once the timeout expires, the mapping is removed and the user will need to create a new logon event before they can be identified again. For normal office environments, a timeout of `9` to `10` hours is usually appropriate. In a highly dynamic environment, a shorter period may be preferred. (In extremely static environments, a timeout may not be needed, although I do not recommend that.)
11. If usernames are to be collected without domains (NetBIOS prefix or UPN suffix), enable **Allow matching usernames without domain**.
12. In PAN-OS 9.1 and older, NTLM can be configured to query the user's browser for credentials. Starting from PAN-OS 10.0, **NTLM** has been retired in favor of Kerberos. Plan accordingly if you're upgrading to PAN-OS 11.2:

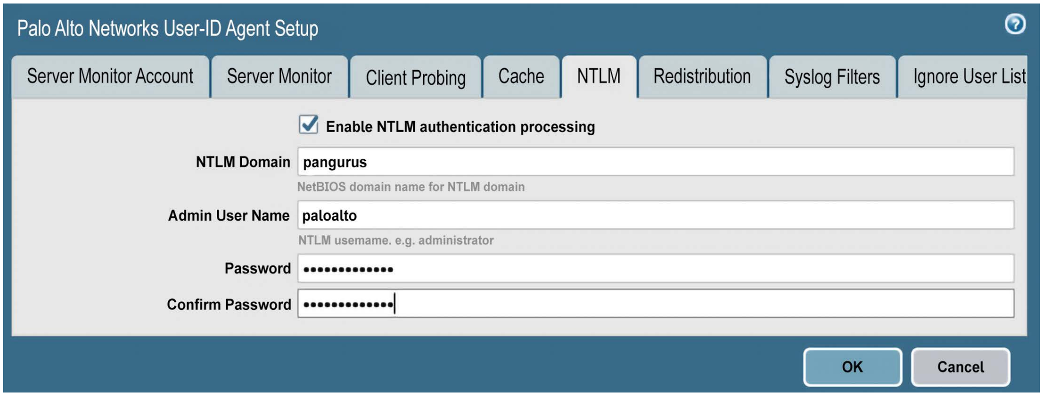


Figure 6.12: NTLM configuration

13. **Redistribution** enables the firewall as a User-ID agent for other firewalls. The firewall can only redistribute locally learned mappings (so, not mappings it has learned from other firewalls or agents).
14. You can add additional syslog filters or check out the predefined ones for inspiration. As you can see in the following screenshot, many vendors have been preloaded, so you don't need to create regexes to interpret syslogs:

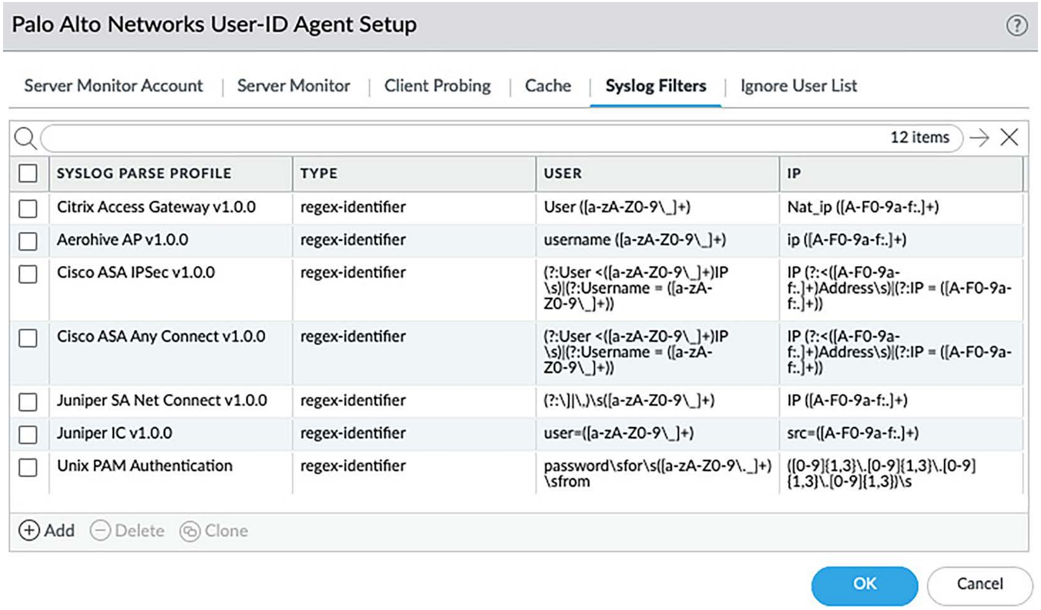


Figure 6.13: Syslog filters for popular network vendors

15. If service accounts or specific user accounts need to be ignored, add them to **Ignore User List**.

You are now able to set up both the agents and the agentless User-ID to collect information from AD or probe the client for the logged-in user. In the next section, you will learn how to configure group mapping so that users can be identified by their LDAP/AD group memberships.

Configuring group mapping

If you are able to identify users on your network, you are also able to create security rules to allow or limit their access to certain resources. **Role-based access control** (**RBAC**) can easily be enforced by binding LDAP groups to security policies, granting members of a certain organization within your company exclusive and reliable access to the resources they need wherever they go.

To get started, we need to create an LDAP profile so we can fetch group information:

1. Go to **Device** | **Server Profiles** | **LDAP** and create a new profile. You will need one LDAP profile per domain in a multidomain or forest configuration.
2. There needs to be at least one server, but there can be up to four for redundancy. Add at least one server by IP or FQDN and don't forget to set the appropriate port (`636` should be the default, `389` for legacy unencrypted systems) if you're going to use TLS encryption.
3. Set the type to `active-directory` unless you have a different deployment (`sun`, `e-directory` or `'other'`).

4. If you set the IP and port correctly, the base **Distinguished Name** (**DN**) will load automatically once you click the drop-down arrow. You can add **Organizational Units** (**OUs**) and **Common Names** (**CNs**) if needed.
5. **Bind DN** is the account that's used to read the directory structure and all members. A regular user-level account is sufficient; no special privileges are required unless you have hardened your LDAP environment.
6. Click **OK** and create additional profiles if there are more domains. If all went well, your LDAP profile should look as follows:

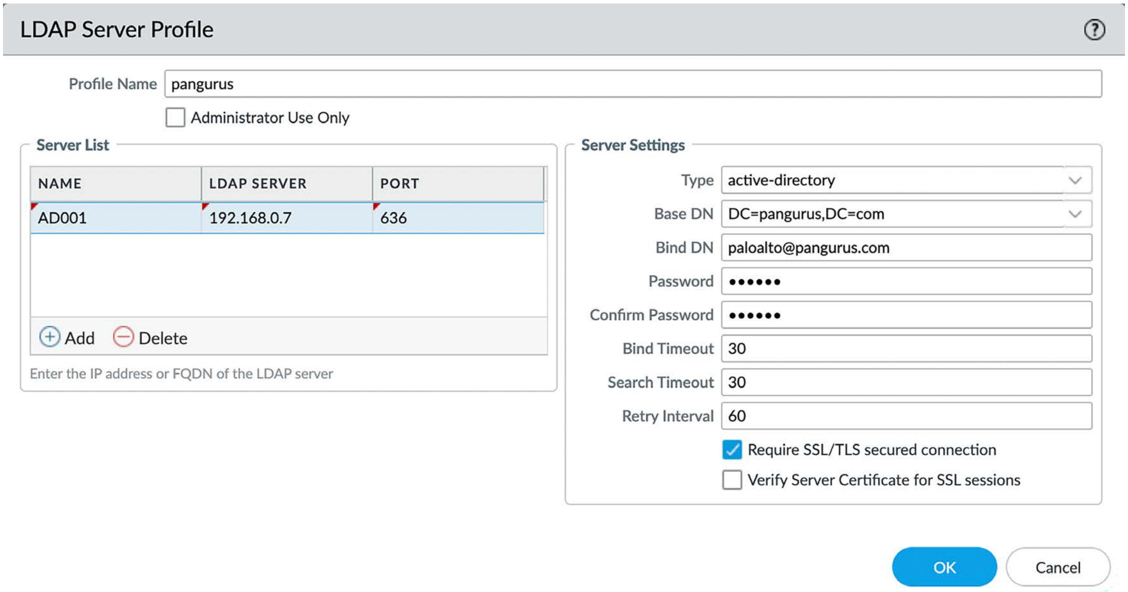


Figure 6.14: The LDAP Server Profile window

If you have **Universal Groups**, do the following:

1. Create an LDAP server profile that connects to the root domain of the global catalog server on port 3268 or 3269 for SSL.
2. Create an LDAP server profile to connect to the root domain controllers on port 389 or 636 for SSL.

This will ensure that you are able to get information from all domains and subdomains.

The next step is to read the available domain tree and select which groups to monitor and keep user information on. Go to **Device** | **User Information** | **Group Mapping Settings** and create a new group mapping object as illustrated in *Figure 6.15*:

1. Create a friendly name and set the LDAP profile you just created.

2. The update interval for the firewall to recheck user membership is 60 minutes, but it can be configured to be between `60s` and `24h`. This timing can be changed in the **Group Mapping** setting **Update Interval**, as shown in *Figure 6.15*. This interval means that when adding a new user to a group on AD, it may take up to an hour before the firewall is made aware of this change. Rather than setting the update interval really low, you can manually refresh the group memberships with one of the following commands:

```
> debug user-id refresh group-mapping group-mapping-name <profilename>
> debug user-id refresh group-mapping all
```

3. In the **User Domain** field, you can optionally add a domain (NetBIOS, not FQDN) to override all user domains retrieved from the LDAP. This could be handy if User-ID picks up specific domains but LDAP has them listed differently. For a global catalog LDAP profile, leave this field empty as it would override all user domains.
4. There are also search filters available for group and user objects (**sAMAccountName** or **user-PrincipalName** (**UPN**) are useful filters for the user object).

The **Server Profile** tab should look similar to the following screenshot:

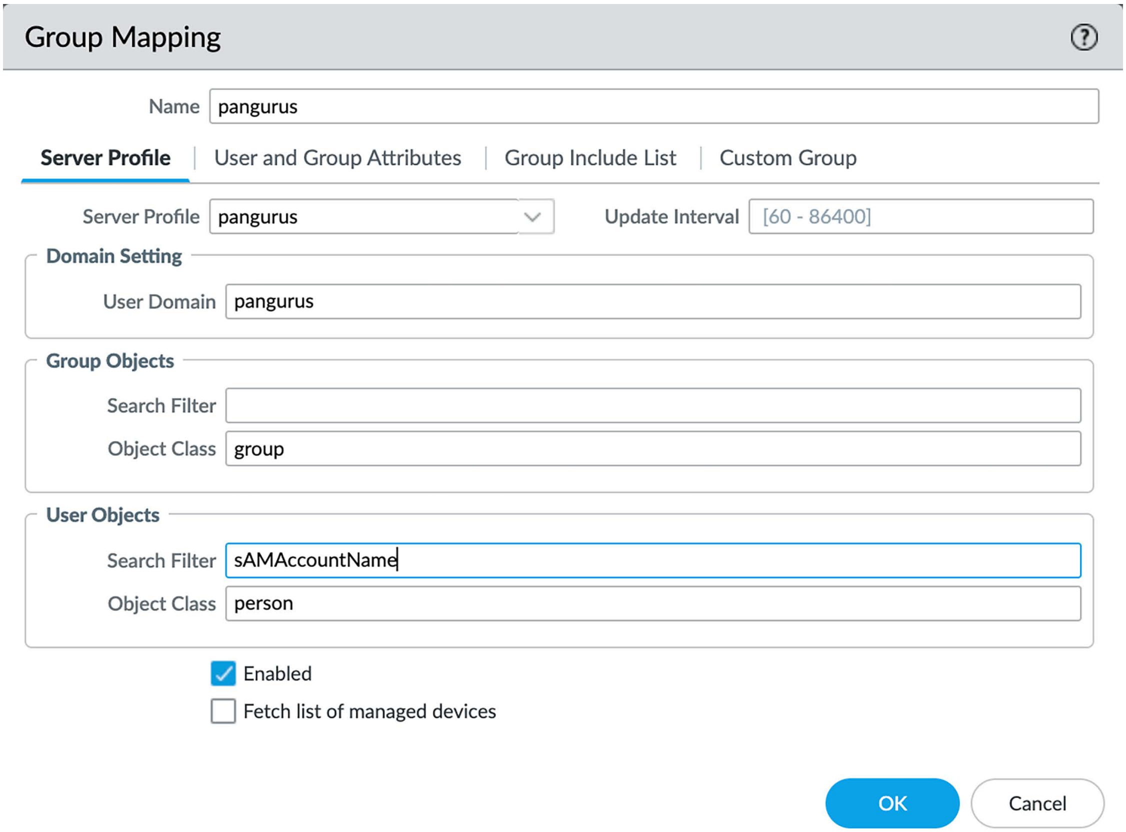


Figure 6.15: Group mapping server profile

In the **User and Group Attributes** tab, you can fine-tune which attributes are included in the returned results. By default, **sAMAccountName**, **mail**, and **User Principal Name** are all set, with **sAMAccount-Name** set as the primary username.

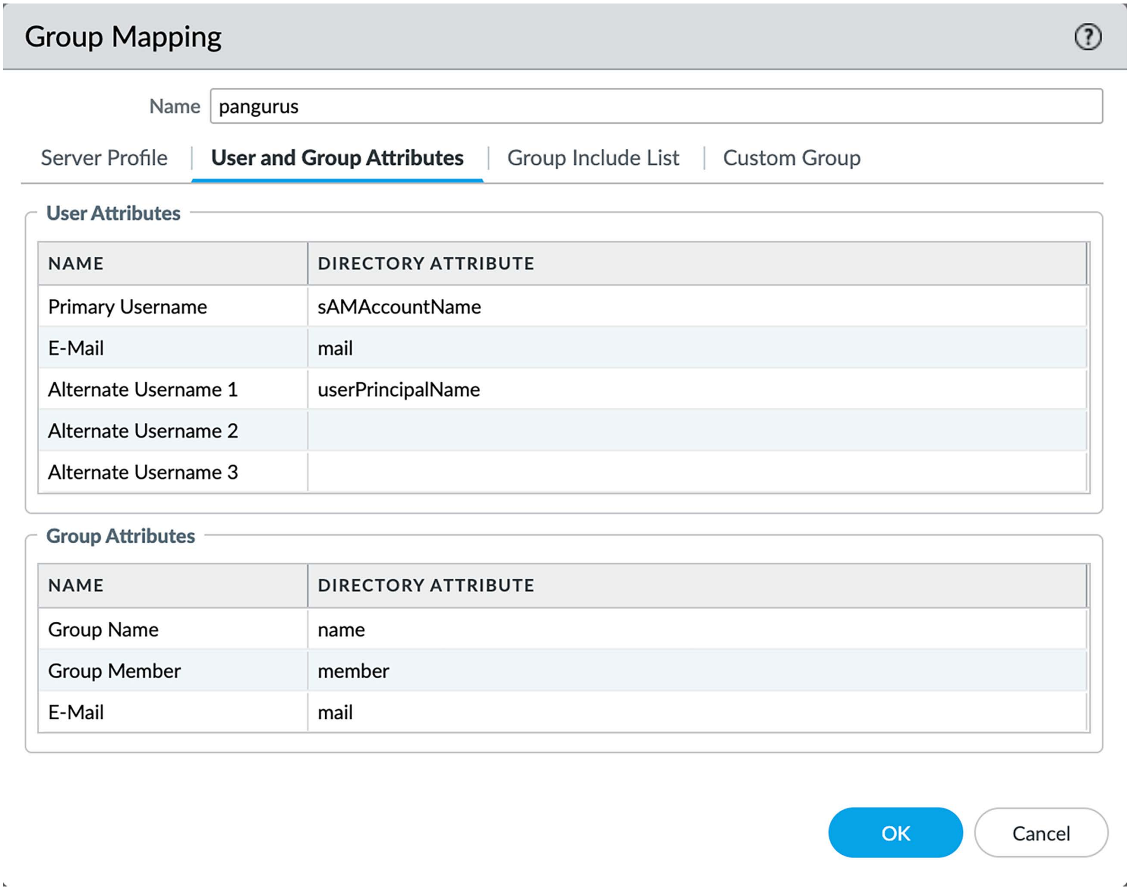


Figure 6.16: Group mapping attributes

It is useful here to review which attribute is returned by your available User-ID sources and set that as the primary username (if the User-ID agent returns UPN usernames, set **userPrincipalName** as the primary username). For Sun or e-directory type servers, the attribute will likely be **uid** or similar.

In the **Group Include List** tab, you can add the groups you want to use in security rules. You can add all the groups you want to create specific rules for by expanding the base DN on the left-hand side and adding groups of interest to the right side, as shown in the following screenshot.

There is no need to add groups that will not be used in security rules, nor the **cn=domain users** group. For rules that should apply to all users, the **known-user** user option is available in security rules to indicate any legitimately identified user:

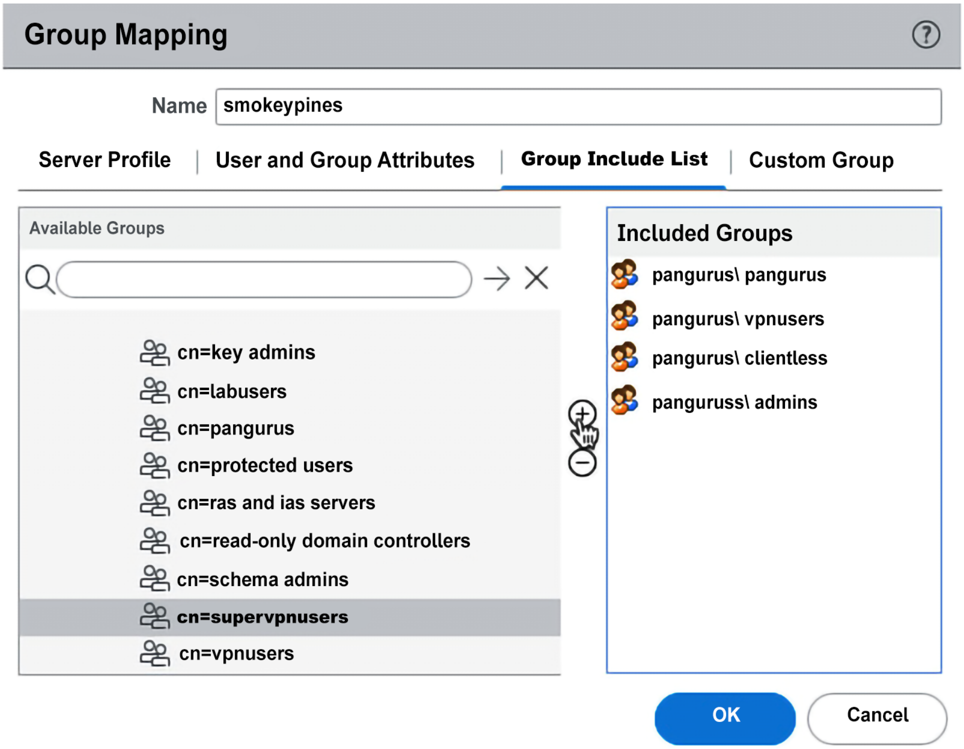


Figure 6.17: Group Include List

If custom attributes are used within your organization, the **Custom Group** tab lets you set filters to identify and record usernames in these attributes. Make sure the attributes are indexed on the LDAP to expedite searches.

A useful command to verify which attributes are captured is `show user user-attributes user all`:

```
admin@firewall> show user user-attributes user all
Primary: example\tomfromit
Alt User Names:
1) example\tomfromit
2) tomfromit@example.com
Primary: example\jimfromhr
Alt User Names:
1) example\jimfromhr
2) jimfromhr@example.com
```

You can also list which users are in each group, to ensure that the data is being retrieved correctly. Retrieve a list for all available groups via `show user group list`. You can use both the DN and NetBIOS formats for the group via `show user group name <groupname>`:

```
admin@firewall> show user group name cn=hr,cn=users,dc=example,dc=lab
short name: example\hr
source type: proxy
source: example.lab
[1] example\jimfromhr
```

Important note

The source type in the preceding code is set as `proxy` because one of the User-ID agents is configured as an LDAP proxy. Without the User-ID acting as a proxy, the source type would be as follows:

```
source type: ldap
```

As you can see in the following screenshot, you can now build security rules where the source (or destination) user group can be selected to grant or deny a group of people access to a resource. The little icon next to the user object indicates whether the object is a group or a user. **known-user** indicates that the firewall will match any user, as long as they are identified:

			Source			Destination	
	NAME	TYPE	ZONE	ADDRESS	USER	ZONE	ADDRESS
1	server access	universal	Trust-L3	any	pangurus-users	DMZ	servers
2	internet access users	universal	Trust-L3	any	known-user	Untrust-L3	any

Figure 6.18: Source users in security rules

With more infrastructure services moving into the cloud, the AD environment may also become cloud-based, which makes having regular LDAP connections less practical or impossible. To consolidate group mapping across multiple cloud-based platforms, Palo Alto provides a free service called the **Cloud Identity Engine** to serve as a convergence point for group mapping.

The Cloud Identity Engine

The **Cloud Identity Engine** (**CIE**) is a free tool provided by Palo Alto Networks to customers with an active support subscription that is capable of combining multiple sources of user information, called Directory Sync, into one entity all of your organization's firewalls can connect to, reducing configuration complexity. It also provides the Cloud Authentication Service, which serves as a single authentication point to multiple **Identity Providers** (**IdPs**) based on **Security Assertion Markup Language** (**SAML**) 2.0, also simplifying configuration if multiple IdPs are available. The CIE can be enabled from `https://apps.paloaltonetworks.com`, which requires an active support account; free (LIVEcommunity) accounts cannot use this tool.

As illustrated in the following figure, you just need to select the **Activate** button below the **Cloud Identity Engine** tile on the main page, fill in some basic information, like a friendly name and the region this tool should be active in, and you're ready to get going:

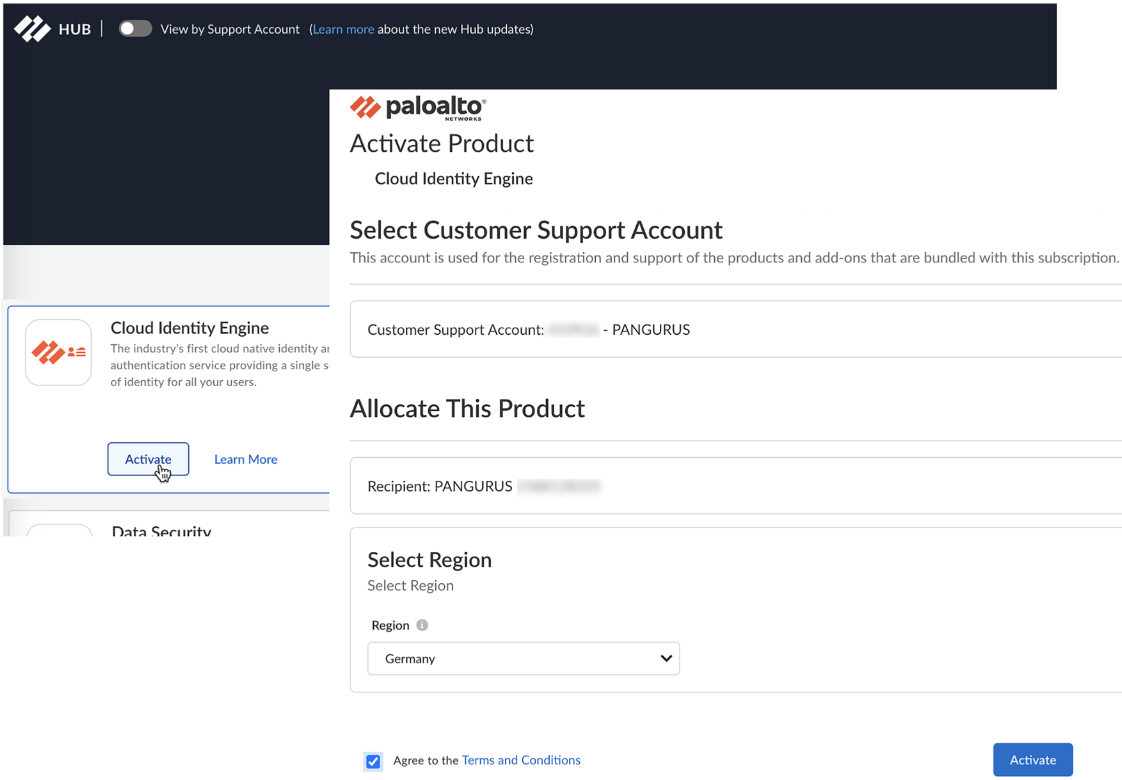


Figure 6.19: Enabling the CIE

The next step is to configure a **Cloud Directory** or an **On-Premises Directory**. The on-premises directory sync is achieved via a downloaded agent software that syncs up to the cloud. Lastly, authentication can be configured.

As you can see in the following screenshot, you can currently set up **SCIM**, **Azure**, **Okta**, or **Google** authentication; more providers will probably be added in the future:

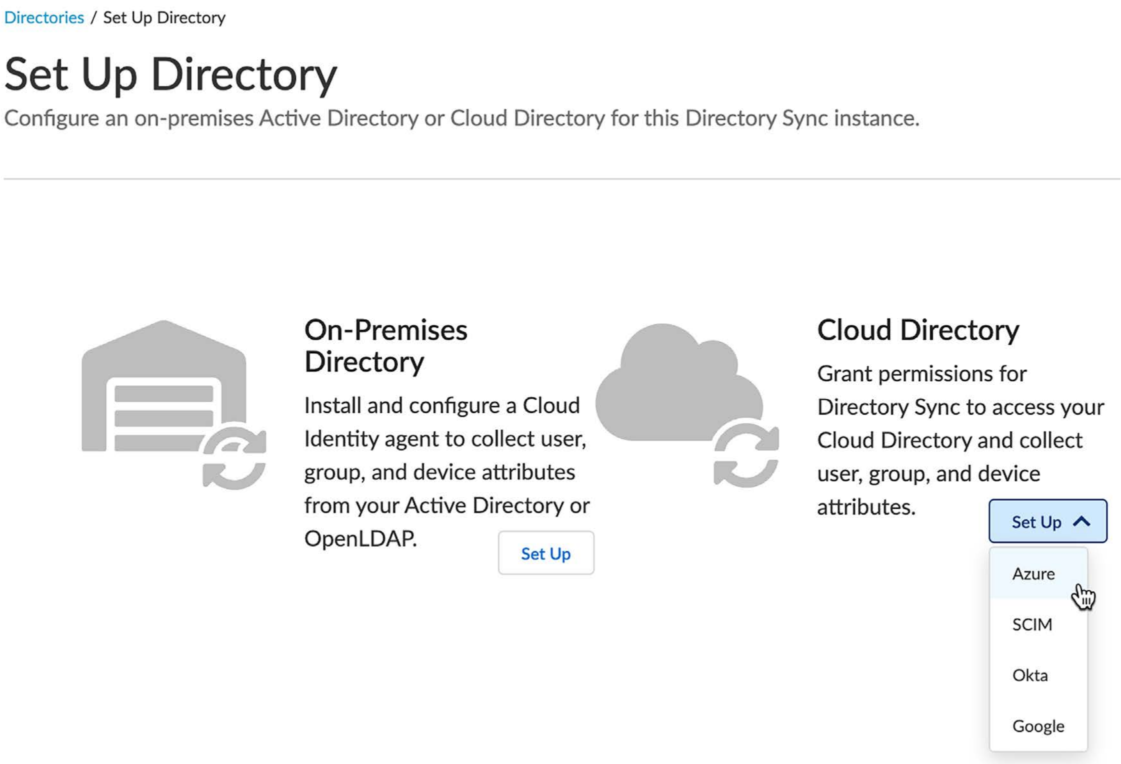


Figure 6.20: Configuring the CIE

To enable a cloud directory, you need to have an account on the appropriate cloud service, and then from the CIE simply provide the appropriate credentials, as you can see in the following screenshot:

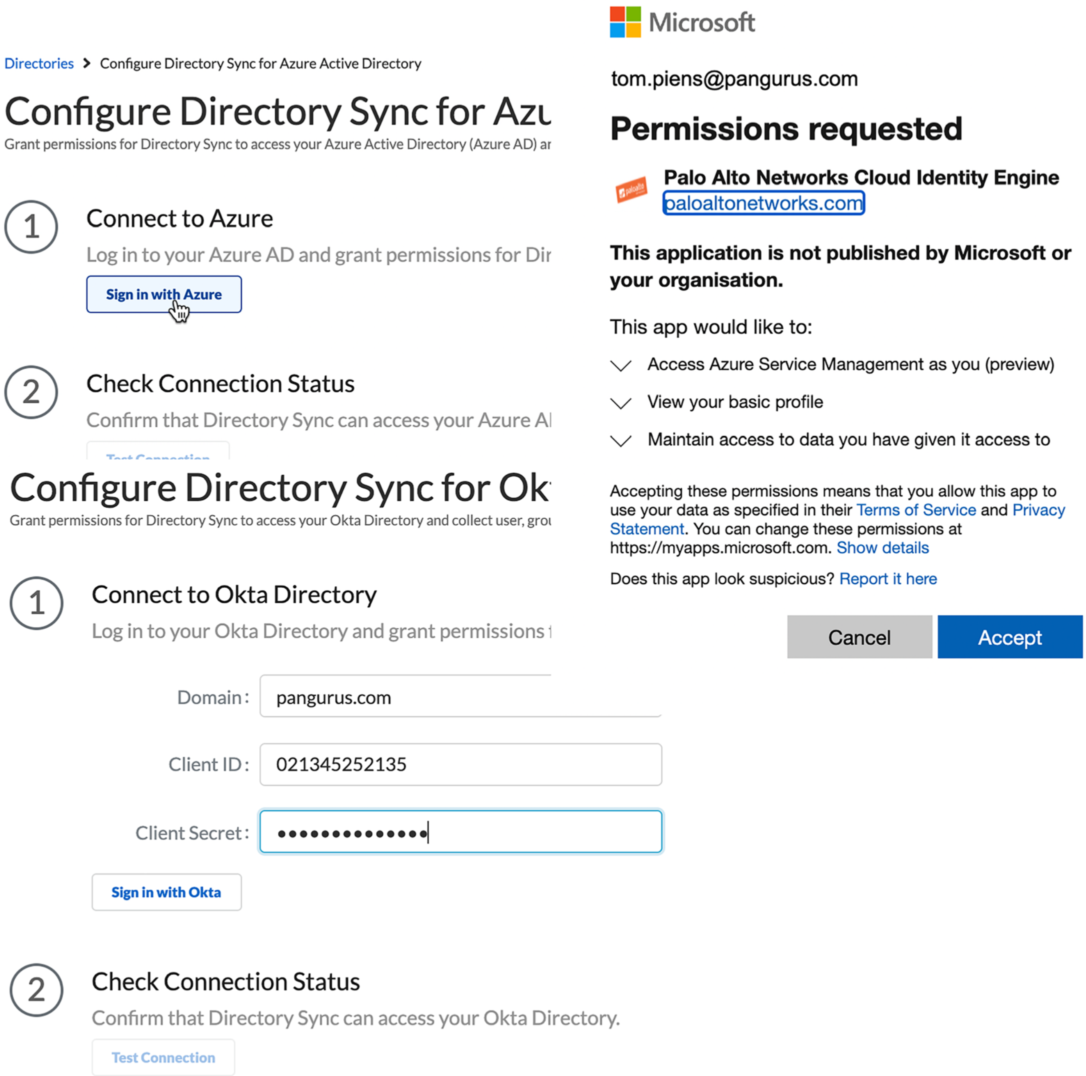


Figure 6.21: Adding a cloud directory

As you can see in the next screenshot, to add an on-premises directory service, you will simply need to download the agent and generate a certificate it will use to communicate with the cloud. You will be asked for this certificate during the installation process:

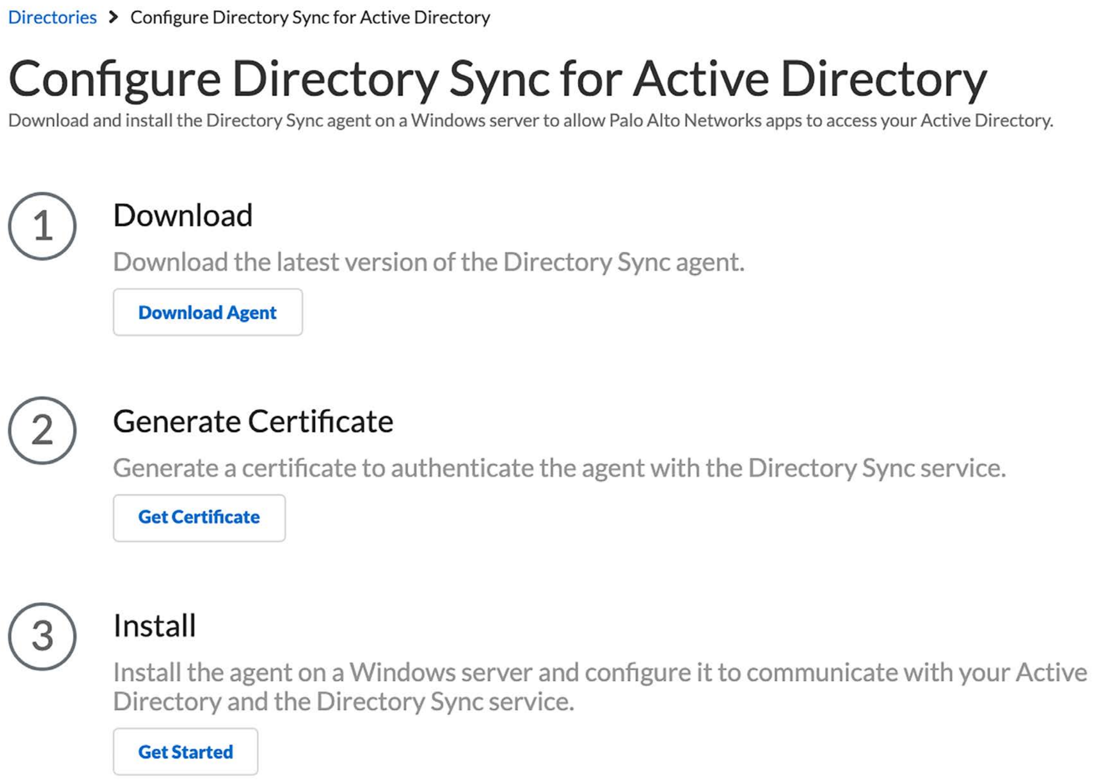


Figure 6.22: Adding an on-premises directory

The next step is to add the CIE to the firewalls that need group mapping information. First, make sure the device certificate has been set in **Device** | **Setup** | **Device Certificate**. If there's no device certificate yet, you can click the **Get Certificate** link and, in a different browser tab, navigate to `https://support.paloaltonetworks.com` and access **Assets** | **Device Certificates** to generate a **one-time password** (**OTP**):

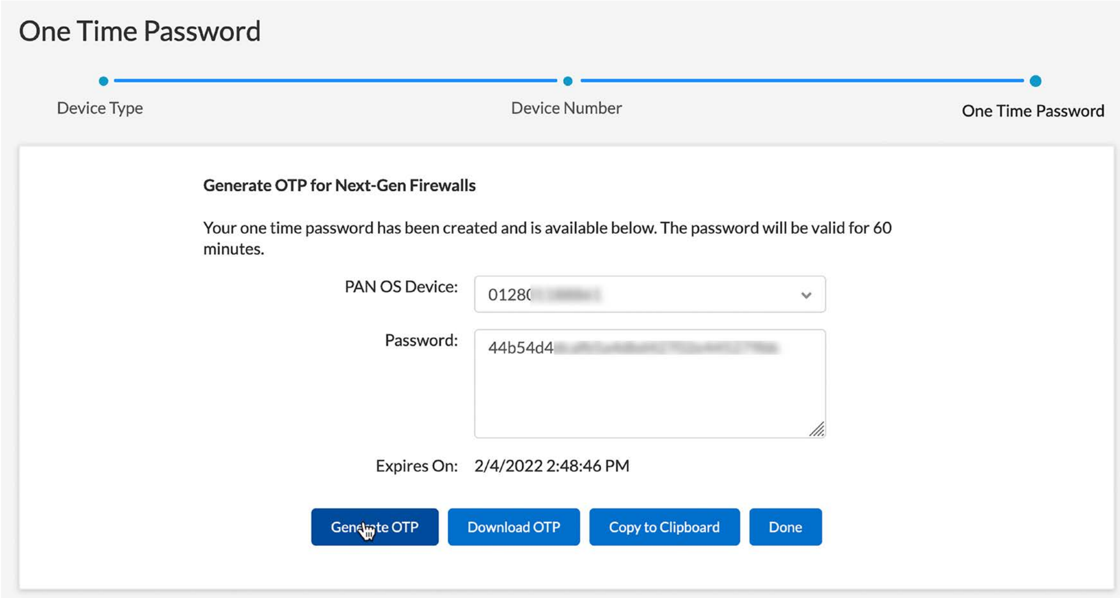


Figure 6.23: Generating a device certificate OTP

Next, navigate to **Device** | **User Identification** | **Cloud Identity Engine** and create a new profile, as illustrated in the following screenshot. In the **Instance** tab, you must select the appropriate region, which will then display the configured CIE.

In the **User Attributes** tab, you can select which attribute is used for the username. It is advisable to pick the attribute that matches the format used to authenticate users. For example, if users are logging on using their email address, select **User Principal Name** or email, if users log on with a shortened or sAMAccountName, pick **Common-Name**. The firewall will normalize usernames in groups but picking the correct attribute will facilitate this process.

In **Group Attributes**, you can select how groups are identified in the directory. You can even elect to collect device serials if these are available in your directory.

The **Device Attributes** tab can be used to collect a managed device's serial number, which can be used by the GlobalProtect portal to verify if a device is in the directory and is in fact a managed device.

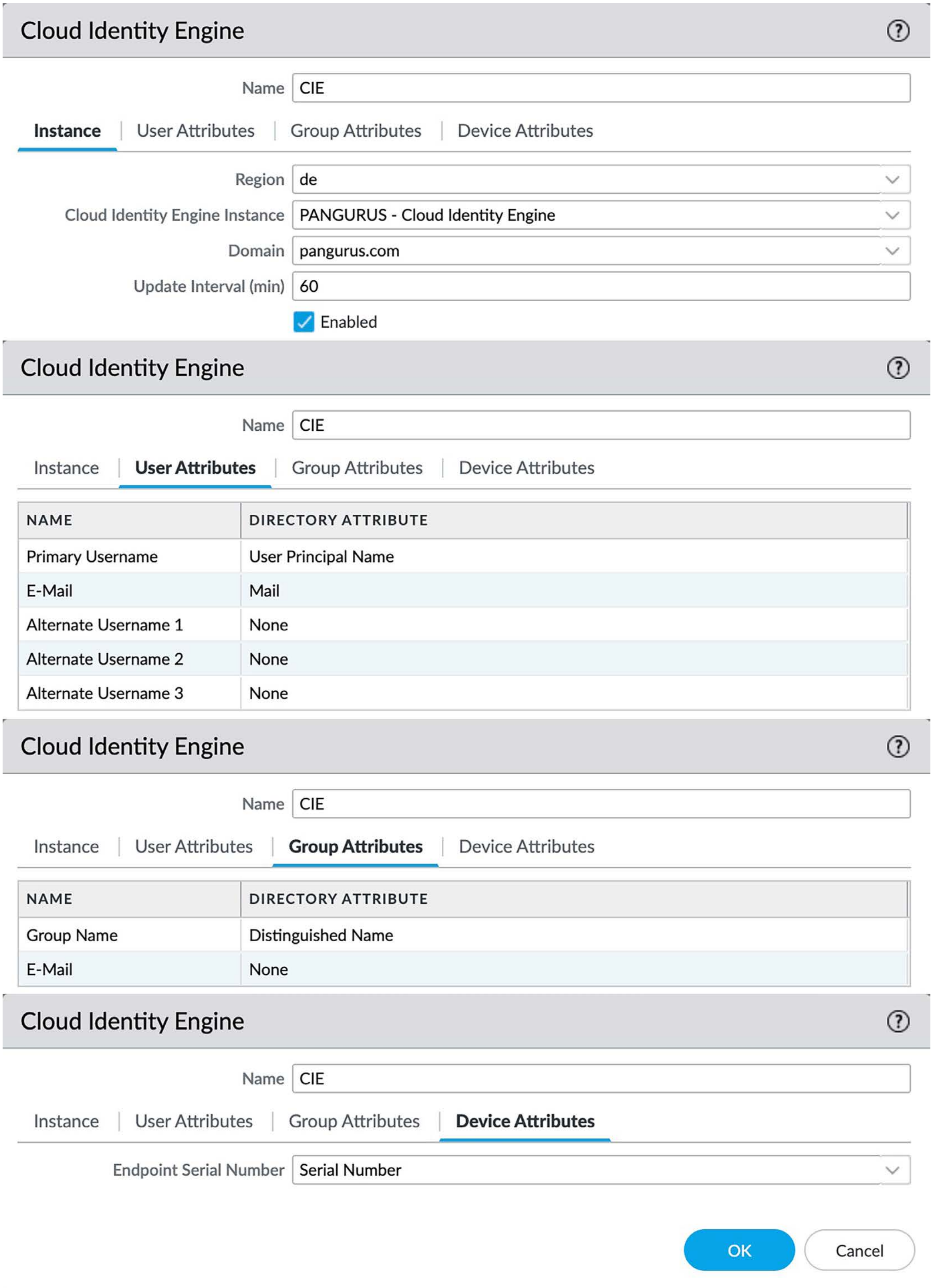


Figure 6.24: Configuring a CIE profile

After you commit this change, your firewall will now start collecting group mapping information, which you can review via the CLI command `show user group list`.

Adding cloud authentication works in the same way:

1. In **Authentication** | **Authentication Types**, click **Add New Authentication Type**.
2. From the available options (**SAML2.0**, **Client Certificate**, **OpenID Connect Protocol** (**OIDC**)) select **Set Up** under **SAML2.0**.
3. Download the service provider (SP) metadata file, which will be used in the IdP.
4. Set a profile name and select the authentication provider of your choice: **Azure**, **Okta**, **PingOne**, **PingFederate**, **Google**, or **Others**.
5. Upload the IdP metadata file if your IdP makes one available after configuring the application, or manually enter all the required information, like SSO URL, ID, and certificate. See the next topic, *Configuring Entra ID (Azure) enterprise applications*, for an example of how to obtain the metadata file.

The **Cloud Identification** page will look like the following screenshot after you have imported the Azure metadata XML:

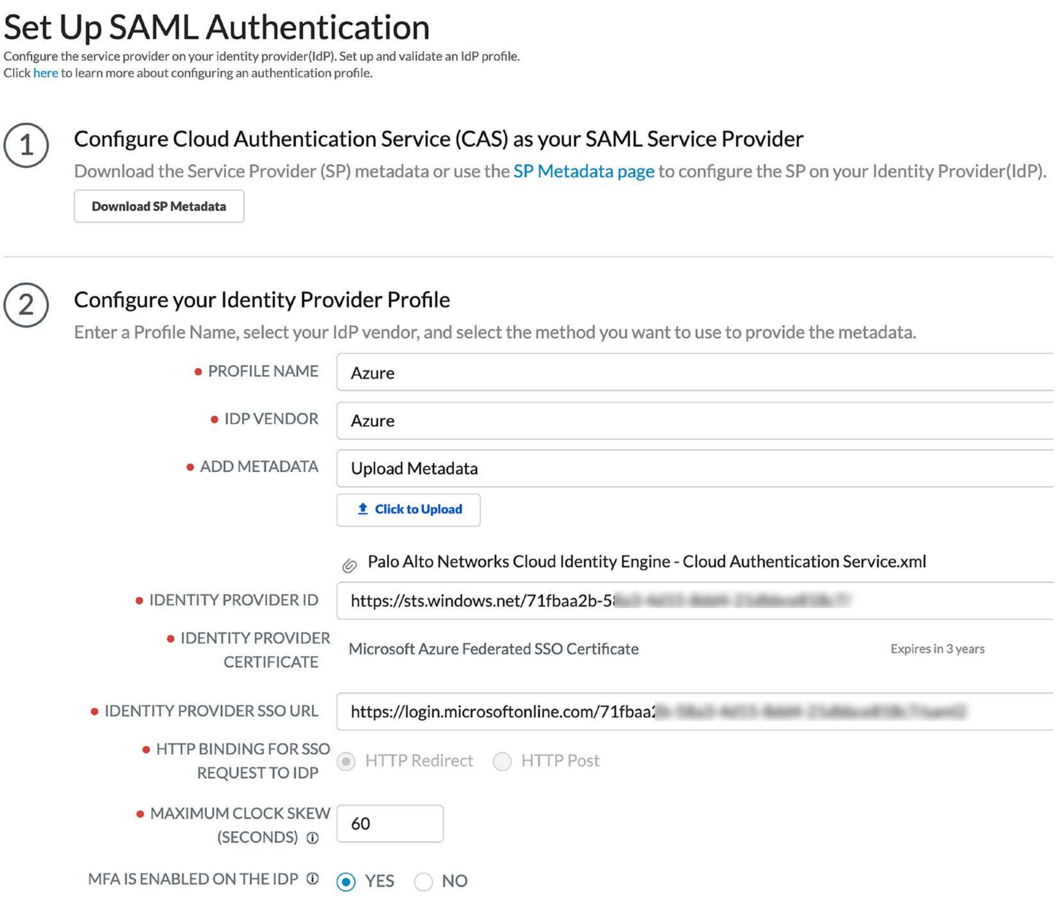


Figure 6.25: Configuring cloud authentication

Next, test the IdP connectivity, and when the test succeeds, select the appropriate user attributes, as you can see in the next screenshot:

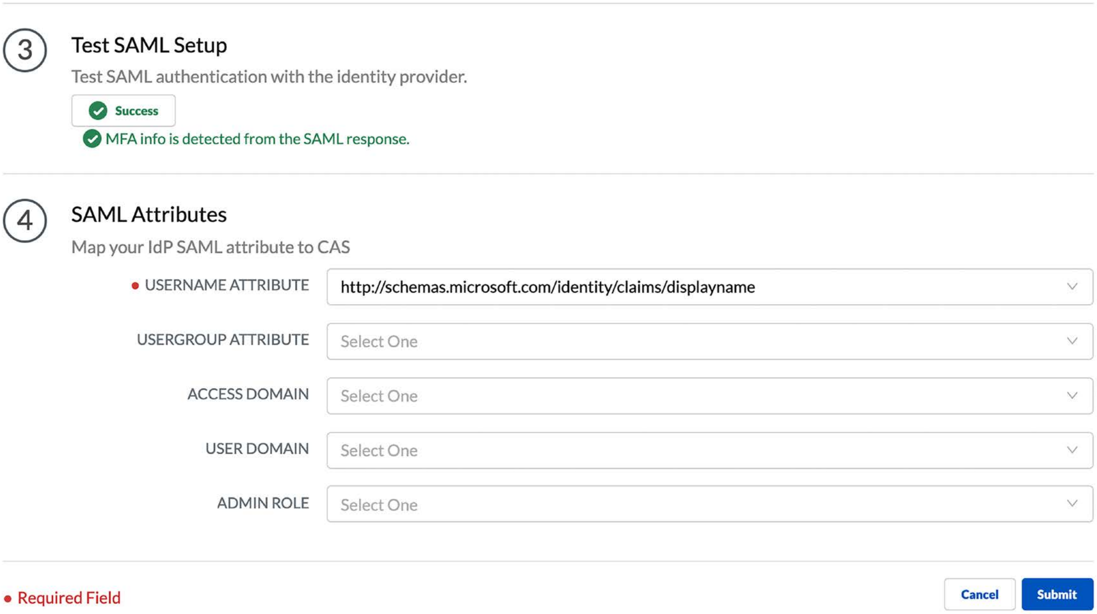


Figure 6.26: Configuring SAML attributes

If you have not set up the **Cloud Identity Engine – Cloud Authentication Service** enterprise application in Azure yet, the steps are outlined in the next section. Other IdPs will have similar steps.

Configuring Entra ID (Azure) enterprise applications

For the cloud authentication to work, you will need the **Cloud Identity Engine – Cloud Authentication Service** application. In the Azure search bar, search **Enterprise Applications**. Once there, click **New Application** and then search for **Palo Alto Networks Cloud Identity Engine - Cloud Authentication Service:**

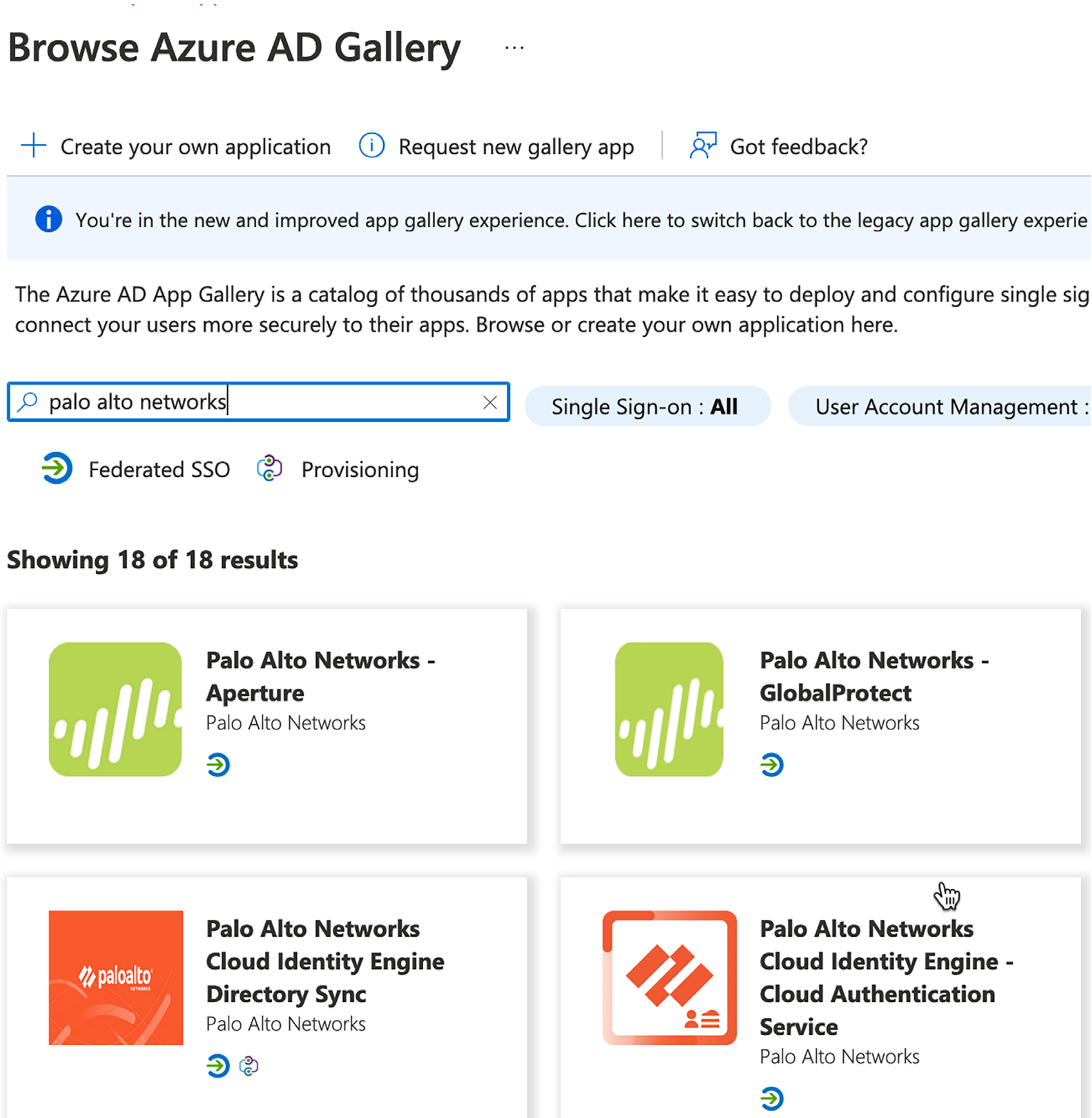


Figure 6.27: Azure enterprise applications

Once the application is created, first assign users or groups. These users will be able to authenticate during the activation process.

Next, select **Single Sign-on**, click **Upload Metadata File** at the top, and upload the `CAS-Metadata.xml` file you downloaded from the **Cloud Authentication** configuration page.

You will need to fill out the regional sign-on URL. Select the one that applies to you from this list:

Region	CIE regional URL
United States	`cloud-auth.us.apps.paloaltonetworks.com`
	`cloud-auth-service.us.apps.paloaltonetworks.com`
Europe	`cloud-auth.nl.apps.paloaltonetworks.com`
	`cloud-auth-service.nl.apps.paloaltonetworks.com`
United Kingdom	`cloud-auth.uk.apps.paloaltonetworks.com`
	`cloud-auth-service.uk.apps.paloaltonetworks.com`
Singapore	`cloud-auth.sg.apps.paloaltonetworks.com`
	`cloud-auth-service.sg.apps.paloaltonetworks.com`
Canada	`cloud-auth.ca.apps.paloaltonetworks.com`
	`cloud-auth-service.ca.apps.paloaltonetworks.com`
Japan	`cloud-auth.jp.apps.paloaltonetworks.com`
	`cloud-auth-service.jp.apps.paloaltonetworks.com`
Australia	`cloud-auth.au.apps.paloaltonetworks.com`
	`cloud-auth-service.au.apps.paloaltonetworks.com`
Germany	`cloud-auth.de.apps.paloaltonetworks.com`
	`cloud-auth-service.de.apps.paloaltonetworks.com`
United States - Government	`cloud-auth-service.gov.apps.paloaltonetworks.com`
	`cloud-auth.gov.apps.paloaltonetworks.com`
India	`cloud-auth-service.in.apps.paloaltonetworks.com`
	`cloud-auth.in.apps.paloaltonetworks.com`

Table 6.1: CIE regional sign-in URLs

The application page will now look similar to the following screenshot; note the region-specific **Sign on URL**. Go ahead and download the metadata file from the **Download** link next to **Federation Metadata XML** at the bottom of *step 3*, **SAML Certificates**:

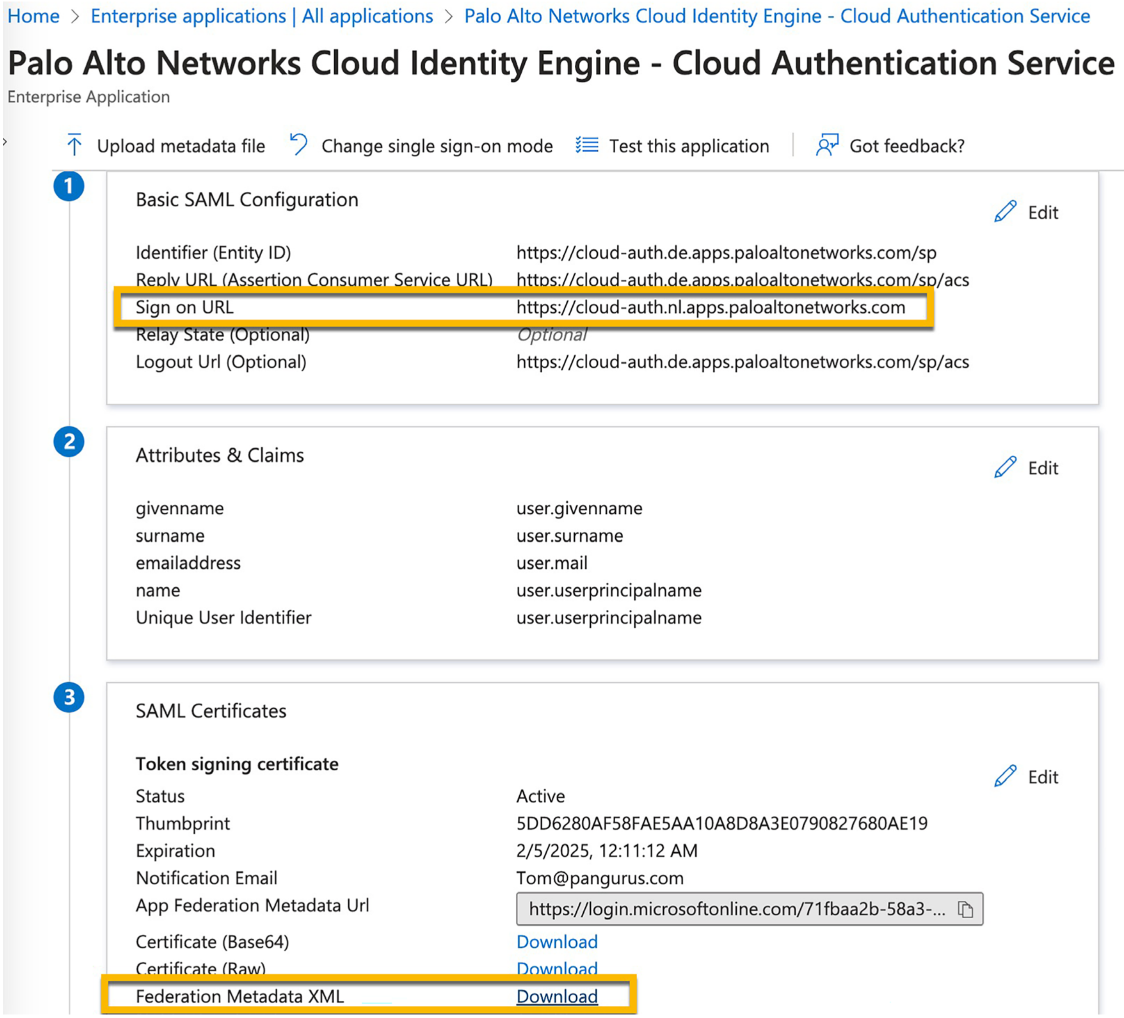


Figure 6.28: Azure enterprise application single sign-on

The last step is to add a new authentication profile on the firewall in **Device** | **Authentication Profile** and set the new profile as **Cloud Authentication Service.** Select the **Region**, **Instance**, and **Profile** as illustrated in the following screenshot. You can now use the new profile where needed:

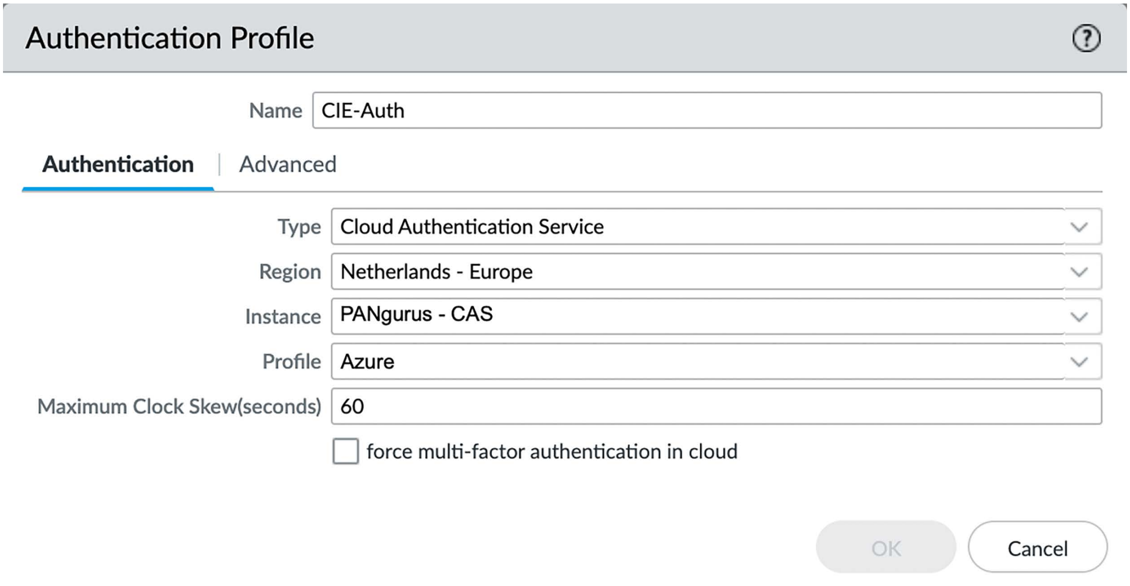


Figure 6.29: Cloud Authentication Service profile

You are now able to use group mapping to apply security rules to sets of users, leverage the CIE to combine on-premises and cloud directories, and leverage cloud authentication to simplify authentication across multiple firewalls and locations.

In the next section, we'll take a look at captive portals, an alternative way to identify users that combines with authentication.

Setting up a captive portal

A captive portal is a service that runs on the firewall and intercepts web sessions to have a user identify themselves. This can be a good addition to your user identification capabilities for unsupported operating systems that do not log on to the network, or guests that come into your network that you want to be able to identify.

It can also help pick up "strays"; for instance, a laptop may be used to roam a campus and hop SSIDs and access points, and it may be assigned a new IP address without generating a new logon event on AD. At this moment, the user becomes unknown and a captive portal can be triggered to have the user log in manually.

To set up a captive portal, we will need to be able to authenticate users, which we will cover first.

Authenticating users

To be able to authenticate users, we need to create an authentication profile that manages which protocol and server will be used. Create a new profile in **Device** | **Authentication Profile**:

1. In the **Authentication** tab, set the desired type (**LDAP**, **local**, **RADIUS**, **TACACS**, **SAML**, **Cloud Authentication**, or **Kerberos**).
2. In **Server Profile**, select a matching authentication server profile. You can create one from the dropdown by clicking the **New** link if you haven't created a profile yet. In most cases, this is just the IP and port of your server.
3. By picking the type, all the common attributes for your preferred authentication method are prepopulated. Make changes if any are needed (for example, LDAP may need **userPrincipalName** instead of the default **sAMAccountName**).
4. **Username Modifier** lets you change how the username is passed on to the authentication server. The default is `%USERINPUT%`, which passes along the user's exact input. `%USERDOMAIN%\%USERINPUT%` changes the user's input to `domain\username` and `%USERINPUT%@%USERDOMAIN%` changes the username to `user@domain.ext`. This could be helpful if your users log on with all kinds of different usernames and your authentication server prefers a certain flavor.
5. If your domain supports **Kerberos Single Sign-on**, enter the Kerberos domain and import the `kerberos` keytab so users are able to authenticate transparently. This URL can help you generate a keytab: `https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/ktpass`.
6. For an LDAP profile, the **Authentication** tab should look similar to the following screenshot:

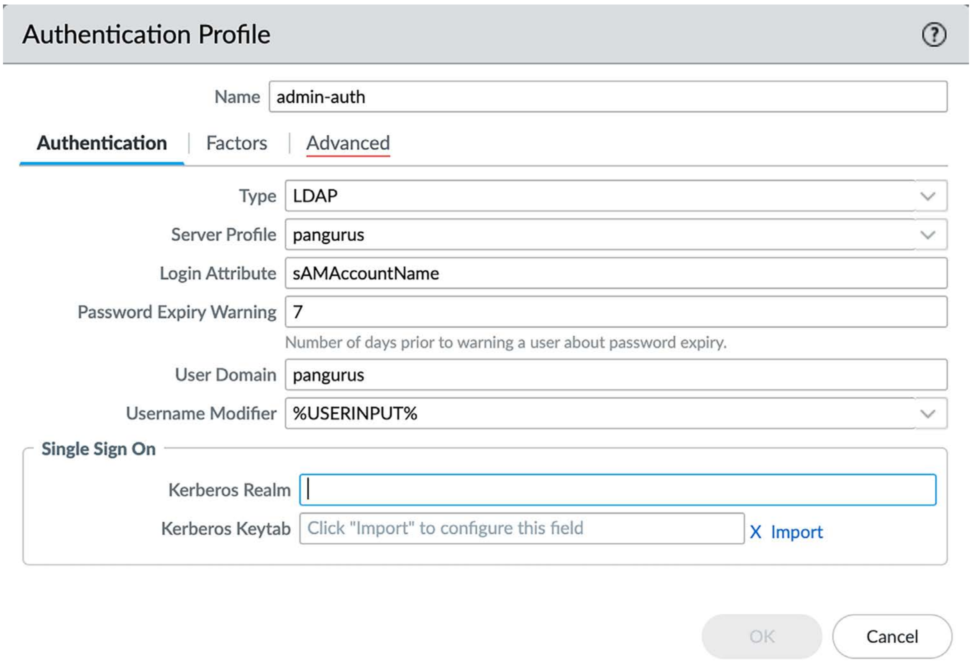


Figure 6.30: Authentication Profile

7. Optionally, you can enable MFA by checking **Enable Additional Authentication Factor** in the **Factors** tab and selecting which MFA provider to use.
8. In the **Advanced** tab, you must select which users will be allowed to authenticate. If all users are allowed to authenticate, add an entry and set it to **[all]**.
9. As per the following example, set **Account Lockout** to `4` failed attempts and set the lockout time to `30` to discourage brute-force attacks. A lockout time of `0` locks the account permanently until an administrator manually unlocks it. If **Failed Attempts** is `0`, no lockout will occur:

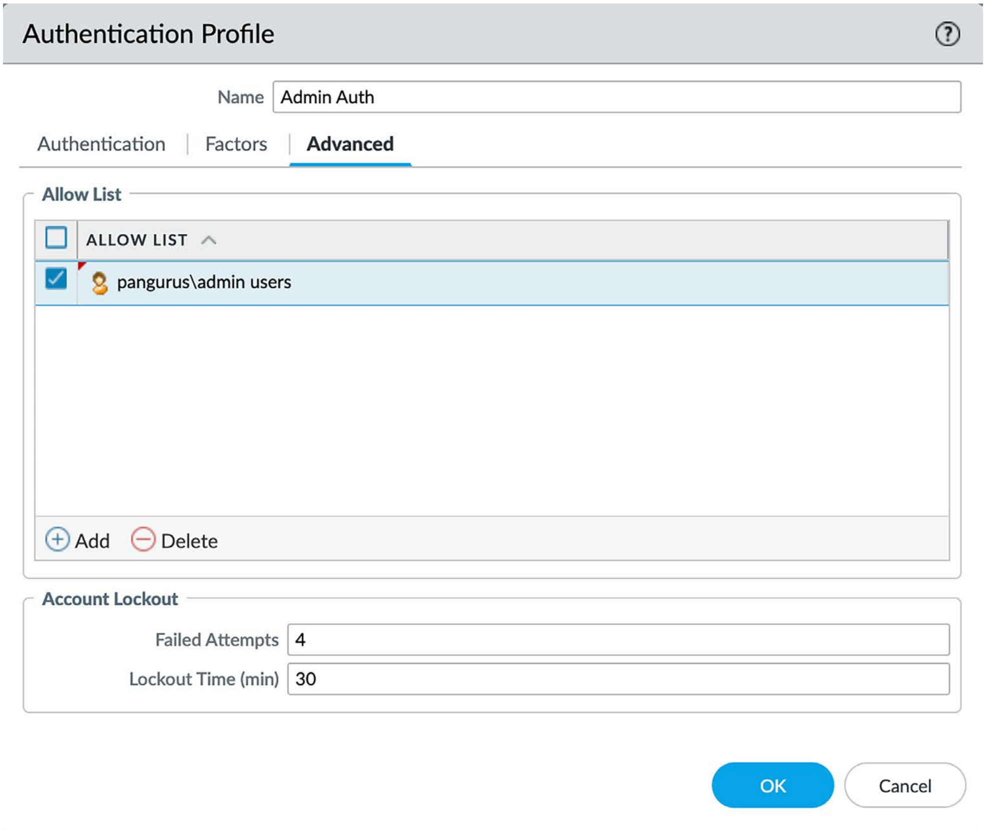


Figure 6.31: Advanced Authentication Profile settings

10. We will also need to create an **SSL/TLS server profile** so that the captive portal landing page uses a trusted certificate. You will first need to set up an appropriate certificate to use in the server profile. In **Device** | **Certificate Management** | **Certificates**, import a server certificate that's signed by your domain CA, or create a new self-signed server certificate that is signed by the self-signed root CA (the one we created for SSL decryption). This will ensure that the clients don't get a certificate error message if the root CA is properly trusted.
11. This certificate CN should be an FQDN (`cp.example.com`) that can be resolved on your internal DNS, or you should have the CN set to the IP address of the firewall interface that will be used as the redirect destination. The generation page should look similar to the following screenshot:

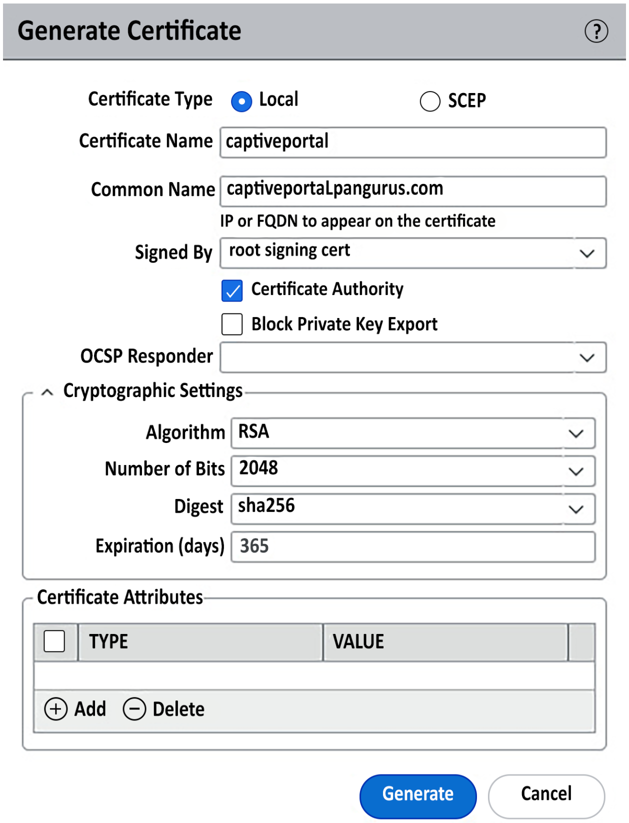


Figure 6.32: Generating a server certificate for the captive portal

12. In **Device** | **Certificate Management** | **SSL/TLS Service Profile**, create a new profile and name it `captiveportal`, add the captive portal certificate, and set **Min Version** to **TLSv1.2**, as you can see in the following screenshot:

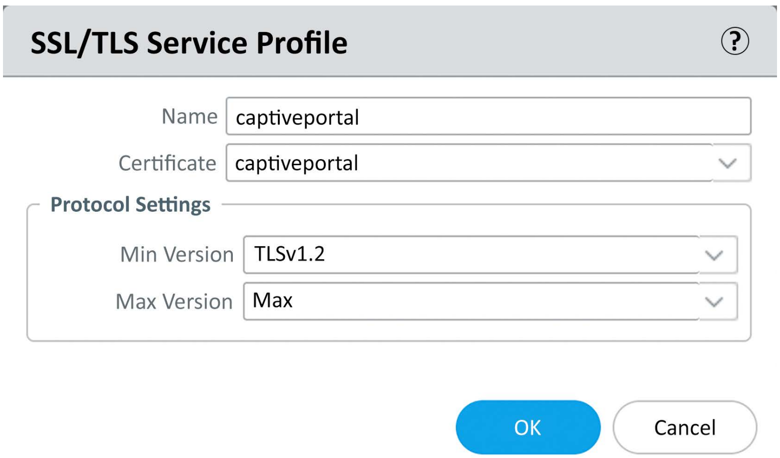


Figure 6.33: Creating an SSL/TLS service profile

13. Next, to accommodate a redirect page on the firewall interface, an **Interface Management Profile** needs to be created that has **Response Pages** enabled. Create one in **Network** | **Network Profiles** | **Interface Mgmt** and set an identifiable name. Enable **Response Pages**, and then enable **Ping** for troubleshooting. The profile should look as follows:

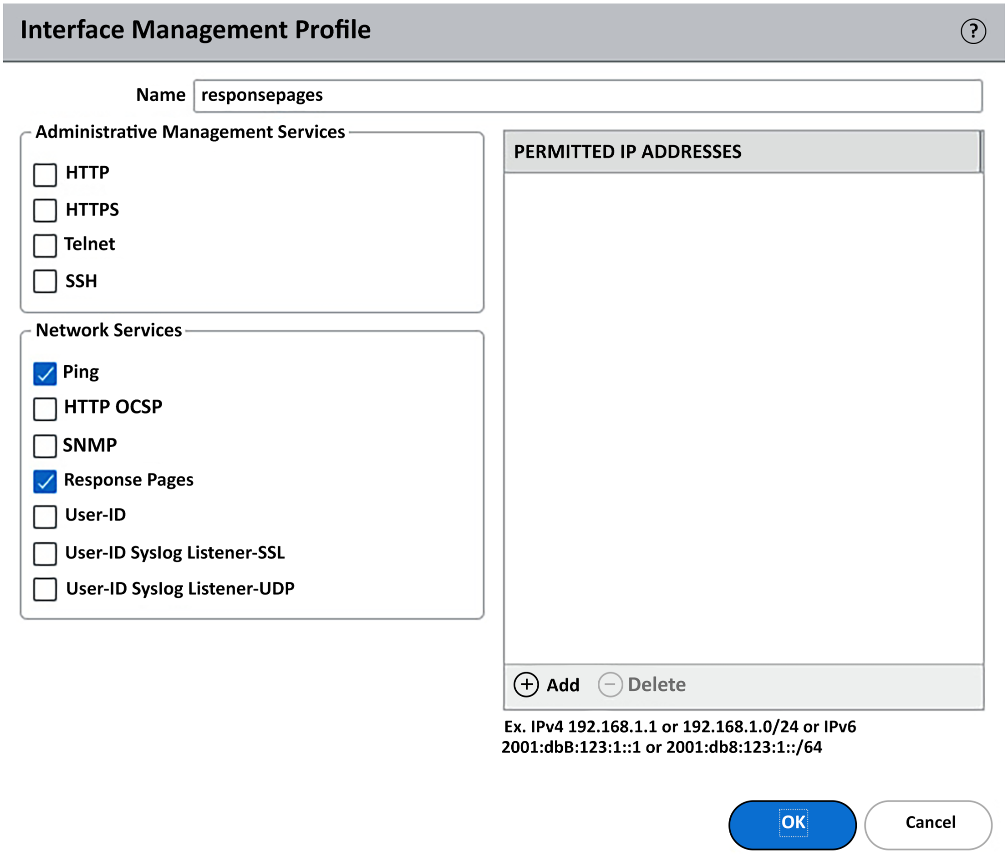


Figure 6.34: Interface Management Profile

14. Attach the profile to the physical or loopback interfaces that will serve the captive portal by going into **Network** | **Interfaces** | **Interface** | **Advanced** | **Other Info** | **Management Profile** and adding the profile.
15. Lastly, **Enable User Identification** must be enabled in the zones associated with the interfaces that host user subnets. Go to **Network** | **Zones** and check the box in every zone that has users who need to be intercepted. Do **not** enable this on an external zone.

Now that all the preparations have been made, we can set up the captive portal.

Configuring the authentication portal

In **Device** | **User Identification** | **Authentication Portal Settings**, edit the settings of the authentication portal:

1. Make sure **Authentication Portal** is enabled.
2. **Idle Timer** (the default is 15 minutes) indicates how long a user can remain idle before their session expires, and **Timer** (with a 60-minute default) indicates how long a user session lasts before the user needs to reauthenticate.
3. The **GlobalProtect** (**GP**) port is used to help GP pop up an MFA authentication dialog if MFA is configured and the user has GP installed; the default port should not be changed.
4. Set the **SSL/TLS Service** profile.
5. Set the **Authentication** profile.

There are two modes to choose from, with **Redirect** being the preferred one:

- **Transparent** mode intercepts the outbound session and impersonates the original destination URL while sending the user an HTTP `401` code to request authentication. Because the firewall is impersonating the remote site, the user may receive a certificate error.
- **Redirect** mode injects an HTTP `302` redirect message, prompting the browser to connect to the redirect host for further instructions. There, the user will be prompted for credentials or get authenticated transparently through NTLM or Kerberos. Redirect mode enhances the user experience while roaming by supporting session cookies and enabling a longer session timer as the cookie travels with the user.

Both modes will also work with HTTPS sessions if you have SSL decryption enabled.

To set **Redirect** mode, follow these steps:

1. Select the **Redirect** radio button to enable **Redirect** mode.
2. Enable **Session Cookie** and **Roaming**.
3. The default timeout of the session cookie is 1,440 minutes, which allows the user to roam for a day without needing to reauthenticate. Decrease this value if this is too long.
4. Set the **Redirect Host**. This needs to match the certificate CN you created in the SSL/TLS service step, being either an FQDN that translates to the data plane interface or the IP of the interface.

Certificate authentication enables you to set a certificate profile with which to authenticate users. User browsers that are not able to present the appropriate client certificate will not be able to authenticate. This is recommended in a high-security network where only known hardware is allowed to authenticate.

NTLM authentication can be used as a fallback transparent authentication mechanism if one of the User-ID agents is set up as an NTLM proxy. It is recommended to use Kerberos as transparent authentication instead (by means of the Kerberos SSO keytab) because Kerberos is a more secure authentication protocol.

Both Kerberos SSO and NTLM depend on the browser supporting either authentication method. If the client browser doesn't support these methods, the user will be presented with a web form to authenticate.

Your captive portal configuration should look as follows:

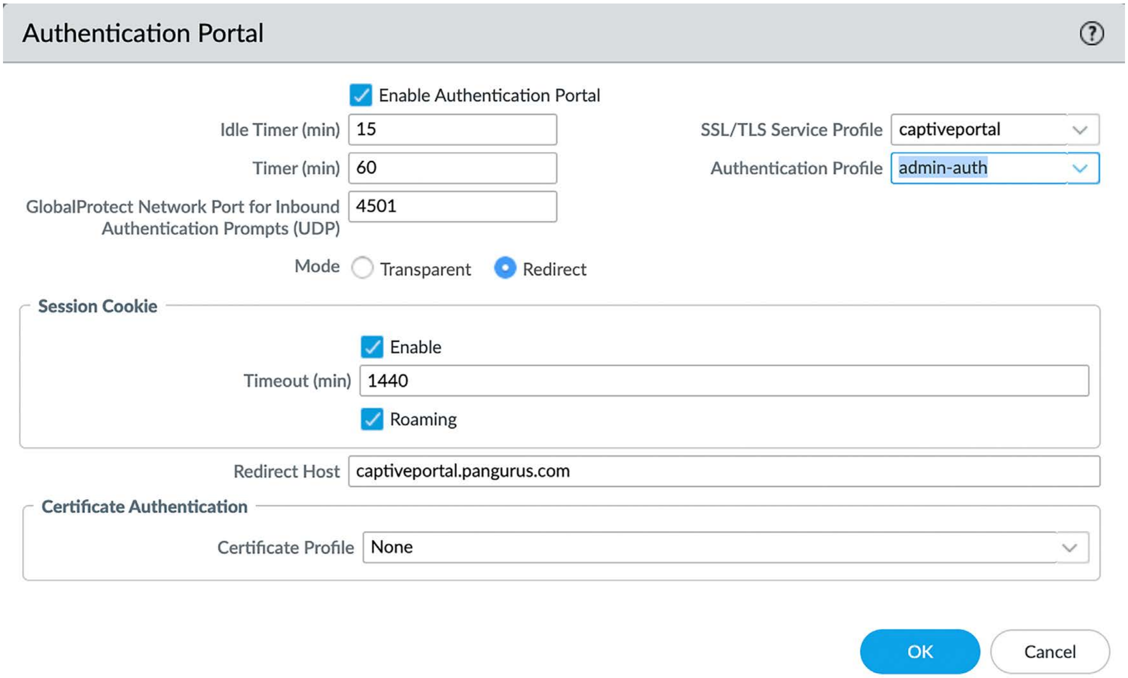


Figure 6.35: Captive portal configuration

The last step is to set up authentication rules in **Policies** | **Authentication**.

Rules are always evaluated from top to bottom, so the most specific rules should be at the top. If you want to allow users transparent authentication through Kerberos, create the rule for this first:

1. Set a friendly name and description.
2. In the **Source** tab, define the zones where users reside that could need captive portal authentication. In the **User** field, you have several options. Select **Unknown** so we can use this CP example to identify new users.

   This is what the other options can be used for:

   - **Any** includes all traffic to be intercepted, including already-known users.
   - **Pre-logon** includes remote users who are connected using the GlobalProtect pre-logon and have not logged in to their client system.
   - **Known-users** includes traffic for which the firewall already has a user-to-IP mapping (this can add a factor of **authorization** to accessing a certain resource).
   - **Unknown** includes traffic for which no user-to-IP mapping exists. *This is the main method to identify users who were not picked up by regular User-ID.*

- **Select** will only include traffic from specific users or groups (this could be used to specifically target guests while leaving employees alone).

3. In the **Destination** tab, provide the destination zone and any destination IP a user may access that warrants the authentication portal to be displayed. Leave the destination IP as **Any** if all traffic should trigger the portal.
4. In the **Service/URL** category, only the `http` service is included by default. Add `service-https` if you have SSL decryption enabled, and any other ports that might be useful. The **URL** category can be added if User-ID is mandatory for only specific URL categories or if explicit authorization is required for a category.
5. In **Actions**, set `default-browser-challenge`, which will use the Kerberos keytab if available in the authentication profile or will use **NTLM** via a User-ID agent.

If needed, you can also create a new authentication enforcement profile with a different authentication profile. This overrides the authentication profile used in the captive portal.

Your rule should look similar to the following screenshot:

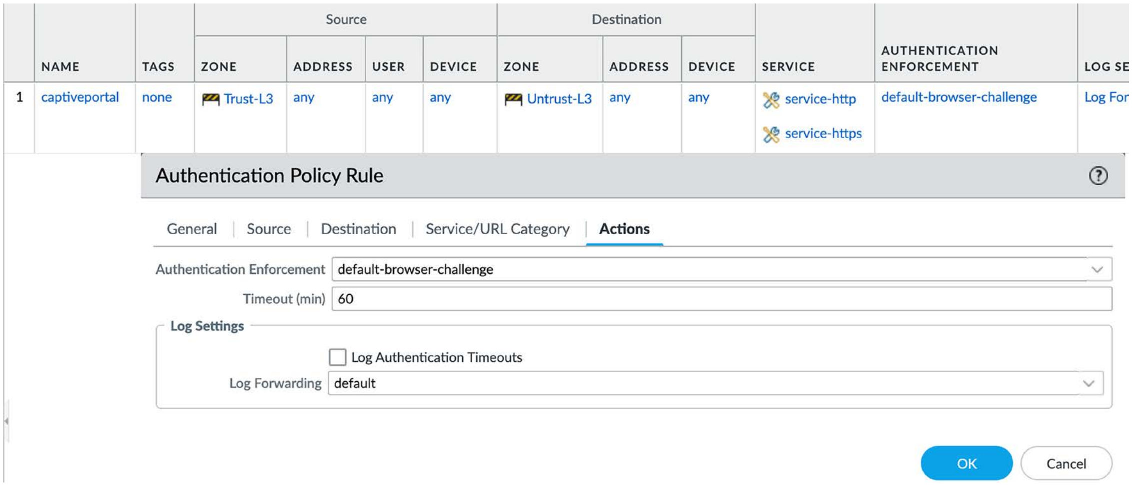


Figure 6.36: Authentication policy rule

Next, repeat *steps 1* through *5* and set the authentication enforcement to `default-webform`, which will present the user with a landing page to enter credentials.

If any address or subnet does not trigger a captive portal intercept (this could be a remediation server or guest proxy), repeat *steps 1* through *4*, set the authentication enforcement to `default-no-captive-portal`, and move it to the top of the rulebase.

Depending on which interface you associated the captive portal with, and which zone the users are connecting from, you may need to configure a security rule to allow these connections. You will find that the captive portal uses one of these ports:

- TCP `6080` is accessed by the client for NTLM authentication

- TCP 6081, if the captive portal was configured without an SSL/TLS service profile
- TCP 6082, when the captive portal is configured with a proper profile

With a little creativity, the captive portal can be active on several interfaces: the certificate needs to be set to an FQDN, each individual interface has the management profile enabled for response pages, and clients in each subnet are served a different IP (by DNS) for the associated redirect host.

Using APIs for User-ID

We saw earlier that you can forward syslogs to the User-ID agent to extract user information, but for those cases where you can't get the desired information from syslogs, you can also use an API to automate user-to-IP mapping, or manually add and delete user mappings.

You will first need to get an authentication key. Make sure the administrator account you are going to use for these operations has API access.

> To get a key, you can use this URL in a browser:
>
> ```
> https://<YourFirewall>/api/?type=keygen&user=<Username>&password=<Pass
> word>
> ```

Alternatively, you can use cURL at the command line:

```
curl -k -X GET 'https://<YourFirewall>/
api/?type=keygen&user=<username>&password=<password>
```

That would give you the following output:

```
<response status="success">
<result>
<key>
LUFRPT1TWFhUNWUk5N1Fjd3ZnMzh3MXlTOVJyb0kxSG5IWk5QTkdPNw==
</key>
</result>
</response>
```

You can now use this key in combination with any API command to change things on the firewall or request information. For example, you can request a list of users by using the following URL in your browser:

```
https://10.0.0.2//api/?type=op&cmd=<show><user><user-ids><all></all></user-ids></user></
show>&key= LUFRPT1TWFhUNWUk5N1Fjd3ZnMzh3MXlTOVJyb0kxSG5IWk5QTkdPNw==
```

Alternatively, you can again use cURL at the command line:

```
curl -k -X GET 'https://10.0.0.2//api/?type=op&cmd=<show>
<user><user-ids><all></all></user-ids></user></show>&key=
LUFRPT1TWFhUNWUk5N1Fjd3ZnMzh3MXlTOVJyb0kxSG5IWk5QTkdPNw=='
```

```
<response status="success"><result><![CDATA[
User Name Vsys Groups
-------------------------------------------------------------------
example\tomfromit vsys1 cn=it,cn=users,dc=example,dc=com
example\jimfromhr vsys1 cn=hr,cn=users,dc=example,dc=com
Total: 3
* : Custom Group
```

Important note

You can browse through all the available API commands by logging in to your firewall and then replacing the URL with `https://<YourFirewall>/api`.

To add users, you can use the following command:

```
curl -F key=<APIkey> --form file=@<file> "https://<YourFirewall>/
api/?type=user-id"
```

For the file that will be used as the source, use the following syntax to add a user:

```
<uid-message>
<version>1.0</version>
<type>update</type>
<payload>
<login>
<entry user="domain\user" ip="x.x.x.x" timeout="60">
</entry>
</login>
</payload>
</uid-message>
```

This is the syntax used to remove a user:

```
<uid-message>
<type>update</type>
<version>1.0</version>
<payload>
<logout>
<entry user="domain\user1" ip="x.x.x.x">
</entry>
</logout>
</payload>
</uid-message>
```

You can add and remove users in the same update by simply adding `login` and `logout` syntax inside the payload.

You can add or remove multiple users at once by adding entries inside the login or logout elements:

```
<uid-message>
<type>update</type>
<version>1.0</version>
<payload>
<login>
<entry user="domain\user1" ip="x.x.x.x" timeout="60">
</login>
<logout>
<entry user="domain\user3" ip="y.y.y.y">
<entry user="domain\user3" ip="z.z.z.z">
</logout>
</payload>
</uid-message>
```

You can also add users to group(s):

```
<uid-message>
<version>1.0</version>
<type>update</type>
<payload>
<groups>
<entry name="groupA">
<members>
<entry name="user1"/>
</members>
</entry>
<entry name="groupB">
<members>
<entry name="user2"/>
</members>
</entry>
</groups>
</payload>
</uid-message>
```

In this section, you learned how to use APIs to control the creation and deletion of user-to-IP mapping entries and to add or remove users from groups. In the next section, we'll see how we can leverage User-ID and URL filtering to protect users from phishing attacks.

User credential phishing prevention

With phishing being a significant attack vector, user education is a very hot topic in many corporations' cybersecurity awareness programs. Being able to prevent users from sharing their credentials on an untrusted website is a good second line of defense in case a user is tricked into submitting credentials to a malicious site.

As you can see in the following screenshot, in the URL Filtering security profile, there is a column called **User Credential Submission**. Any categories set to **block** will not allow users to submit credentials.

A user will not be allowed to log on if a site is categorized as belonging to the **malware** category and if **malware** is set to **block** for **User Credential Submission**.

Any category set to **continue** will first warn the user that they are submitting credentials to a site and will require acknowledgment of their actions. Any category set to **alert** (with logging) or **allow** will let the user submit their credentials:

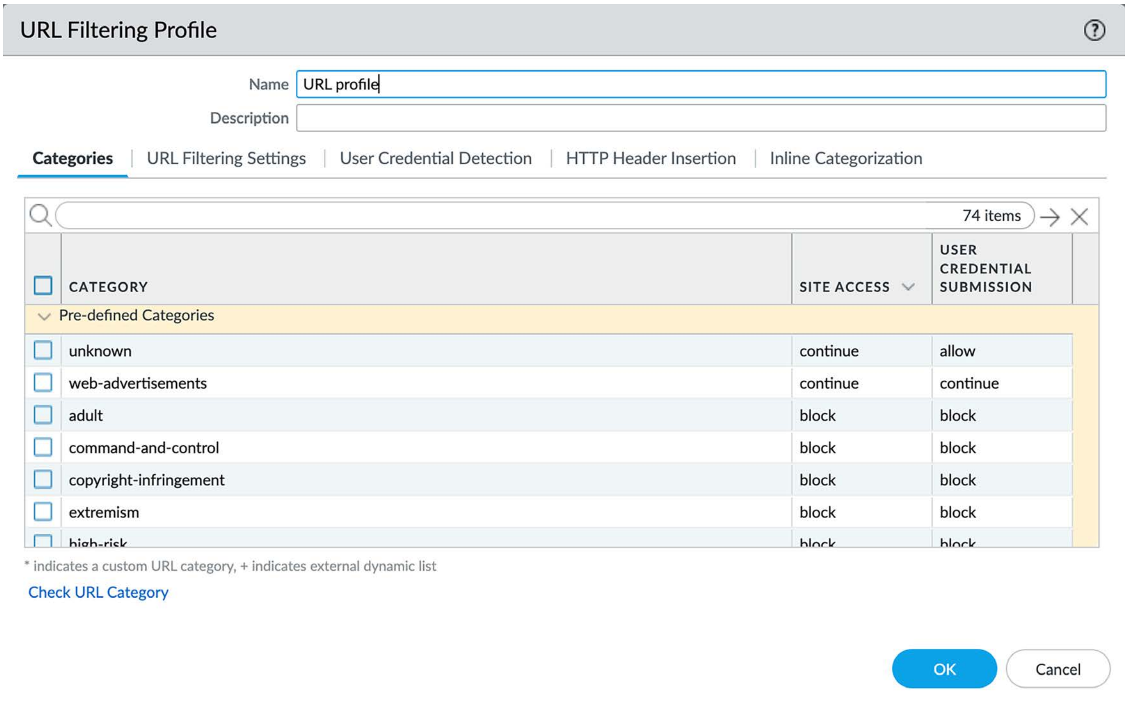


Figure 6.37: The URL Filtering Profile page

Important note

SSL decryption is required to be able to look inside a flow and intercept login credentials submitted by the user for inspection.

Take this one step further and access the **User Credential Detection** tab to enable the detection of actual corporate user credentials. This will help distinguish between users logging on to Facebook with their private account and those doing so with their corporate emails, as well as helping to distinguish whether they are using the same password as they do in the corporate environment.

If the submitted credentials do not match the detection method result, the user will be allowed to log on; otherwise, the **User Credential Submission** action is applied.

There are three options available, and all methods require User-ID to be already set up on the firewall. Verify that the user-to-IP mapping uses the same format as the primary username in LDAP (for example, if the primary username is **UserPrincipalName**, the user-to-IP mapping should also display UPN usernames). The three options available are as follows:

- **Use IP User Mapping**: This lets the firewall compare the credentials submitted to the website to the username in the user-to-IP mapping that it gets from User-ID. If a match is detected, the URL filtering profile will apply the action defined in the **User Credential Submission** column.
- **Use Group Mapping**: The firewall uses User-ID group mapping to match the submitted username to a username known in the group mapping profile. This method only matches usernames against LDAP group membership.
- **Use Domain Credential Filter**: This enables the firewall to verify the username and password of a submitted credential and check whether they belong to the logged-in user. This method is the most thorough as it can also detect password matches, but it does require that a User-ID agent and a User-ID credential service add-on (`UaCredInstall64-x.x.x-x.msi` from the support portal software updates) are installed on a **Read-Only Domain Controller** (**RODC**). Since you must install these agents on a separate domain controller, do not use the User-ID agent to collect user-to-IP mappings. The credential service add-on creates a bloom filter for all the usernames and passwords that the firewall can periodically fetch from the User-ID agent to then match credential submissions. Usernames and passwords are not saved on the firewall.

Each method allows you to set a log severity when a valid credential is detected. By default, URL filtering logs have a severity of **informational**; set the severity to **medium** or higher.

As shown in the following screenshot, the **Credential Detected** column can be enabled in the URL filtering log to reveal whether corporate credentials were matched in browsing sessions:

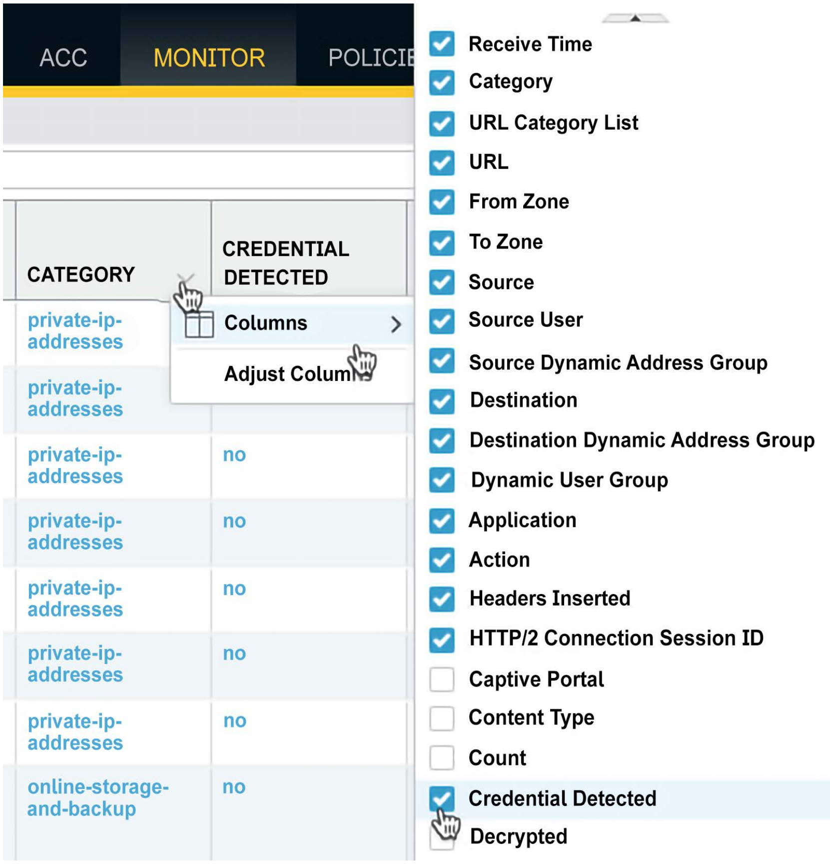


Figure 6.38: Enabling the CREDENTIAL DETECTED column

Implementing this feature will ensure your users do not accidentally or deliberately share corporate credential information outside your network and will discourage the use of corporate credentials for personal sites.

Summary

In this chapter, you learned how to set up the User-ID agent and the TS agent software agents on a server, and also how to properly configure agentless configuration on a firewall. You learned how LDAP groups can be leveraged to categorize users and apply security and which user attributes can be used to tailor the configuration to your needs. You also learned methods to prevent users from accidentally submitting corporate credentials to untrusted website categories.

In the next chapter, we will learn how to manage and consolidate configuration for multiple firewalls using Panorama.

7

Managing Firewalls Through Panorama

In this chapter, we will learn about Panorama, a central management platform that enables administrators to manage firewalls located in different locations or in the cloud in real time. You will learn how to create shared objects and policies, as well as how to use device groups to add some region- or purpose-based policies that can be deployed to multiple similar firewalls. You will also learn how to manage logs, push content updates from a single location, and keep track of your **inventory**.

In this chapter, we're going to cover the following main topics:

- Setting up Panorama
- Device groups
- Setting up templates and template stacks
- Panorama management

By the end of this chapter, you will be able to centrally manage all of your firewalls and consolidate shared configuration among groups of devices.

Technical requirements

For this chapter, you are expected to have a basic understanding of how to manage and maintain **virtual machines (VMs)** on any of the major hypervisor technologies (KVM, NSX, Hyper-V, ESX, and so on) or cloud providers (Azure or Amazon Web Services).

A copy of the Panorama configuration we touch on in this chapter can be found at `https://github.com/PacktPublishing/Mastering-Palo-Alto-Networks-Third-Edition`.

Setting up Panorama

Before you get started, you will first need to decide how you want to deploy Panorama as there are many options that can influence your choices. Panorama can be deployed as a physical appliance like the M-300 (up to 16 TB of storage) or the M-700 (up to 48 TB of storage) appliances, or a VM image, both locally and in the cloud. All of these options have advantages over the others. A physical appliance can either be deployed as a Panorama instance or as a Log Collector, which can be bundled and spread out to make it more resilient and bandwidth efficient, while keeping physical control over logs. VMs are very easy to deploy and run on nearly all common hypervisors, so no hardware is needed to deploy them. Cloud-based Panorama allows the admin optimal access from any location for management and firewall access.

If Panorama is to be deployed as a VM, the first step is to determine the *minimum* system prerequisites. Before you can do that, you need to determine the flavor of Panorama that be deployed:

- **Management Only** mode will separate firewall management from log collection. In **Management Only** mode, Panorama will not be able to collect logs locally and will need to rely on Log Collectors or Strata Logging Service.
- **Panorama** mode allows Panorama to serve two purposes at once: central management of your firewalls and collecting logs from these firewalls.
- **Log Collector** mode removes the central management functionality from this VM and restricts its function to receiving logs from the managed firewalls. If a Log Collector VM is deployed, at least one other VM in **Panorama** or **Management Only** mode needs to be deployed so firewalls can be directed to send logs to the collector.

Next, you can determine your VM's minimum requirements:

Panorama VM in Management Only mode		Panorama VM in Panorama mode		Panorama VM in Log Collector mode	
System Disk 81 GB		System Disk 81 GB		System Disk 81 GB	
Up to 500 managed devices	16 CPUs 32 GB memory No logging disks	Up to 500 managed devices Up to 10k logs/sec	16 CPUs 32 GB memory 4 x 2 TB logging disks	Up to 15k logs/sec	16 CPUs 32 GB memory 4 x 2 TB logging disks
Up to 1,000 managed devices	32 CPUs 128 GB memory No logging disks	Up to 1,000 managed devices Up to 20k logs/sec	32 CPUs 128 GB memory 8 x 2 TB logging disks	Up to 25k logs/sec	32 CPUs 128 GB memory 8 x 2 TB logging disks

Up to 2,500 managed devices	32 CPUs 256 GB memory Additional disk with 92 GB capacity	Not supported		
Up to 5,000 managed devices	56 CPUs 256 GB memory 240 GB SSD on M-series, 224 GB on ESXi (increase manually after deploying on the 81 GB)	Not supported		
Minimum requirements for extended logging capacity		2 TB – 8 TB, 16 CPUs, 63 GB memory 10 TB – 24 TB, 16 CPUs, 128 GB memory Logging disks must be 2 TB in size each		

Table 7.1: Panorama minimum requirements

The following resource lists all the requirements in greater detail: `https://docs.paloaltonetworks.com/panorama/11-1/panorama-admin/set-up-panorama/set-up-the-panorama-virtual-appliance/setup-prerequisites-for-the-panorama-virtual-appliance`.

Important note

Panorama on ESXi does not support quiesced snapshots or VMware vMotion.

The next step is to configure Panorama so that it can manage firewalls.

Initial Panorama configuration

Panorama can be deployed in a number of virtual environments, including KVM, NSX, Hyper-V, and ESX, and cloud providers such as Amazon Web Services and Microsoft Azure. So, for example, you can simply download the Panorama **Open Virtual Appliance** (**OVA**) image from `https://support.paloaltonetworks.com`, in the **Software** section, and deploy it in a VMware ESXi environment, as shown in the following screenshot:

Figure 7.1: Deploying a Panorama OVA in ESXi

Deploying one of the pre-packaged VMs has the advantage that you don't need to choose the correct guest OS or select the correct number of CPUs or the amount of RAM; all these parameters are taken care of automatically, but it's always wise to double-check the VM parameters before spinning it up for the first time.

Once the deployment is complete and you have double-checked that your VM meets the requirements, start your virtual appliance and register Panorama in the support portal using the **UUID** and **CPUID**.

To ensure that Panorama is in good working condition, navigate to **Panorama** | **Setup** | **Management** and follow these steps to ensure that the base system configuration is in good order:

1. Set **Hostname**, **Domain** (`example.com`), and **Login Banner**.
2. Set an appropriate **Time Zone**, **Date**, and **Time** to match the location Panorama is hosted in, or mostly managed from. All logs are received in UTC with their respective local deviation. Setting Panorama to an appropriate time zone ensures all logs are displayed at the appropriate local time (for example, an event at 5 a.m. PDT will show as 3 a.m. if Panorama is set to EST).
3. Ensure that the serial number you received after registration has been set properly.

   The resulting configuration should look similar to the following screenshot:

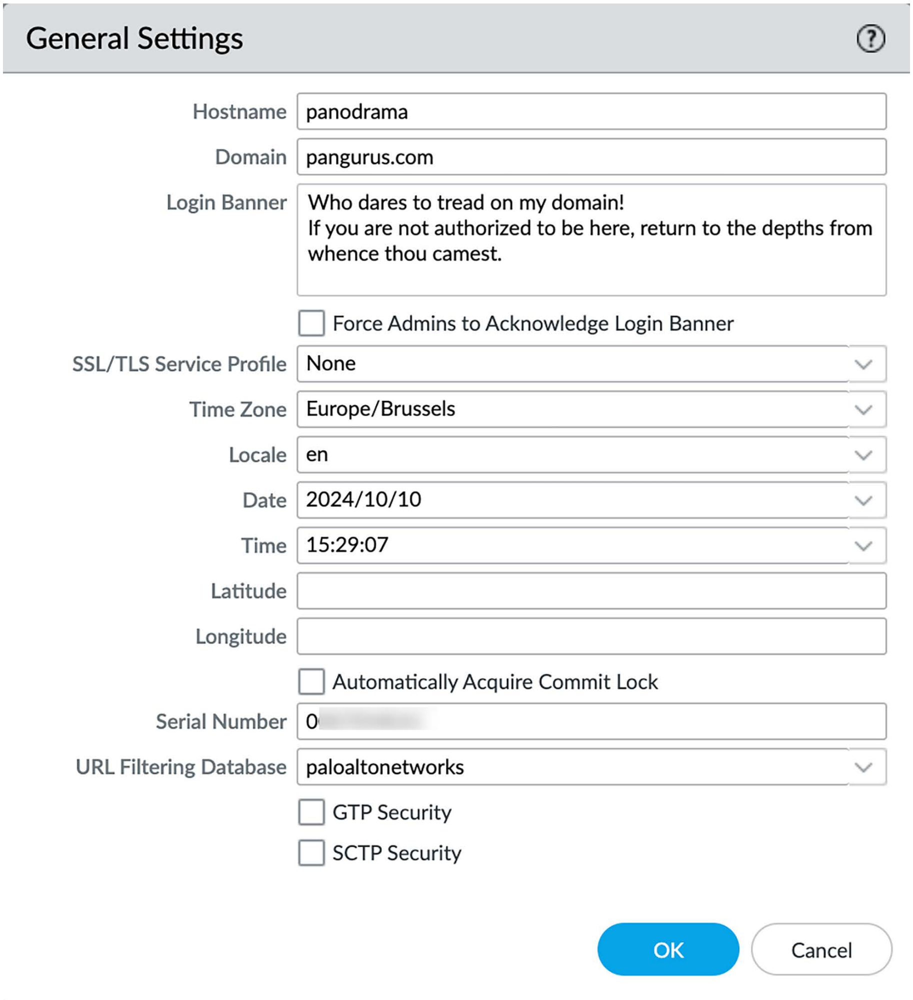


Figure 7.2: The Management page's General Settings pane

4. Set **SSL/TLS Service Profile** with the minimum version set to `TLS1.2` and add in the **General Settings**.

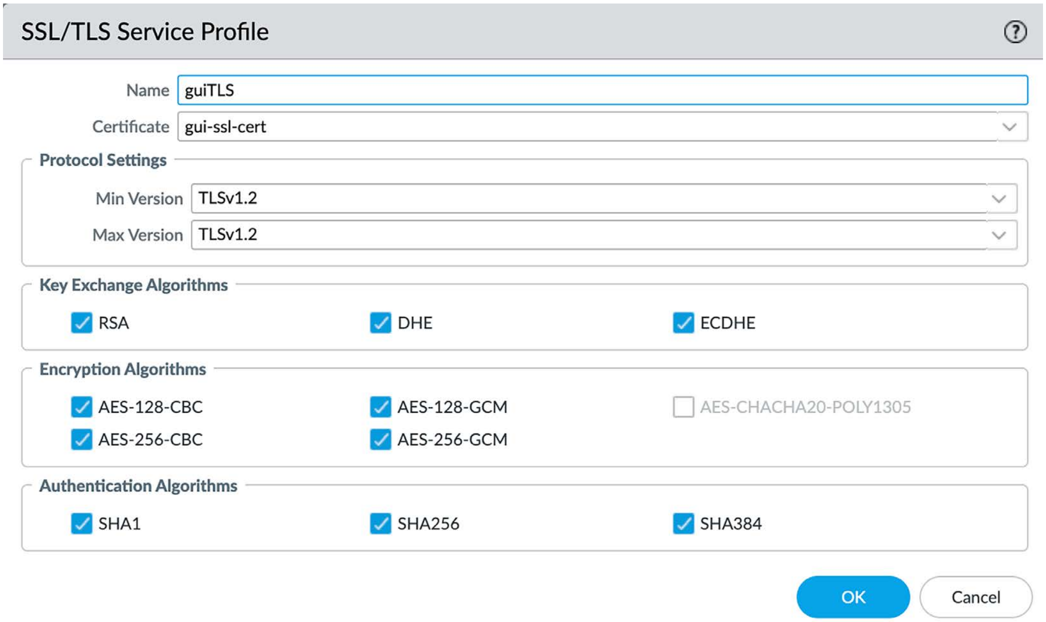


Figure 7.3: SSL/TLS Service Profile

Next, review **Secure Communication Settings**. By enabling **Customize Secure Server Communication**, you can manually set **SSL/TLS Service Profile** and **Certificate Profile**, and then create a list of identifiers that can be used for communication between the firewalls and Panorama, as you can see in the following screenshot. This requires the firewalls and Panorama to be provisioned with an SSL/TLS service profile that uses certificates signed by the same root **Certificate Authority** (**CA**) so that they can establish trust. Currently, up to 25 identifiers can be added:

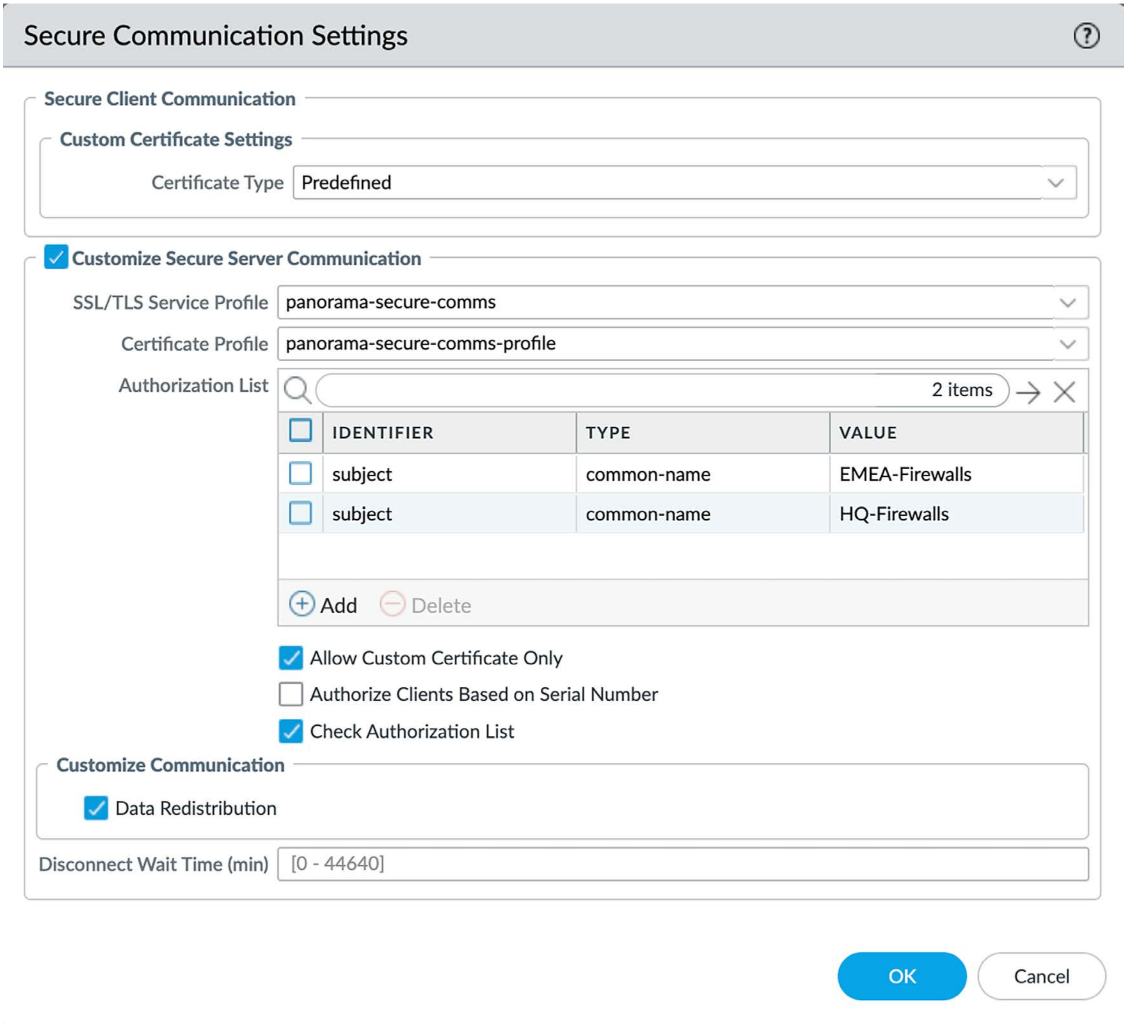


Figure 7.4: Secure Communication Settings

If a custom certificate is used for secure communications, remember to set a timely reminder to renew the certificate before it expires or you may lose access to the firewalls until the certificates are renewed.

The same certificate can be used to secure User-ID redistribution among Panorama and the firewalls by checking the **Data Redistribution** checkbox in the **Customize Communications** section.

The firewall side will look similar to the following screenshot. You can find this in **Device** | **Setup** | **Management** | **Secure Communication Settings**:

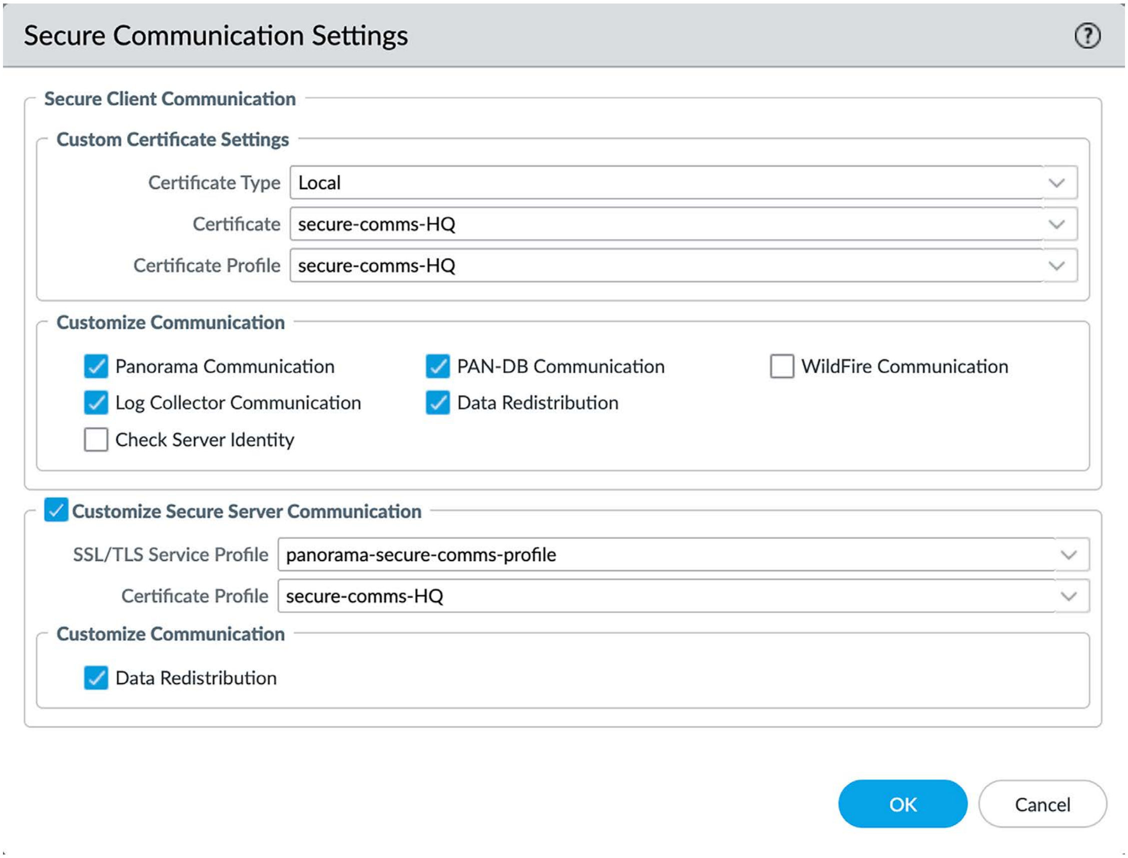


Figure 7.5: Firewall secure communications

Then, go to **Panorama** | **Setup** | **Services** and set the following parameters:

1. Set the DNS and NTP servers.
2. You can change the FQDN object refresh interval and set a timer to expire stale FQDN entries. By default, FQDN objects are refreshed every `1800` seconds and stale entries (entries that can't be updated) are not timed out.
3. Add a proxy configuration if the Panorama's outbound connections (dynamic updates and such) need to be redirected through a proxy server.

Important note

Beware of timing out stale entries if, for example, only one FQDN object exists in a security rule as the source or destination. If it goes stale, timing out may cause unexpected behavior as this would remove the object from the security rule at the data plane level. In this case, it may be prudent to not time out stale entries.

Your **Services** configuration should look similar to the following screenshot:

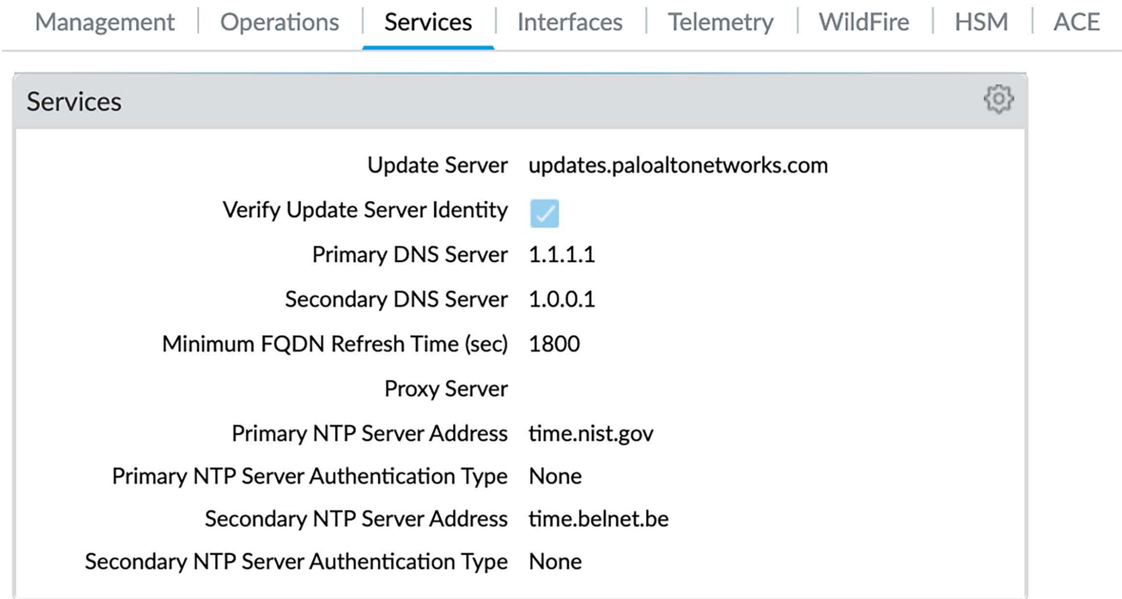


Figure 7.6: Panorama services

Lastly, go to **Panorama** | **Setup** | **Interfaces** and set the following configurations:

1. In the **Management** interface, set **IP address**, **Netmask**, and **Default Gateway**.
2. If Panorama can also be reached over the internet, add a **Public IP Address** value. This allows remote firewalls to communicate with the public IP assigned to Panorama.
3. If Panorama will be used to redistribute user-to-IP mappings to downstream firewalls, you need to enable **User-ID** here.
4. **Permitted IP Addresses** determines which IP addresses are allowed to connect to the management interface. If you choose (recommended) to set restrictions, make sure to add the firewall IP addresses here as well.
5. Remember that external firewalls that communicate over the internet to Panorama may need to be added by their public IP, while internal firewalls will likely need to be added by their actual management, or service route, IP.

6. Additional interfaces can be enabled and used to take some load off the management interface or provide an **Out-of-Band** (**OoB**) connection for certain services, such as **Device Management**, **Collector Group Communication**, and **Device Deployment** (pushing out software and updates to firewalls).

Your management interface settings should now look similar to the following screenshot:

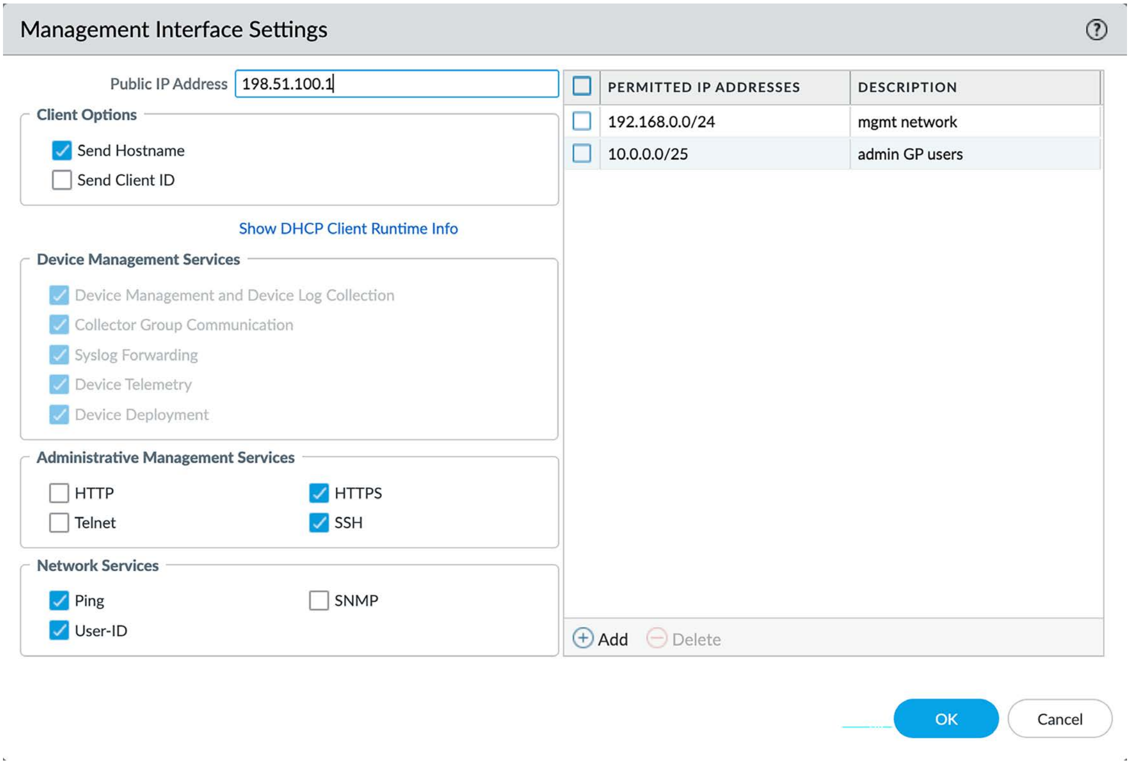


Figure 7.7: Panorama interface

Note that there is no IP configuration in this screenshot. This is because the Panorama featured is hosted on Azure and is set to DHCP mode by default. A locally hosted Panorama will have **IP**, **Netmask**, and **Default Gateway** fields.

Now that Panorama is set up, the next step is to make sure it can receive logs from the firewalls.

Panorama logging

Once deployed, Panorama can be configured to operate in one of two modes: **Panorama** mode or **Management Only** mode. By default, the VM is deployed in **Management Only** mode. In this mode, the following conditions apply:

- The appliance does not support receiving logs forwarded by firewalls directly.
- Either a Log Collector group using Panorama appliances (M-100 through M-700) or VMs set to **Log Collector** mode need to be configured, or cloud logging (Strata Logging Service, formerly Cortex Data Lake) needs to be enabled.

The second operational mode, **Panorama** mode, has the advantage of being more scalable for medium environments:

- Panorama can have 1 to 12 disks or partitions of 2 TB each, up to a total of 24 TB in RAID (10) configuration.
- Additional storage can be added by deploying logging appliances and configuring Log Collectors.

Panorama deployments that have been around for longer may still be in **Legacy** mode, which has been deprecated:

- Logs are stored in a single log partition that is part of the system disk (`sda`).
- The default log partition can be replaced by adding a second disk (`sdb`) of up to 8 TB (pre-ESXi 5.5, this capacity was limited to 2 TB).

> **Important note**
>
> **Legacy** mode was discontinued in PAN-OS 9.0 as a configurable mode and only exists on Panorama instances that were installed on PAN-OS 8.1 or earlier. Upgrading to PAN-OS 9.0 from **Legacy** mode will retain this mode, but once the system is changed to **Panorama** or **Management Only** mode, it can no longer be reverted.

One drawback of **Legacy** mode is that an existing log partition cannot be expanded, so if you initially add a 2 TB drive and later need a larger one, you will need to replace the 2 TB disk with a larger one. **Legacy** mode also supports the Log Collector configuration, using physical appliances as Log Collectors.

If you want to receive logs directly on a Panorama appliance, you will need to switch to **Panorama** mode. You can switch from any mode to **Management Only** or **Panorama** mode, but you can't go back to **Legacy** mode once you have changed to either of the new modes:

```
> request system system-mode management-only
> request system system-mode Panorama
```

Once you execute this command to change the system mode, you will be prompted to confirm it by pressing *Y* if you are sure, after which Panorama will reboot in the new mode.

Adding disks to Panorama

To be able to add disks to it, Panorama needs to be shut down. You can add one disk up to 8 TB for **Legacy** mode and any size larger than 2 TB (but in 2 TB increments) for **Panorama** mode. A **Panorama** mode VM will automatically partition any disk into 2 TB partitions, so you can add a 24 TB (or smaller) disk at once and **Panorama** mode will automatically split it into 12 2 TB partitions. **Management Only** mode will not take any actions with disks added to its virtual appliance.

If an additional disk is added after a second drive has already been added (**sda** and **sdb**), **sdc** may not immediately become active and may need to be enabled by an admin. The status of the new disk can be checked using the following command:

```
> show system disk details
    Name : sdc
    State : Present
    Size : 2048000 MB
    Status : Available
    Reason : Admin enabled
    Name : sdc
    State : Present
    Size : 2048000 MB
    Status : Available
    Reason : Admin disabled
```

If the new disk is set to `Admin disabled` it can be enabled by issuing the following command:

```
> request system disk add sdc
```

Log collection options

There are three main ways to collect logs:

- Using **Legacy** mode
- Using Strata Logging Service
- Using Log Collectors

In **Legacy** mode, nothing needs to be set; Panorama will simply register logs to its local database.

Legacy mode was deprecated in PAN-OS 9.0, so it can only exist in Panorama systems that were deployed in earlier versions that have not been changed to a newer mode since.

Strata Logging Service logs to the cloud. The advantage is that it is scalable, it is located virtually *near* your firewalls so that you don't need to deploy Log Collectors all over the place, and, depending on your log volume, it may cost less than buying appliances or backhauling logs over expensive WAN links. Enabling it is fairly simple:

1. Make sure Panorama is already registered and has a valid support license by going to `https://support.paloaltonetworks.com` and then clicking on **Assets**.
2. Acquire a cloud services auth code from your sales contact.
3. Activate Strata Logging Service via **Assets** | **Cloud Services** | **Activate Cloud Service Auth Code**.
4. You will be asked for the Panorama serial number and logging region. Enter it.
5. Once you accept the terms, the license will automatically be added to Panorama.
6. Next, click on **Generate OTP**.
7. Select **Panorama** and copy the **One Time Password** (**OTP**) to the clipboard (or to a text editor, as we will need it in a moment).
8. Access your Panorama instance and navigate to **Panorama** | **Licenses** to select **Retrieve license keys from license server**.
9. Access **Panorama** | **Plugins** and click on **Check Now**. Download the latest **Cloud_Service** plugin and then proceed to install it.
10. A new item will appear in the navigation to the left, just below **Plugins**, called **Cloud Services**. Access the **Status** submenu, paste the OTP, and then click **Verify**.

You can check whether connectivity with Strata Logging Service is successful by reviewing **Panorama** | **Cloud Services** | **Status**.

Deploying Log Collectors

Lastly, the most common deployment is to use **Log Collectors**. Log Collectors need to be deployed before they can be added to Panorama. If Panorama has been set to **Panorama** mode, it will also function as a Log Collector. You will need to add the local Panorama instance as a Log Collector before managed firewalls can forward logs to it.

Additionally, you can add a Panorama HA peer and additional M appliances to increase the capacity and fault tolerance.

Important note

The M appliance does not have a web interface enabled unless it is configured in **Panorama** mode. Connect to its console via terminal emulation, `TTY (9600-8-N-1)`, or use SSH on the **management port**.

Before you can add an M appliance as a Log Collector, it needs to be prepared:

1. Configure the management interface. Set `DNS` and `NTP`.
2. Register the device and add licenses.
3. Set the system to `logger` mode:

```
> request system system-mode logger
```

4. Build RAID pairs by adding `A1`, `A2`, `B1`, `B2`, and so on, depending on the number of disks in your system:

```
> request system raid add A1
> request system raid add A2
> show system raid detail
Disk Pair A       Available
Status            clean
```

5. Add the Panorama IP. Add both IPs if you have a Panorama cluster, and then **commit**:

```
# set deviceconfig system Panorama-server <IP1> Panorama-server-2 <IP2>
# commit
```

In **Panorama** | **Managed Collectors**, you can add all your Panorama and M appliances:

1. Enter the Panorama or Log Collector serial number and the IP address. If the Panorama instance is part of a cluster, add the HA peer's IP as `Panorama Server IP 2`.
2. If you add the local Panorama serial number, Panorama will remove all the additional fields (IP address, domain, DNS, and so on) as it already has the details, as in the following screenshot:

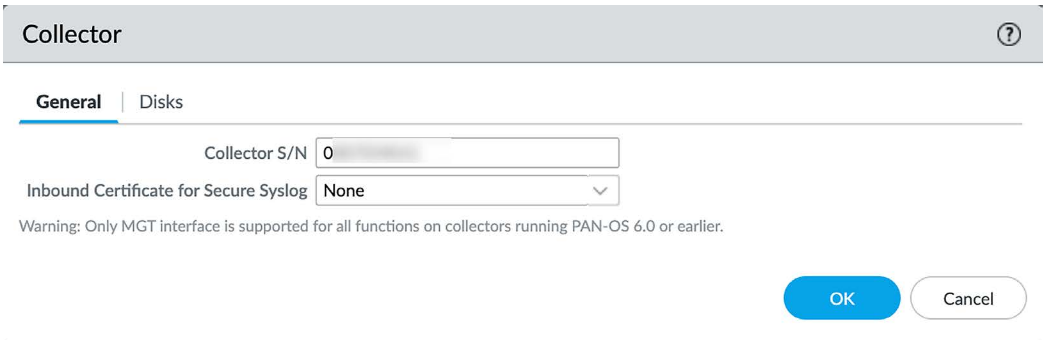


Figure 7.8: Local Panorama Log Collector

3. If you are adding an external Log Collector, the dialog window will look similar to the following screenshot. Fill out the Log Collector details and the management properties that it should be configured with once it connects to Panorama. In the **Authentication** tab, set the admin password:

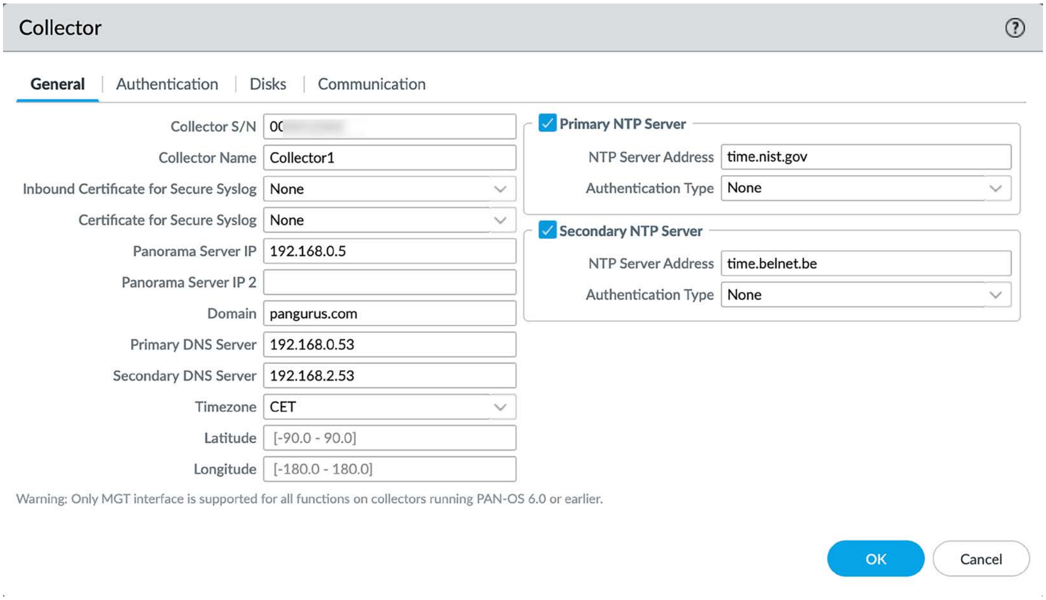


Figure 7.9: External Log Collector

4. You can add the DNS and NTP settings you want the device to use if these have not been configured yet.
5. Click **OK** and then **Commit to Panorama** and **Push to Devices**. This will enable Panorama to retrieve the disk pairs.
6. If you set up **Secure Communication** earlier, set up the **client** side of the Log Collector in the **Communication** tab.
7. Reopen the collector and, on the **Disks** tab, add all available disk pairs, as shown in the following screenshot. Some devices will only have a single disk, while large platforms may have up to 12. Click **OK** and **Commit to Panorama**, followed by **Push to Devices**:

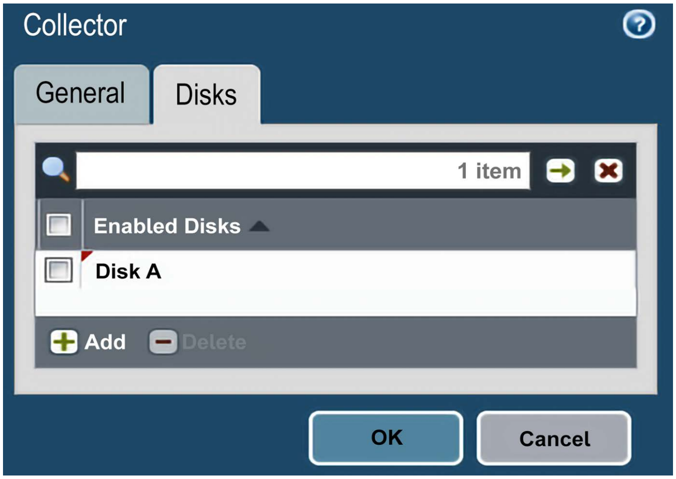


Figure 7.10: Adding disks to the Log Collector

8. Repeat this for all additional Log Collectors. If you add more than one Log Collector, bundle them by creating a new collector group in **Device** | **Collector Groups**.
9. Add the Log Collector(s) to the new group, as shown in the following screenshot:

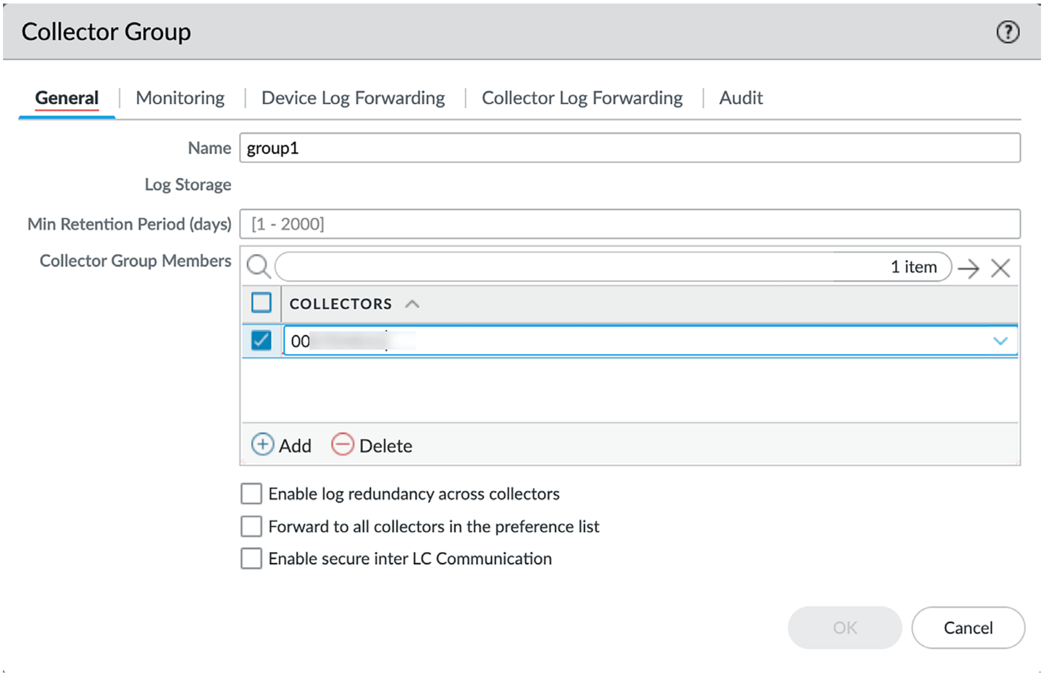


Figure 7.11: Adding Log Collectors to a Log Collector group

The following options are available for log distribution among multiple Log Collectors in a group:

- Without enabling any of the options, the default behavior is for any given firewall to send its logs to the first Log Collector in the preference list, falling back to the next one if the connection to the preferred one is lost.
- Selecting the **Enable log redundancy across collectors** option will instruct all the collectors in a group to synchronize logs among themselves so all collectors have all logs.
- With the **Forward to all collectors in the preference list** option enabled, all firewalls will send their logs to all of the collectors in the preference list.
- You can also select the **Enable secure inter LLC communication** option, which applies secure communications certificates for inter-Log Collector sessions.

10. If you have multiple collectors, you can create preference lists under **Device Log Forwarding**, which allows you to set a list of preferred Log Collectors per firewall so you have more control over where logs are primarily sent. This can be very useful if your firewalls are distributed across several locations, and there's a collector close to each location.
11. In the following screenshot, you'll see that firewalls 1 and 2 will prefer collector 1 and have collector 2 as a backup, while firewalls 3 and 4 will prefer collector 2 and have collector 1 as a backup:

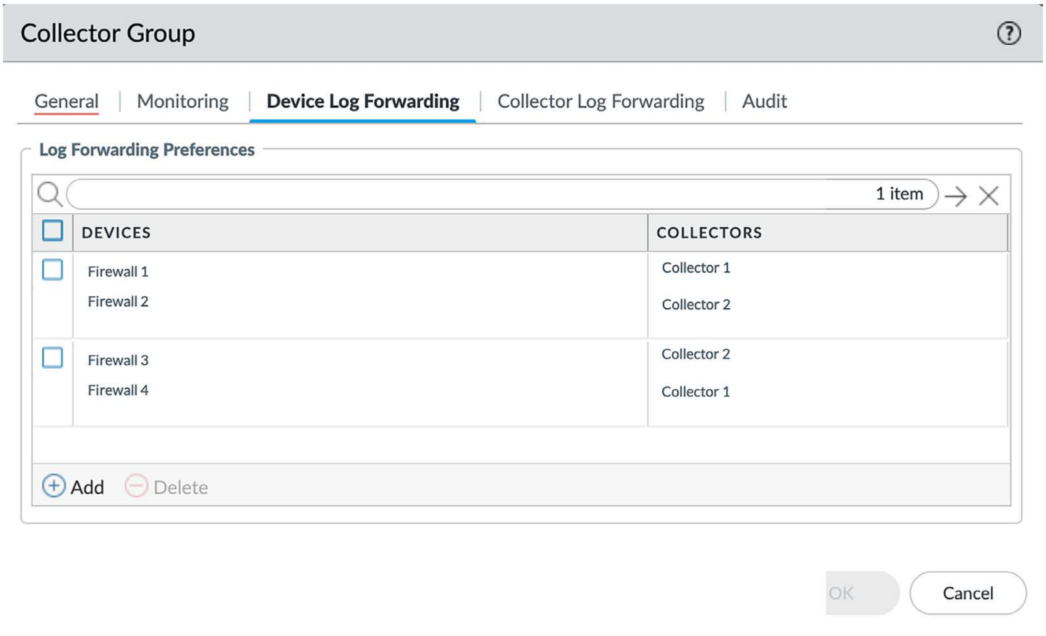


Figure 7.12: Collector group preference list

12. Click **OK** and then **Commit and Push**.

Important note

Panorama uses **Commit to Panorama**, which writes the configuration to Panorama's running config. The **Push to Devices** option will write the configuration, such as templates and policy, to managed devices. **Commit and Push** does both actions in one job. I recommend doing these steps separately. Each method has its perks: doing a commit and a push separately grants more control over which elements are pushed, while **Commit and Push** will only process the changes made in the current session by the current administrator.

You have now learned the differences between the physical and virtual Panorama appliances and can start up Panorama from scratch. You can also choose which logging solution is best suited to your needs.

In the next section, we will learn how to add managed firewalls and create rule bases for groups of firewalls and individual devices.

Device groups

Before we can start managing devices, they first need to be connected to Panorama. On the Panorama side, the device is added by its serial number, and on the firewall side, the Panorama IP address needs to be added. This means the firewall always makes a connection to the Panorama server. Any connections originating from Panorama are backchanneled over the **continuous** connection that a firewall has with its central management.

There are two TCP ports that are used for communication:

- `TCP\3978` is a connection initiated by the firewall and used for all communications between the firewall and Panorama or collectors. Panorama uses this connection to context switch to a firewall or push a configuration over while the firewall sends logs through the connection. Collectors also use it to connect to Panorama. Log collectors communicate with collector group members via `TCP\28270` on PAN-OS 11.0 and earlier. From PAN-OS 11.1 and later, this has changed to ports `TCP\9300` and `TCP\9302`.
- `TCP\28443` is used by managed devices to retrieve content and software updates from Panorama.

The first thing we'll need to do is add the managed devices to Panorama and set up groups to manage them.

Adding managed devices

You can add any firewall that needs to be managed by Panorama by its serial number in **Panorama** | **Managed Devices** | **Summary**. Clicking **Add** will prompt you for a serial number, as shown in the following screenshot. You can add multiple serials at the same time as long as you stick to one per line:

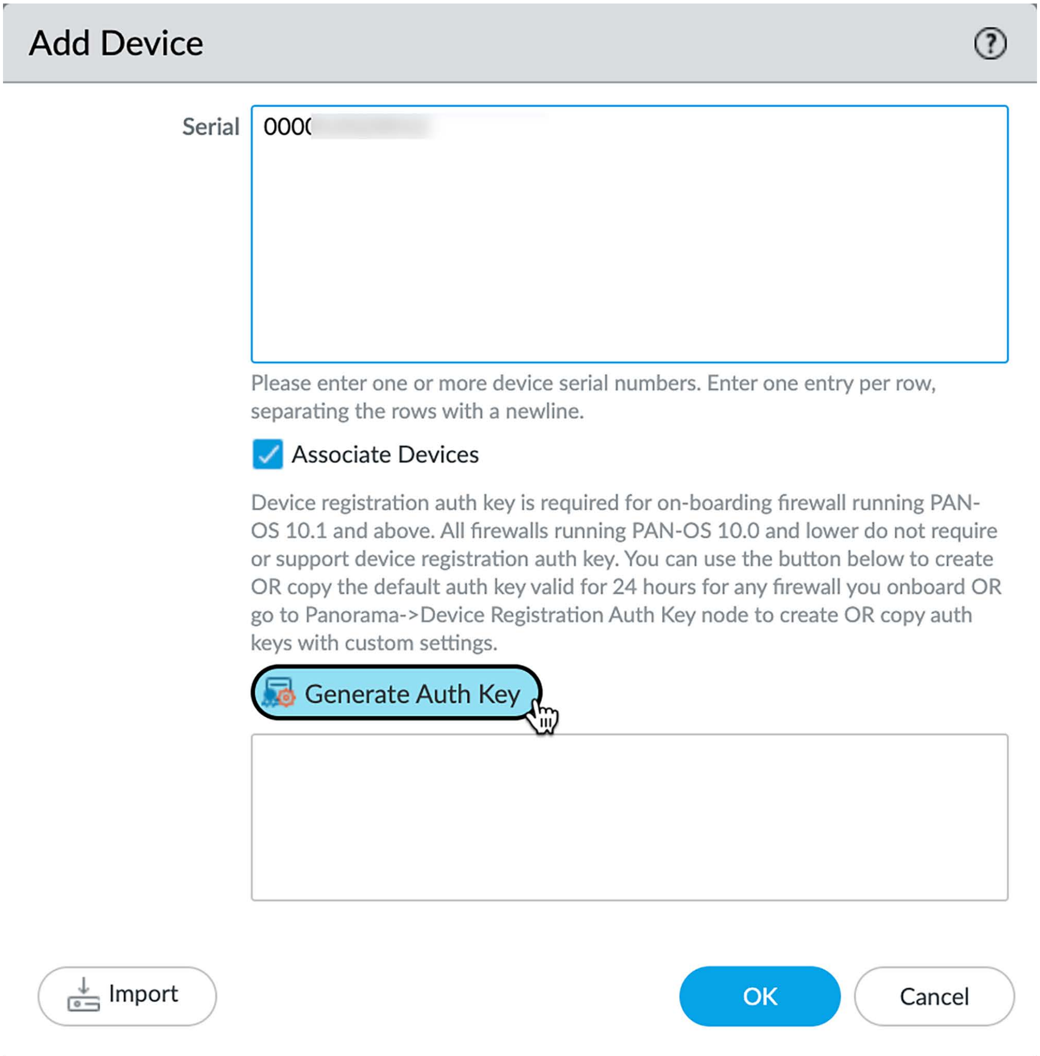


Figure 7.13: Adding new managed devices

The **Generate Auth Key** option will generate a device registration authentication key that is used to establish the first connection from the firewall to Panorama. You can either generate a key for this specific import by clicking **Generate Auth Key** and storing the key to be used on the firewalls, or you can create a key in the **Panorama | Devices Registration Auth Key** section so the key can be used by many different firewalls.

The **Associate Devices** checkbox is used to immediately attach a freshly added firewall to a specific device group and template. If you leave this box checked, when you click **OK** you are taken to the next page where you can assign the new firewall to a device group, template stack, collector group, or collector. Here, you can also enable **Auto Push on 1st Connect**, which automatically pushes out any configurations associated with it when the device connects to Panorama for the first time (be very careful with this last option as it could push an incomplete configuration), and **To SW Version**, which automatically upgrades the firewall to the desired PAN-OS version as soon as it connects to Panorama:

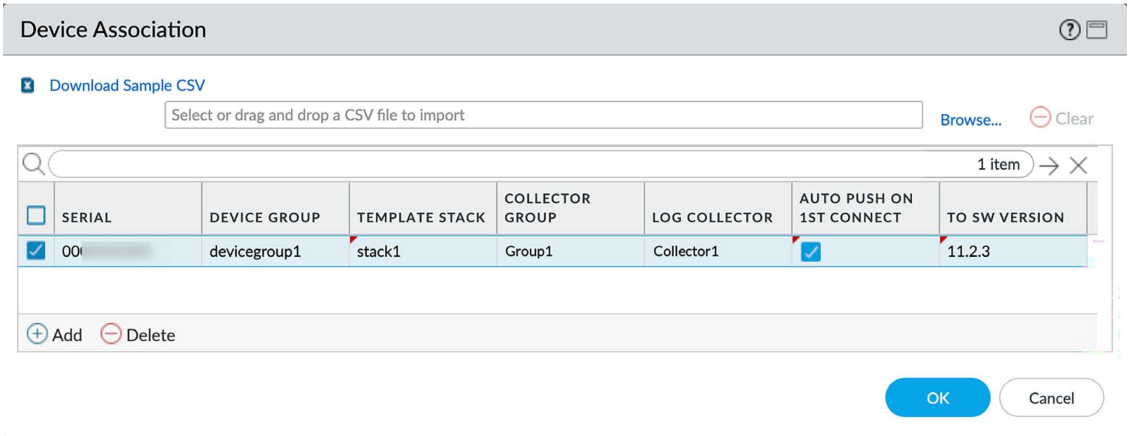


Figure 7.14: Associating new devices

You can also prepare and import a **comma-separated value** (**CSV**) file containing all of the serial numbers, their associated device groups, templates, and suchlike. You can download a sample CSV from the **Device Association** screen and edit it as needed. The CSV should be formatted like this:

```
"serial","device-group","template","collector-group","log-collector","auto-push-on-first-connect","to-sw-version"

"firewall1","devicegroup1","templatestack1","group1","collector1serial;collector2serial",true,"11.2.3"

"firewall2","devicegroup2","templatestack2","group2","collector3serial;collector4serial",true,"10.2.4"
```

For now, since you do not have device groups and templates set up yet, follow these steps:

1. Uncheck the **Associate Devices** checkbox and click **Generate Auth Key**. Copy and save the key separately. Click **OK** to add the device in the inventory.

2. Then, in the individual firewalls, go to **Device** | **Setup** | **Management** | **Panorama Settings** and add the IP to your primary and secondary Panorama instances, as in the following screenshot (if you intend on having a Panorama cluster deployed). Be mindful of whether you use a public or private IP depending on how the firewall connects to Panorama. You will need to paste the auth key you generated in the previous step on Panorama. At the top, you can also set the firewall to be managed by the cloud service, which we will cover in the next chapter:

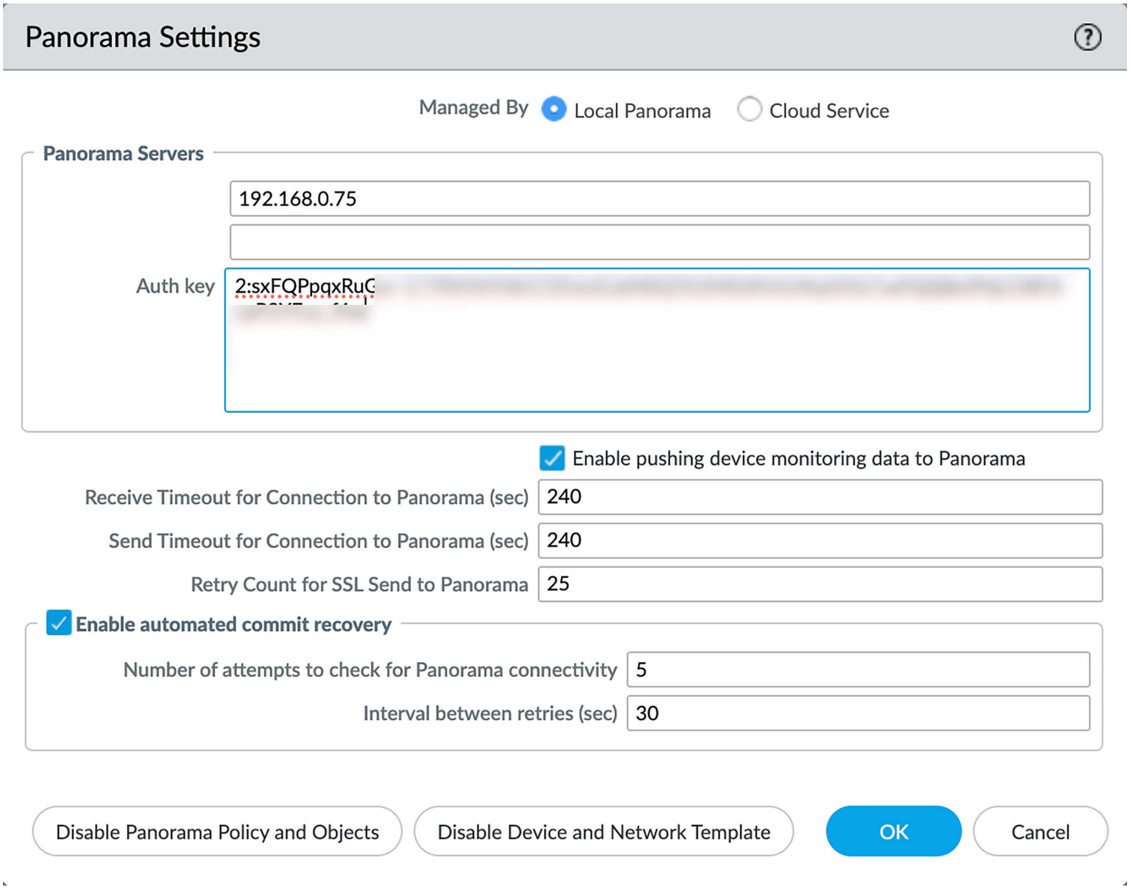


Figure 7.15: Adding the Panorama configuration to the firewall

Your firewall should now connect to Panorama immediately. You can check the status in **Panorama** | **Managed Devices** | **Summary**. If your firewall doesn't connect, make sure you pasted the right auth key and that you used the correct IP for Panorama on the firewall. Lastly, verify that there isn't a firewall blocking the connection from the firewall to Panorama.

As you can see in the following screenshot, if you added managed devices that are in an HA cluster, Panorama can link them if you check the **Group HA Peers** checkbox. Visually, this will not only help you identify HA pairs, but also let you reassociate both peers at the same time, or push updates to both peer members simultaneously:

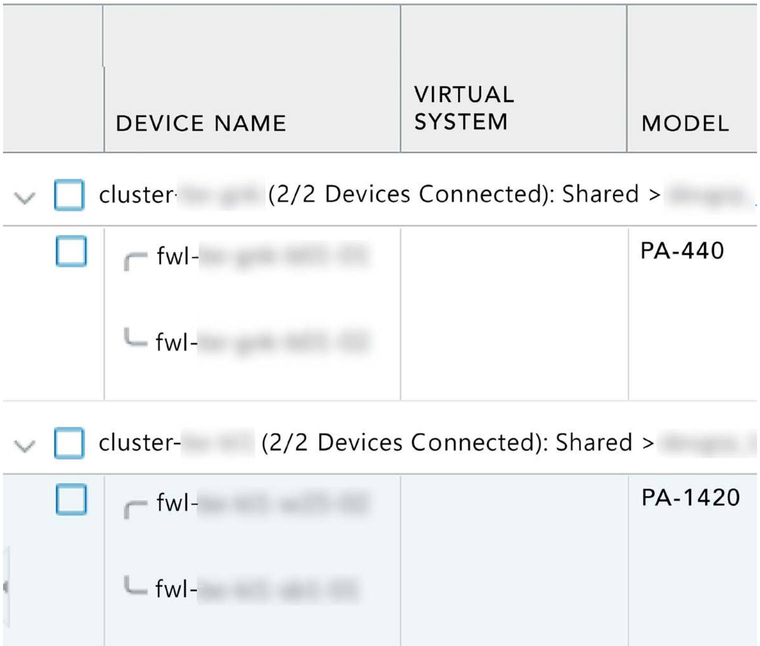


Figure 7.16: Grouping HA peers in managed devices

You can now add managed devices. The next thing we'll need to do is create the device groups.

Important note

Firewalls managed by Panorama must be on the same or a lower PAN-OS than Panorama e.g. firewall on 11.2.5 and Panorama on 11.2.5, or firewall on 10.2.12 and Panorama on 11.2.5.

If the firewall is on a higher version, make sure to upgrade Panorama to the same or higher PAN-OS *before* connecting the firewall.

Preparing device groups

Next, we will create device groups that will contain firewalls according to their characteristics or locations.

> **Important note**
>
> The main purpose of device groups is to bundle rule bases and policy objects so that all members of the same device group are configured to use them while not deploying them to other groups. It's important to keep device groups as simple as possible as there is inheritance to consider, which could overcomplicate your deployment if there is no real need to segregate your firewalls.

When you add a new device group in **Panorama | Device Groups**, you can provide a name and select which devices belong to it, but also, at the bottom, you can select the parent device group and the master device:

- **Parent Device Group** lets you nest device groups where the parent group shares all its objects and rules with the child group
- **User ID Master Device** lets you pick one firewall in the group that will forward all its user ID information (user-to-IP mapping and group memberships), which can then be used in security rules

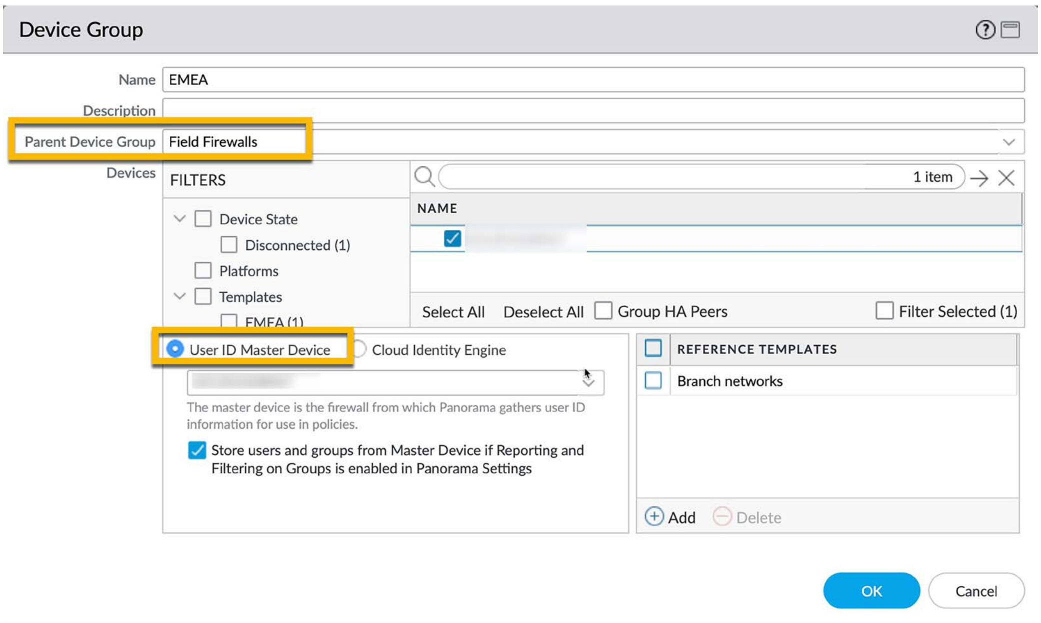


Figure 7.17: Grouping HA peers in managed devices

Important note

Shared is the **grandparent** device group, and any objects created in **Shared** will be made available on all managed devices, regardless of the device group they are in individually.

An example set of device groups can be seen in the following screenshot:

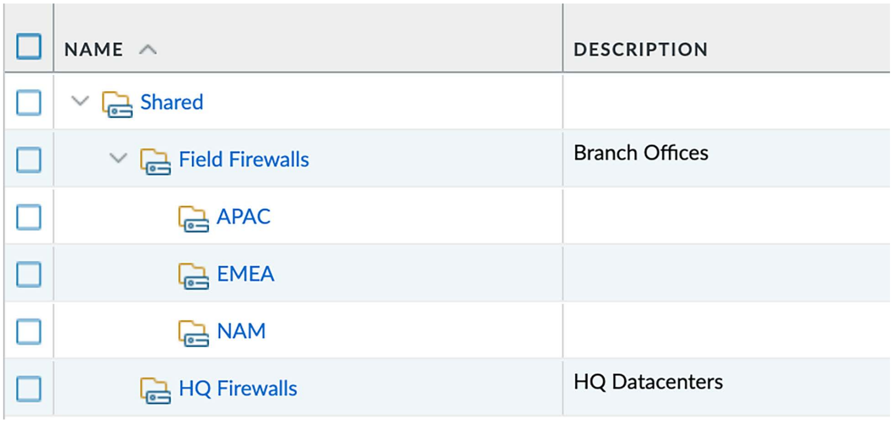


Figure 7.18: Nested device groups

In the preceding scenario, the inheritance of rules and objects would work like this:

- Any objects or rules created in **EMEA** would only be visible to firewalls in the **EMEA** device group.
- Any objects or rules created in the **Field Firewalls** device group would be visible to all firewalls in the **APAC**, **EMEA**, and **NAM** device groups, but not to **HQ Firewalls**.
- Any objects or rules created in the **Shared** device group will be visible to all firewalls, regardless of which device group they are placed in.

This inheritance allows the administrator to set generic rules for management access, security rules for dynamic updates, or access to Panorama, Strata Logging Service, or Log Collectors at the **Shared** level. A rule only needs to be created once to apply it to all firewalls. More localized configuration can then be added to a child device group, like outbound VPN rules for **Field Firewalls** and inbound rules for services hosted at **HQ Firewalls**, each time ensuring one set of rules is pushed out to all the members, but not to firewalls in a different branch.

You have now learned how to add new managed devices and place them in device groups. In the next section, we will learn how to create policies for your device groups.

Creating policies and objects

The goal of device group rules and objects is to manage everything from a central location and, where possible, end up with no local configuration on the firewalls.

While creating objects and rules, you always need to be mindful of the device group you are in while you create new objects. As the following screenshot illustrates, I am about to create a new address object while I am in the **EMEA** device group. If I do not check **Shared**, this object will only be usable by managed devices in the **EMEA** device group:

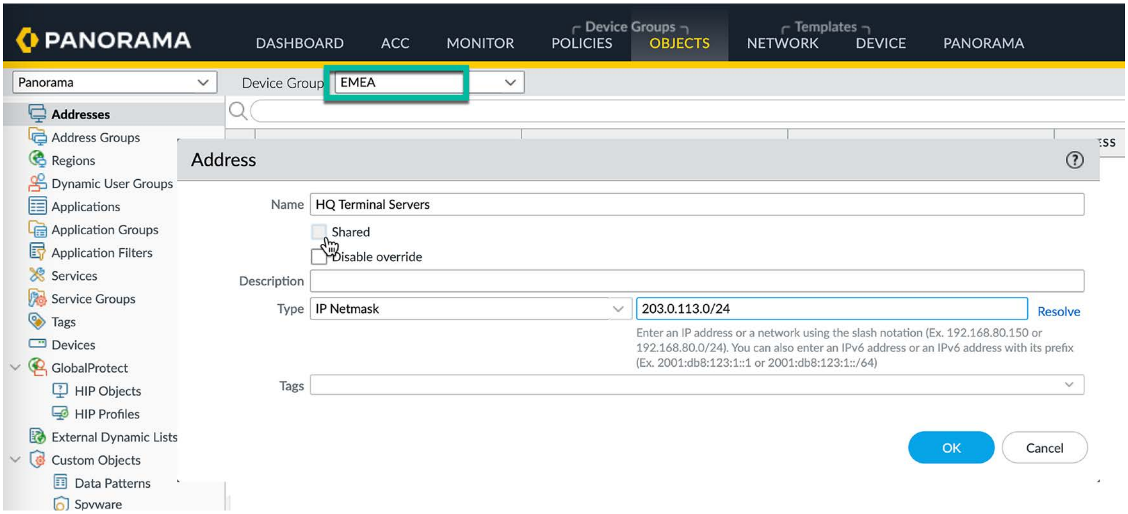


Figure 7.19: Device group context

This is because objects that were created in a specific device group cannot be set to **Shared** afterward. They can, however, be *moved* or *cloned* to the **Shared** device group.

Important note

If you create an object in **Shared** and create another object with the same name in a nested device group, the object in the nested device group will override the object in **Shared**.

For example, when you have **object1** in **Shared** with an IP address of 10.0.0.1 and **object1** in the EMEA device group with an IP address of 10.0.10.1, firewalls in the **APAC** and **NAM** device groups will know that this object's IP address is 10.0.0.1, while firewalls in the **EMEA** device group will have its IP address as 10.0.10.1. This allows you to create security rules at the shared level that still have different IP addresses in firewalls of different device groups.

Most objects that are pushed from Panorama can be overridden by a local firewall admin. Address objects that are not shared can be set to **Disable Override** so that local admins are not able to change them.

Especially when using nested device groups, rule bases will be built-in layers. A unique concept in Panorama is the use of **pre- and post rules.** These are placed before and after local rules on the device. This enables administrators to set rules that override locally configured rules or make sure there are clean-up rules in place after local rules. The order of device groups' pre- and post rules is illustrated in the following diagram. Since rule bases are always evaluated from top to bottom, the **Shared** pre-rules will always be hit first, and the **Shared** post rules last, just before the default rules:

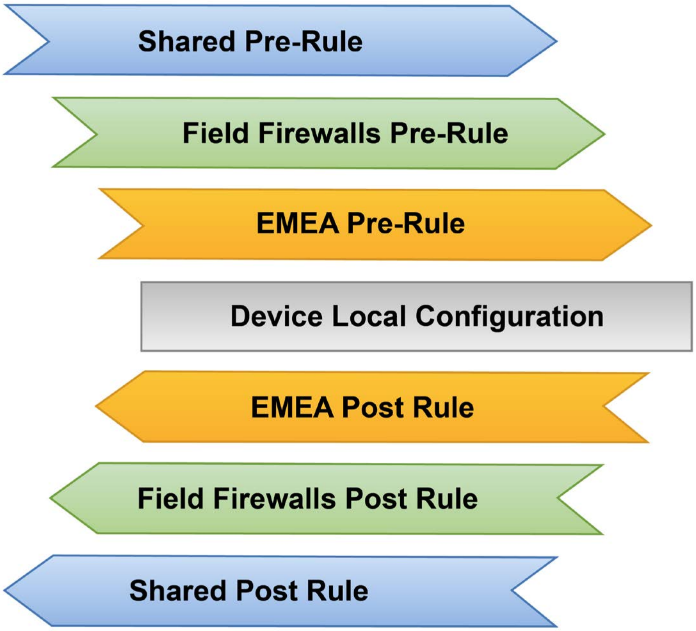


Figure 7.20: The order of pre- and post rules

Important note

In *Figure 7.20*, the **EMEA** device group sees rule 3 as a native rule, while rules 2 and 1 are created in the device group it is nested in. The **Field Firewalls** device group sees rule 2 as a native rule, while rule 1 belongs to the device group it is nested in. For the **Field Firewalls** device group, rule 3 is not visible because it belongs to a child device group and does not apply to any devices in Field Firewalls. The **Shared** device group only lists its own native rules.

Depending on the device context that you are currently in, some rules will be invisible, visible and editable, or visible and uneditable. Rules with an orange background belong to a higher-up device group. On the firewall, all Panorama rules will have an orange background and cannot be edited unless the local admin explicitly overrides it:

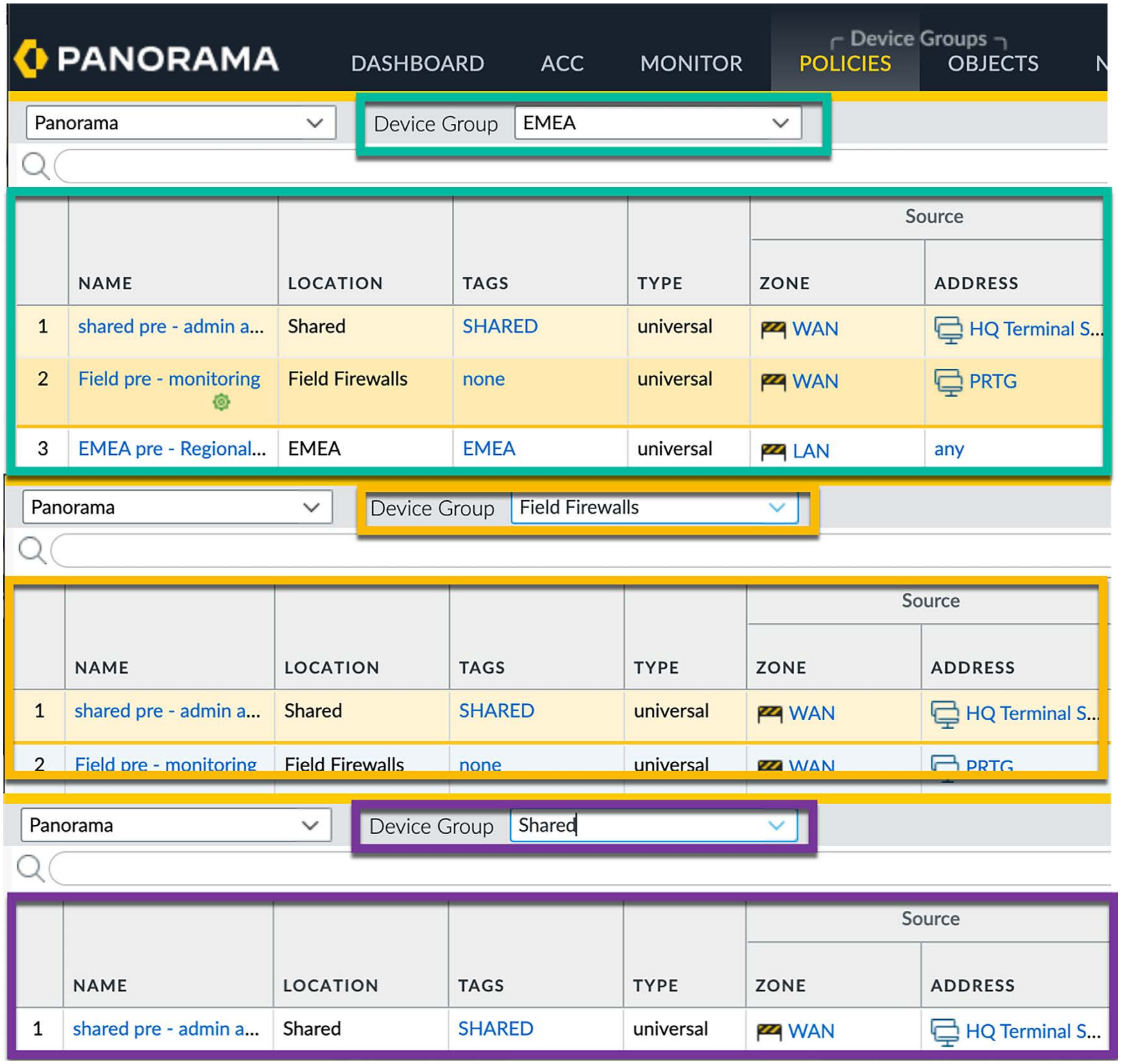


Figure 7.21: Security rules in the device group context

On top of being able to control which rules are deployed to a certain group of firewalls, rules have an additional tab, **Targets**, which can be used to control, even more specifically, which firewalls a rule is applied to. This can help prevent the need for another nested device group if there are very few exceptions to the norm (for example, one firewall may need to be configured to allow access to a legacy server).

When you disable a local rule on a firewall, or in Panorama, the rule will appear grayed out. When a rule is disabled in Panorama, it will no longer appear on the local firewall rulebase.

Now that you can create device groups, there are a couple of things you should do, or at least know about, to make your life easier.

Important things to know when creating objects in device groups

When you first create rules, the zones may not be known by Panorama yet. This is because the Device Groups are not directly linked to the templates where zones are configured. There are a few ways to solve this issue:

- When you create a rule, you can simply type the zone name as it is known on the firewall, or as you set it in a template, and then click **OK**. After your first time typing in a zone, the device group will learn the zone name and it will appear in the dropdown thereafter.
- If your zones are configured in a template already, you can set this template as a **Reference Template** in the device group, as shown in the following screenshot. This way, the device group will be loaded with all the information available in the template.

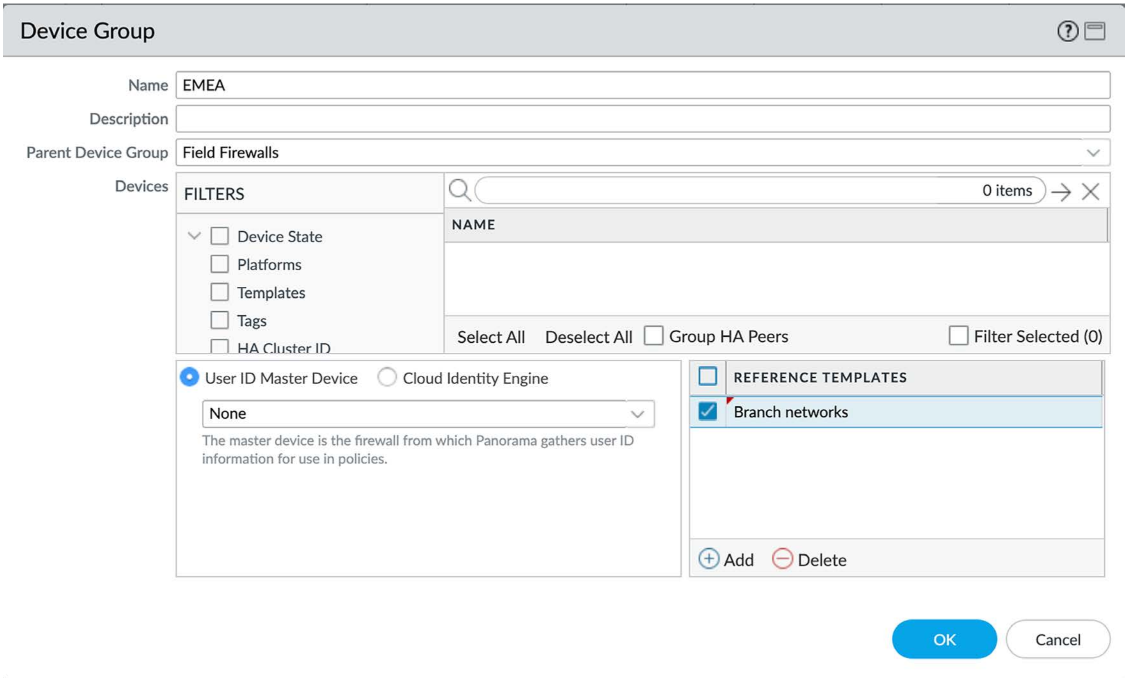


Figure 7.22: Reference templates in a device group

Rules can be **cloned** to other device groups or **moved** to a different device group.

Rules cannot have the same name as a rule in a nested device group, as this will cause a conflict during commit, but can share a name with a rule in a device group in the same tier (different branch). For example, there can't be a rule named **rule1** in both **Shared** and **EMEA**, but you can have a rule named **rule1** in EMEA and **APAC**. To prevent accidental naming conflicts, cloned rules will be postfixed with a number, like -1.

Setting up default attributes

To simplify rolling out your security rules, a Log Forwarding profile and a security profile group can be created that will automatically be added to each new rule you create.

In **Objects** | **Log Forwarding**, create a **Log Forwarding** profile and call it `default`. Check the **Shared** box and add all the relevant log types that should be forwarded to Panorama by default (such as traffic, threats, URLs, and WildFire). This will ensure that every security policy you create going forward has the log forwarding profile set and sends logs to Panorama. Your profile should look as shown in the following screenshot:

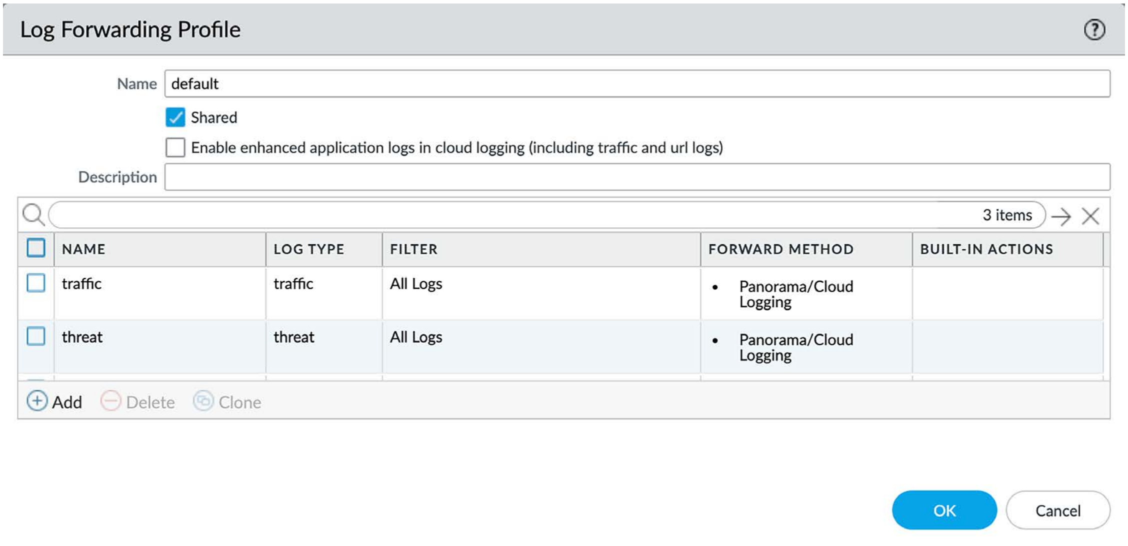


Figure 7.23: Log Forwarding profile

In **Objects** | **Security Profiles**, create the security profiles, and in **Objects** | **Security Profile**

Groups, create a new group, which you should call `default`. Set it to **Shared** and add all the security profiles you just created. This will ensure that every new security rule created automatically comes loaded with security profiles:

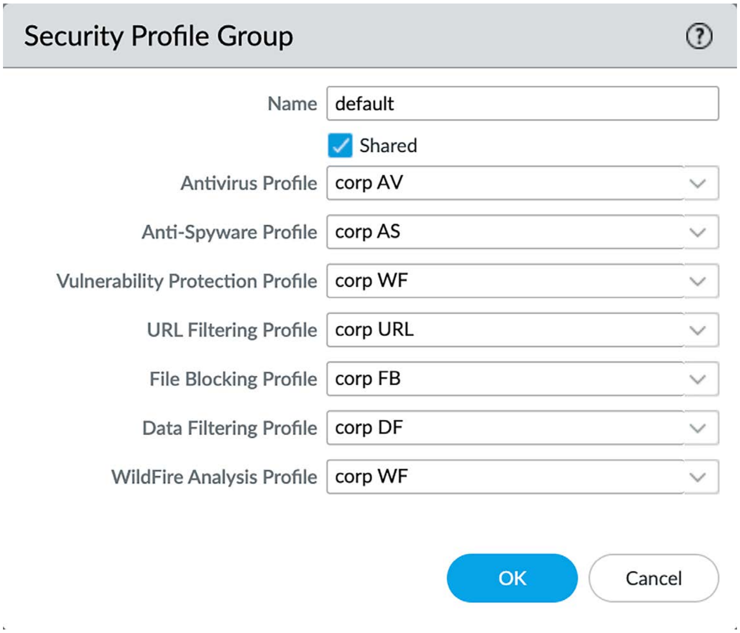


Figure 7.24: Security Profile Group

Important note

The intent of the preceding two **default** profiles is to create a baseline profile that will fit most cases, which is why they should be set to **Shared**. **Tuned** profiles can be created per device group if needed. Just like objects, security profiles have a hierarchy so any Security Profile placed in a nested device group that has the same name as the Shared profile will override the profile for this device group.

For example, placing a corporate URL security profile in the EMEA device group allows you to tune the URL filtering policy for EMEA firewalls without needing to change the Security Profile in all security rules.

You can now create device groups, and you understand what advantages and disadvantages are associated with nesting them. You can add managed devices and have learned how to create pre- and post rules. In the next section, we will learn about templates and template stacks, as well as how to aggregate common device configurations.

Setting up templates and template stacks

Templates are a great way to deploy a common device configuration across your managed devices. A template is a profile where you can set parameters in the **Network** and **Device** sections of the configuration for your managed devices. For example, you can set the same DNS servers, NTP servers, and domain name for all your firewalls.

To allow more flexibility, you create **template stacks** for each firewall or firewall cluster and add templates to the stack as needed. A stack is a container that can hold several template profiles to combine their configurations into a tailored config bundle for a specific (set of) firewall(s).

Considering the previous example of three regions and an **HQ** location, we could create four template stacks – one for each firewall – in **Panorama** | **Templates**, and then add the associated firewalls to each **template stack.** To do this, follow these steps:

1. Create templates that contain broad configuration parameters and will be used to fulfill a certain task, for example:

    - You can create an **admin template** containing all the security team admin accounts and authentication profiles, a standardized log-in banner, password complexity settings, and so on.
    - You can create a **network template** containing all the zones and basic interface configuration, as well as zone protection profiles.
    - You can create a **management template** containing the management interface DNS and NTP settings and update schedules.

    The possibilities are endless (until you reach 1,024 templates, which is the current limit). You can only have up to 8 templates in a stack, so be mindful not to go overboard.

2. Add template stacks as needed, usually one per firewall or firewall cluster, but these could also be deployed per region or by purpose. In each stack, you must add the firewalls that belong to the stack and the templates that will be added to the stack. Note the following:

    - Configuration made in the template stack has priority over added templates, but as a rule of thumb, you should set all configurations in a template
    - Templates assign priority from top to bottom as they are added to the stack. A setting in the top template will overrule the same setting in consecutive templates

The following diagram paints a simplified picture of the relationship between template stacks and templates. The **Admin** template is applied to all the stacks so all firewalls will receive this configuration. The **Network** template is only added to the **HQ** and EMEA stacks, so only firewalls in these two stacks will receive the configuration. The **Management** template is applied to EMEA, APAC, and **NAM**, but not **HQ**, so all firewalls except those in **HQ** will receive this configuration:

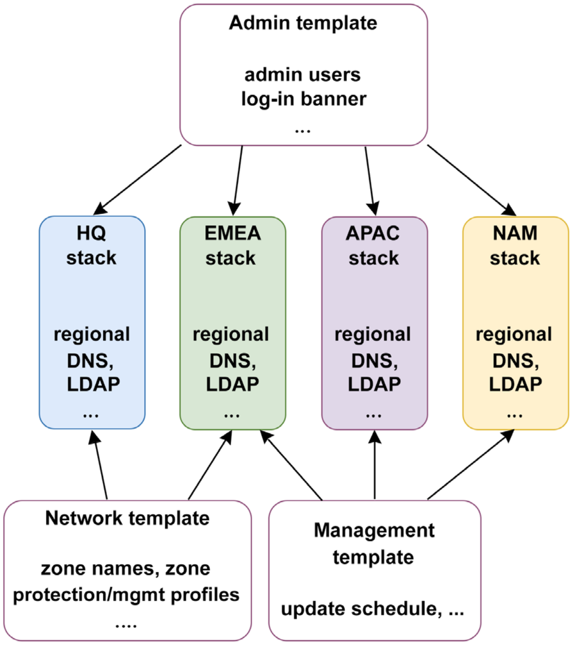


Figure 7.25: Templates and template stacks

As the following screenshot shows, all you need to do to edit a template is select it from the **Template** dropdown in the section of the configuration where you want to add the configuration. An important caveat is that Panorama is not fully aware of some settings that could be active on your firewall. As shown by the **Mode** dropdown, the default assumption is that the firewall is a multi-VSYS system, it is running in **Normal** (not CIFS or CC) mode, and that VPN is enabled.

This will cause Panorama to show options that might not be available on the firewall (such as VSYSx on a configuration that is intended for a single VSYS system, weak encryption options on a FIPS-enabled system, and so on). You can either set these options from the dropdown to remove unavailable config options or keep track of specific device limitations yourself:

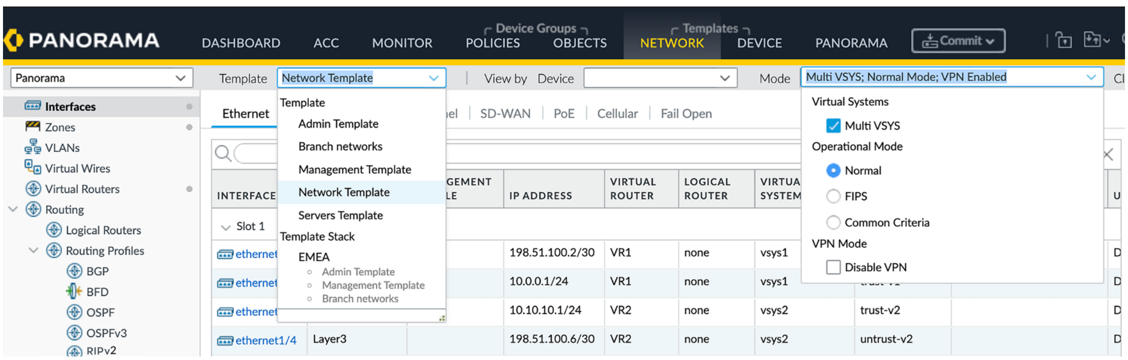


Figure 7.26: Template selection and configuration mode

Important note

Keep in mind that templates are independent and have no inherent relationship to each other. Objects created in one template are not accessible in another. Objects can be overridden in the stack by arranging them from top to bottom in the desired order of precedence. For example, a virtual router in one template cannot reference interfaces in another template. Combine them into one template and rely on variables to customize the values.

You have now learned how to plan out template stacks and how to leverage separate templates to ensure all your firewalls get the configuration they need while simplifying the configuration repository. In the next section, we are going to learn how we can use variables to customize and tailor shared configuration to individual firewalls.

Leveraging variables to customize common configurations

If you want to standardize a configuration, such as using the same interfaces and zones across all your firewalls, you may run into an issue where you can't configure the IP addresses in a template that is shared with multiple stacks or individual firewalls in a stack, as each firewall will need local IP addressing. Variables can be used in these cases, so shared configuration becomes customizable per firewall or per stack. There are many places where variables can be used, such as data plane and management interfaces, **High Availability** (HA) configuration, and virtual routers (BGP peers, for example). A variable can be created by clicking the **Variable** option in the dropdown while creating an object. A variable's name should always start with the $ sign followed by an easily identifiable description:

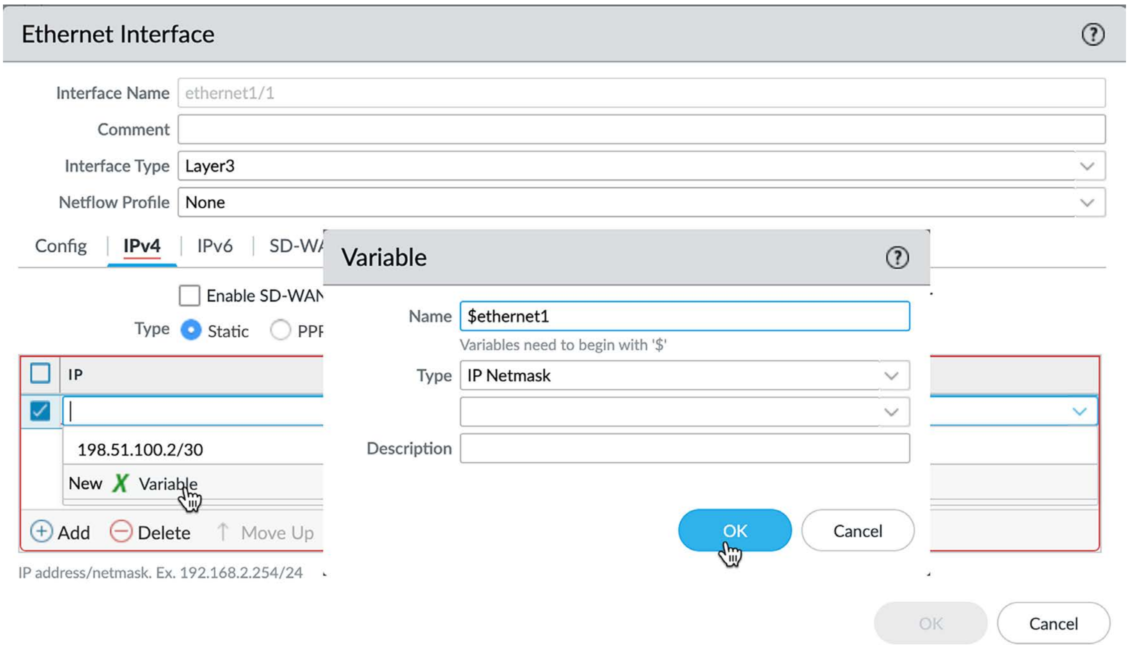


Figure 7.27: Creating a variable

The variables are automatically added in the template you first created them in, but they can be recreated on each template stack to have a unique configuration for each stack. Navigate to **Panorama** | **Templates** and click **Manage** to add and configure the templates on the stack:

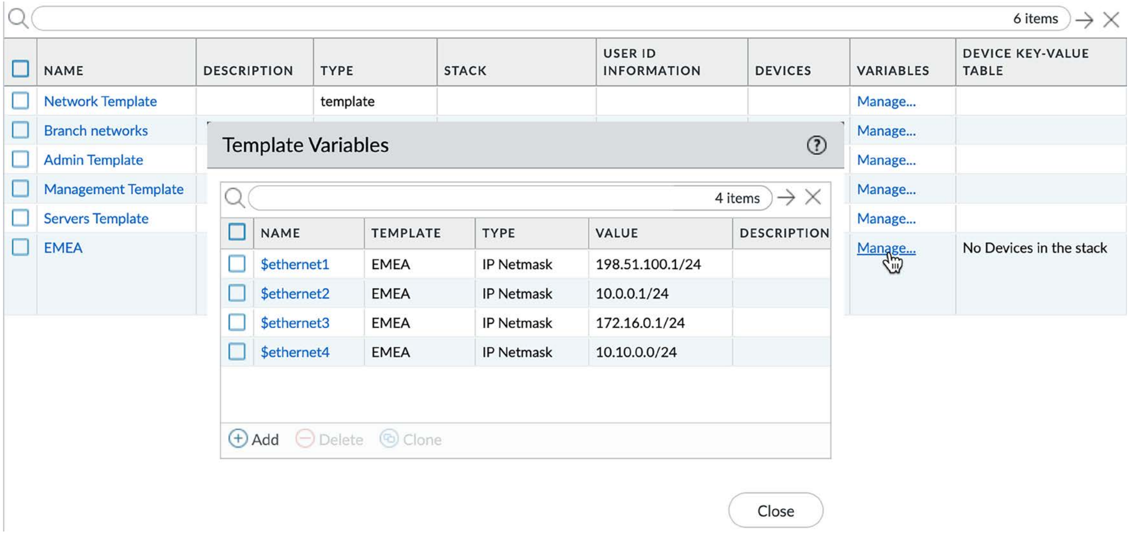


Figure 7.28: Creating a variable on a stack

Following the previous two screenshots, we can now set the interface configuration in a template so all firewalls receive the same configuration, zones, and so on, but the IP associated at the firewall level is set by the variable.

If you need to fine-tune variables even further to the individual firewall (if, for example, there are multiple firewalls in the same stack), you can also override variables at the device level. Navigate to **Panorama** | **Managed Devices** | **Summary** and scroll to the right until you see the **Variables** column. Click **Create** and select **No** on the dialog.

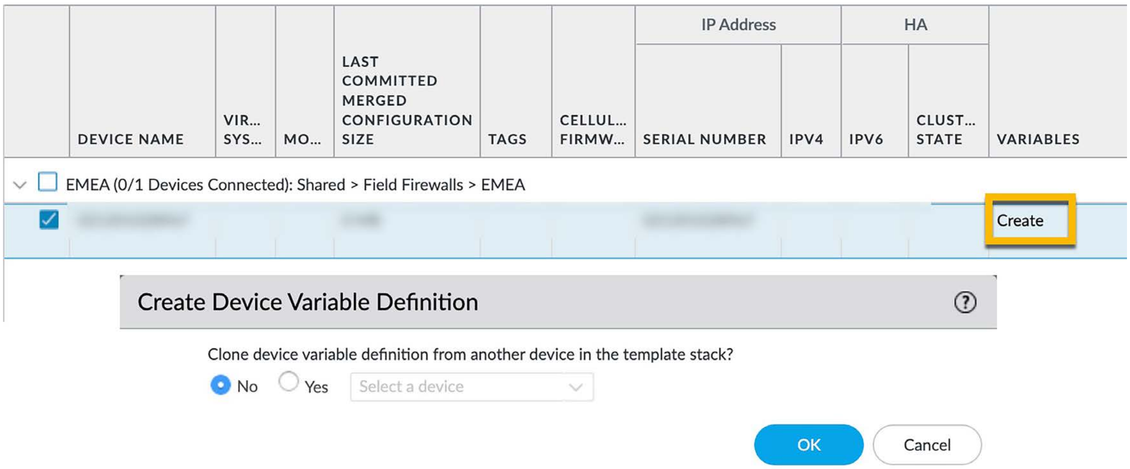


Figure 7.29: Creating variables at the device level

Then, select the variable you want to change and click **Override**. You can now set the variable at the device level:

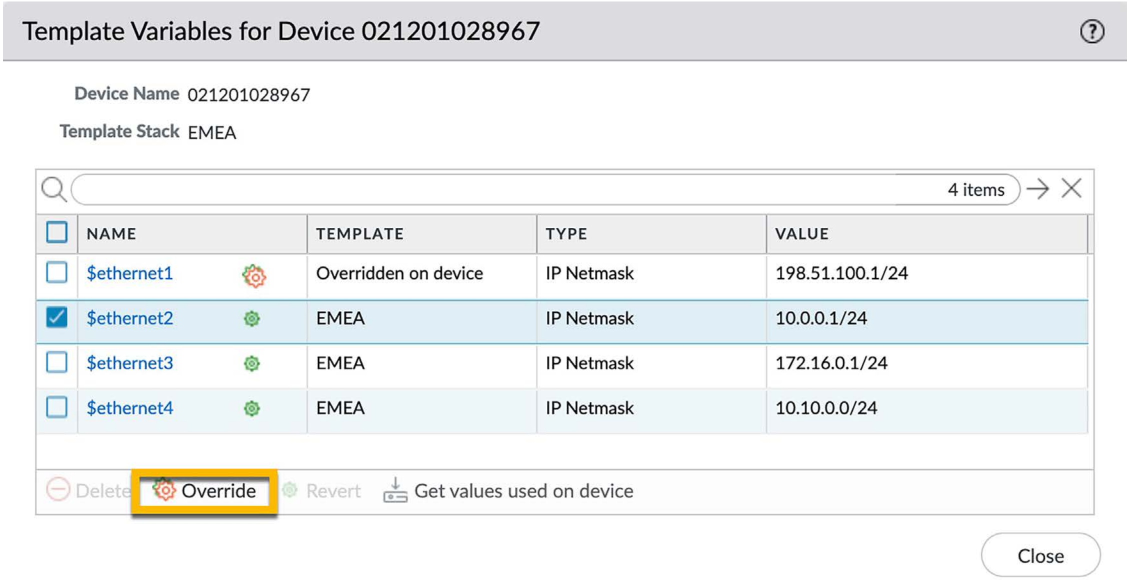


Figure 7.30: Overriding variables at the device level

Panorama management

In this section, we will learn about the simple management tasks that you would normally need to perform on each firewall individually (which can be very time-consuming) and how they can be centralized and made much easier to manage from Panorama. The first task of an administrator is to make sure all firewalls have up-to-date signatures and content packages.

In the following subsections we'll learn about managing system tasks like dynamic updates from Panorama, how we can onboard previously unmanaged systems and how we can set up High Availability for Panorama.

Device deployment

Content updates can be managed through Panorama in two ways. A template can be created that sets a local update schedule on each firewall, which will require each firewall to connect to the update server individually and collect and install updates. Another method is setting an update schedule on Panorama and pushing out updates to all devices. The second method gives you a little more control over what is being pushed to the managed firewalls and when, but does increase bandwidth usage on the Panorama site or cloud provider.

To schedule updates that are sent out from Panorama, you can do the following:

1. Go to **Panorama** | **Device Deployment** | **Dynamic Updates** and click on **Schedules** at the bottom.

2. Create a schedule for **Apps & Threats** and do the following:
   - Set **Recurrence** to `Every-30-Minutes` at `24` minutes past the half-hour to prevent conflicts with other update schedules.
   - Action **Download and Install.**
   - Select the devices that should receive these updates
   - **Threshold** is intended to hold off on installing a package and rechecking after the specified amount of time in case there is a recall. Set this to `6` hours or more.
   - **Application threshold** waits to activate new applications for a specified amount of time so that the security team can review the possible impact on security policies. Leaving this option blank will simply go ahead with the installation of new App-IDs.
3. If you have firewalls without a Threat Prevention license, create an app-only schedule with a recurrence of `Daily at 22:05`.
4. Create a schedule for **Antivirus**:
   - Set **Recurrence** to `hourly`.
   - Set **Minutes Past Hour** to a random number so that there is no conflict with the **Apps & Threats** updates.
   - Action **Download and Install.**
   - Select the devices that need to receive these updates.
   - Set **Threshold** to `3` hours.
5. Create a schedule for WildFire:
   - Set **Recurrence** to every minute (or `15` if bandwidth is an issue).
   - Action **Download and Install.**
   - Select the devices that will receive WildFire updates.
6. Create duplicate schedules in case there are firewalls in vastly different time zones.
7. In **Panorama** | **Dynamic Updates**, also set schedules for Panorama's own updates that mimic the preceding recurrence, but to a different minute past the hour to avoid conflicts.

A URL database update is only required if the target firewalls are not capable of performing cloud category lookups, as a URL database update (seed file) will also purge and replace the local URL cache on the firewall.

Upgrading firewall OSs can be done from Panorama as well, through **Panorama** | **Device Deployment** | **Software.** After clicking on **Check Now**, every available PAN-OS version will be listed next to every available platform:

1. Download the PAN-OS version of the platform you want to upgrade.
2. Click **Install.** Panorama will then display all the matching managed devices that are eligible to be upgraded.

3. Select the devices that need to be upgraded and do the following:
   - Click **OK** to install the software **without rebooting** the target firewall.
   - Select **Upload only to device** to upload but not install the software image, and then click **OK**.
   - Select **Reboot device after install** to also reboot the firewall once the installation completes, and then click **OK**.

> **Important note**
>
> While upgrading a firewall is fairly straightforward, it is recommended and encouraged to plan accordingly and have someone standing by at the site of the upgrade in case something does go awry.

Plugins and GlobalProtect client packages can be distributed in the same way.

In **Panorama** | **Device Deployment** | **Licenses**, you can review all the licenses deployed across all your devices and their expiration dates.

You can now manage all aspects of provisioning your firewalls with content updates and upgrading them from Panorama. Next, we will review how to import an existing firewall into Panorama.

Migrating unmanaged to managed devices

Unmanaged devices that have already been fully configured may need to be integrated into Panorama, which can be challenging. Instead of trying to gradually replace local configuration with Panorama templates and device group configuration, a firewall can be imported and its configuration converted into a template and a device group per VSYS:

1. Add the firewall as a managed device (do *not* associate the device with device groups or template stacks at this time) and select **Commit to Panorama**.
2. Add the Panorama IP in the firewall in **Device | Setup | Panorama Settings** and click **Commit**. Once the commit is complete, verify whether the device is connected to Panorama under **Panorama** | **Managed Devices** | **Summary**.
3. In **Panorama** | **Setup** | **Operations**, click on **Import device configuration to Panorama**.
4. In the dialog, you can select the freshly added managed device:
   - You should name the template so that it is easily identifiable.
   - The default name for the device group is the firewall name. If there are multiple VSYS, the device name will be the VSYS name, so add a prefix to easily identify the firewall in the device groups.
   - By default, all of the firewall's shared objects are imported as **Shared** objects for Panorama. If you do not want other firewalls to receive these objects, uncheck the option and all the objects will be imported as part of the new device group.

- Select whether rules need to be imported into the pre- or postrule base.

5. The import also creates a new stack and a template that contains all the devices' configurations. If you do not want to add the Panorama shared templates to the newly created stack yet, you can add them later once you've verified that the device has been successfully integrated.
6. You can now use the device group or template context switches to review whether the configuration has been imported properly. To do this, click on **Commit to Panorama**. Do *not* select **Commit and Push**. In **Panorama | Setup | Operations**, you can now click **Export or push device config bundle**. Select the appropriate device serial number and click **OK** to send the bundle to the device. This operation will push all the configuration in the device group and template stack back to the managed device and replace the local config with the Panorama-managed config. Sending the bundle is preferred as it sends everything in one swift operation. A regular **Push to Device** may cause some dependency issues requiring multiple commits and forced template values.

Important note

Replacing the device's local configuration with the Panorama template configuration *will* cause a brief connectivity interruption as the entire configuration is replaced, which will cause some services to briefly restart loading the newly received configuration. Account for this possibility and plan accordingly if the target environment is sensitive to connectivity issues.

If you import two cluster members following the procedure we just discussed, you will end up with two separate template stacks and device group entries. One member can be moved into the other member's stack and device group to unify the cluster configuration.

Important note

When adding two members of a cluster to the same template stack, ensure that all the cluster configurations, hostnames, and management interface configurations are either removed from the template and configured locally, or variables are used to ensure each peer has its unique configuration maintained.

You have now learned how to manage and maintain devices in your Panorama instance. In the next section, we will learn how to set Panorama in HA mode so that it becomes more resilient to failure.

Panorama HA

Compared to the firewall HA, Panorama HA is much less complicated. The only conditions are the following:

- Both HA members must have the same device type, version, and mode (for example, both are M-600 and in **admin-only** mode).
- They should be on the same PAN-OS and have the same set of licenses for smooth operation.

To enable HA, follow these steps:

1. Go to **Panorama** | **High Availability** and do the following:
    - Enable **HA**
    - Set **Peer IP**
    - Enable **Encryption**
2. In **Election settings**, do the following:
    - Set the priority for this Panorama instance to **Primary**. The primary Panorama instance will be responsible for pushing configuration to firewalls, but both members can be used for configuration, log queries, and reports.
    - **Preemptive** should be enabled in most cases so that the primary member always returns to its active status.
3. Repeat the preceding steps for the peer, replacing **Peer IP** with the first Panorama instance and setting the priority to **Secondary.**
4. On both peers, go to **Panorama** | **Certificate Management** | **Certificates** and select **Export HA key** for each peer and **Import HA key** to import the key onto the other peer.

Unlike firewalls, however, Panorama sticks to the primary and secondary roles throughout failures. **passive-secondary** will become **active-secondary** if the primary Panorama instance experiences an outage. There are two important considerations to keep in mind:

- The device assigned as **Secondary** cannot be used to deploy software or manage licenses.
- A device in the **Passive** state cannot manage a shared policy or deploy software and manage licenses.

In other words, the secondary Panorama should not be used for most configuration tasks unless the primary Panorama is unavailable.

Panorama uses `TCP/28` for encrypted connections between MGT interfaces. If you do not enable **encryption**, connections are set up on `TCP/28769` and `TCP/28260`.

In the last section of this chapter, we're going to take a look at a couple of bits of information to keep in your pocket while working with Panorama.

Replacing one device with another

If a device ever needs to be *replaced*, whether due to a defect followed by a **Return Merchandise Authorization** (**RMA**) or an upgrade, rather than manually adding it to all the device groups and stacks, a simple `replace` command is available to switch the serial number of the old device with the new one so that the configuration is immediately set accordingly:

1. Set up the replacement device to a minimum base configuration by registering it, moving the appropriate licenses over via the RMA process, and upgrading it to the appropriate PAN-OS and content version. Configure the management interface and add Panorama IP addresses.

2. Run the replace command from the Panorama CLI to replace the old serial number with the new device's serial:

```
replace device old xxxxxxxx new yyyyyyyyyy
```

3. Push the config bundle via **Panorama** | **Setup** | **Operations** by clicking **Export or push device config bundle.**

Tips and tricks

Committing a configuration on Panorama requires extra steps before it becomes a running configuration on a firewall. In the top-right corner of the web interface, you have several options. **Configuration** | **Save** saves your candidate, while **Configuration** | **Revert** undoes any configuration changes since your last save or commit.

Commit to Panorama activates your changes as the running configuration for Panorama, but this configuration still needs to be sent out to the firewalls.

Push to devices sends the running configuration on Panorama out to the firewalls. If you click **Edit Selection**, you will open the dialog window shown in the following screenshot. From here, you can click **Preview Changes** to compare the Panorama running config to the firewall running config to see which configuration elements will be changed, added, or deleted.

Merge with Device Candidate Config is enabled by default, as shown in the following screenshot. If a local admin is making changes to the firewall, they may not be ready to have their changes committed, so you can disable this option to prevent mishaps. If you don't want to include template configuration, you can either disable the option at the bottom or uncheck all the devices on the **Templates** tab. **Force Template Values** can be used to overwrite any local configuration with template values:

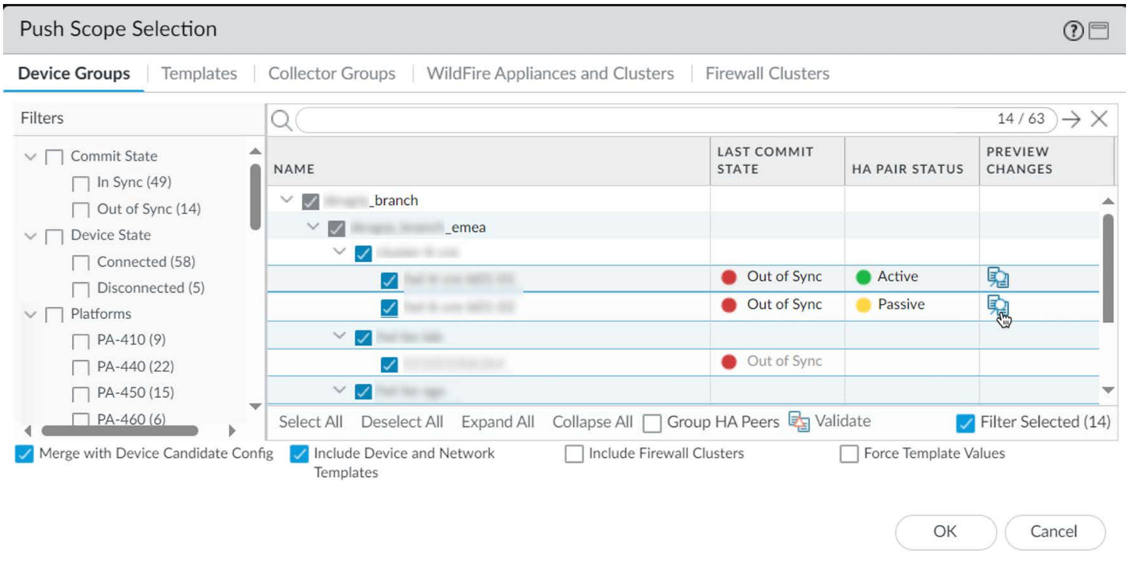


Figure 7.31: Edit Selection in Push Scope Selection

Commit and Push does both of the preceding actions (commit to Panorama and push to devices) in one go. This is a great option if you only made a small change and want to push it out immediately.

If you want to check what the state of the local firewall is (what configuration is in the candidate and the locally running config), you can use the device context switch to connect to the local web interface of your target device.

This connection is backchanneled over the connection the firewall makes to Panorama. This can also be helpful if you lose direct access to a remote firewall that still has an active link to Panorama. As shown in the following screenshot, simply click on the **Context** dropdown and select the device you want to connect to:

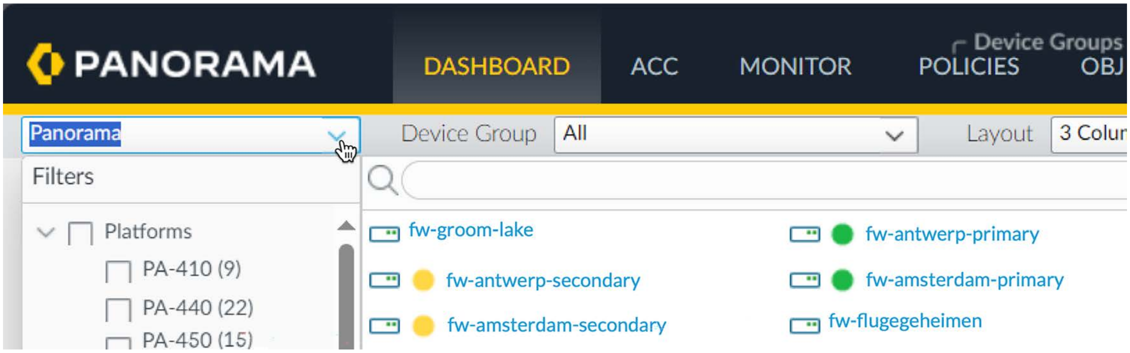


Figure 7.32: The device context switch

If, at one point, you do need to temporarily **override** a configuration parameter pushed by Panorama, you can connect to the firewall and, as shown in the following screenshot, select the object that has a template value and click on **Override** at the bottom of the page:

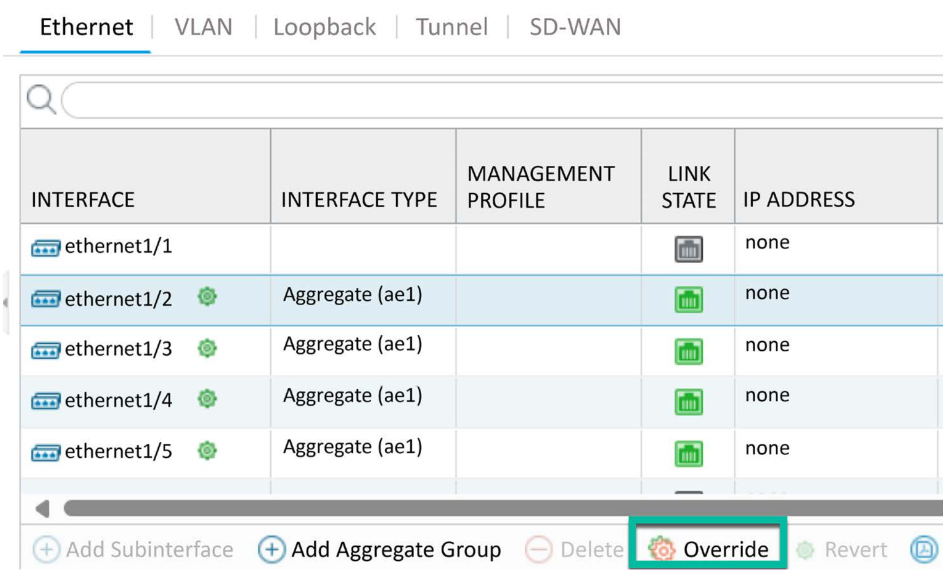


Figure 7.33: Applying an override to a Panorama template configuration

You can then change the parameters and commit to activate the new configuration. If you later want to revert to the Panorama template settings, you can select the object and click **Revert**.

Panorama can also function as a **user ID collector and redistribution center**, which can be implemented as follows:

1. Add all your deployed user ID agents (server-installed user ID agents or firewall-sourced clientless user ID collectors) to **Panorama** | **Data Redistribution** | **Agents**, Panorama will start to collect all the user-to-IP mapping from these agents and store them locally.
2. Then, go to **Panorama** | **Setup** | **Interfaces** | **Management Interface** and enable **User-ID Services**. Panorama can then be targeted by firewalls as a user ID agent.
3. In your template (stack), go to **Device** | **User Identification** | **User-ID agents** and add a new user ID agent. Instead of **Host** and **Port**, use the serial number and put in the serial number of your Panorama instance. If Panorama is set up in HA mode, add another user ID agent and add the serial number of the **second** Panorama instance.
4. Your firewalls will now collect all their user ID information from Panorama.

To back up configuration files, go to **Panorama** | **Scheduled Config Export** and create a backup profile:

1. Give the profile a friendly name and check the **Enable** box.
2. Set a convenient time, such as `22:30`.
3. Select the protocol to use for transfer. SCP is preferred as it provides encryption.
4. Set the hostname, port, path, username, and password.
5. Select whether you want to use **PASV** mode if you selected FTP as the transfer protocol.

This scheduled backup will save a bundle containing all the Panorama configuration settings and the managed device local configuration so that you have a handy backup of all the configuration settings.

Another great feature to remember is the ability to recover to a previous configuration if connectivity to Panorama is lost after a commit. In **Device** | **Setup** | **Management** | **Panorama Settings**, you can find **Enable automated commit recovery**. If this option is checked, a connectivity check is performed after each commit. If, due to the commit, the test fails (for example, by adding a security rule that blocks Panorama connections), the config will be rolled back. If you do need to make a change that will interrupt connectivity for a while, like changing to a new Panorama IP, first disable this option before pushing the change.

With all this information added to your arsenal of knowledge, you will be able to deploy a fully functional Panorama and import or deploy firewalls. The templates and device groups will help you to consolidate all the shared configuration parameters and quickly bring new devices up to company standards.

Summary

In this chapter, you learned about the Panorama central management platform and how it can be used to make managing groups, clusters, and geographically spread-out firewalls, users, and locations much less complex. You learned how device group configuration and templates can be used to simplify and make configuration consistent across all of your managed devices.

If you're preparing for the PCNSE, take note of what the templates and device groups are for. You should be able to explain how firewalls are deployed and managed from Panorama. Also, take special note of the automatic commit recovery, as this is part of the PCNSE blueprint.

In the next chapter, we will review the best practices for upgrading firewalls and Panorama.

8

Managing Firewalls Through Strata Cloud Manager

In this chapter, we will learn about Strata Cloud Manager (SCM), the new cloud-based central management platform that enables administrators to manage their firewalls and Prisma Access instances without needing to deploy Panorama. You will also learn how to onboard firewalls and manage Strata Logging Service (SLS).

In this chapter, we're going to cover the following main topics:

- Setting up Strata Logging Service
- Activating Strata Cloud Manager
- Managing Strata Cloud Manager
- Associating devices to Strata Cloud Manager
- Managing devices and device configuration through workflows
- Exploring dashboards

By the end of this chapter, you will be able to centrally manage all of your firewalls and consolidate shared configuration among groups of devices.

Setting up Strata Logging Service

Unlike Panorama, which you can deploy from a hardware platform, as a VM on a local hypervisor or via a cloud marketplace, Strata Cloud Manager and Strata Logging Service can only be deployed on the Palo Alto Networks hub at `https://apps.paloaltonetworks.com`. (The hub is a portal from where you can access all the cloud services such as Cloud Identity Engine (CIE), Strata Logging Service, etc.)

Panorama can leverage local log collectors or direct firewalls to send logs to SLS. SCM requires the use of SLS to collect logs from all managed firewalls. In other words, you need to set up SLS as well if you plan to manage your firewalls via SCM.

When you order SLS from your vendor, you will receive an email with an activation link, which will take you to the activation page.

Activating Strata Cloud Manager

Strata Cloud Manager has a few different ways to be activated at the time of writing, depending on which products you own and how you are activating it:

- If you have standalone firewalls and want to achieve the same sort of centralized management as Panorama, you can deploy a free version of SCM called **Strata Cloud Manager Essentials**. To receive logs in the cloud, you'd need to add a SLS license.

  You can activate SCM Essentials via this activation link: `https://apps.paloaltonetworks.com/activation/scm-essentials.`

  After your SCM tenant has been deployed, you may need to reach out to your Palo Alto sales team to activate it for **next-generation firewall (NGFW)** management (which is also free).

  I've used this license to cover the majority of content in this chapter as it is free to use, so you'll be able to use it easily too.

- The **Strata Cloud Manager Pro** license is a paid subscription license that takes all the features of SCM Essentials and adds additional features like the Policy Optimizer and Policy Analyzer, which help you track compliance and rule usage, custom dashboards, and more. A full comparison can be found here: `https://docs.paloaltonetworks.com/strata-cloud-manager/activation-and-onboarding/strata-cloud-manager-licenses-and-support`.
- Additionally, you can also purchase an **AIOps for NGFW** license, which bundles advanced telemetry and reporting, with the functionality of SCM. AIOps is an add-on subscription license that processes the uploaded telemetry and provides some additional insights on the SCM dashboard.

> There is also a free AIOps license (AIOps Free), but this does not provide the SCM functionality.

- If you're activating Prisma Access and select **cloud managed** while activating, this will automatically deploy SCM.
- If you have the appropriate licenses already loaded into your account, you can also deploy SCM directly from your hub account at `https://apps.paloaltonetworks.com`.

If you're going to deploy a hybrid environment where you will be managing both physical firewalls and Prisma Access from the same SCM, you will need to use the activation link for SCM Essentials before you activate Prisma Access.

Any services not activated yet but with a valid license (or available for free) attached to your support account will show in the hub portal with an **Activate** button in the tile. If you didn't receive the activation email or misplaced it, you can continue from the **Activate** button or the activation link on the previous page.

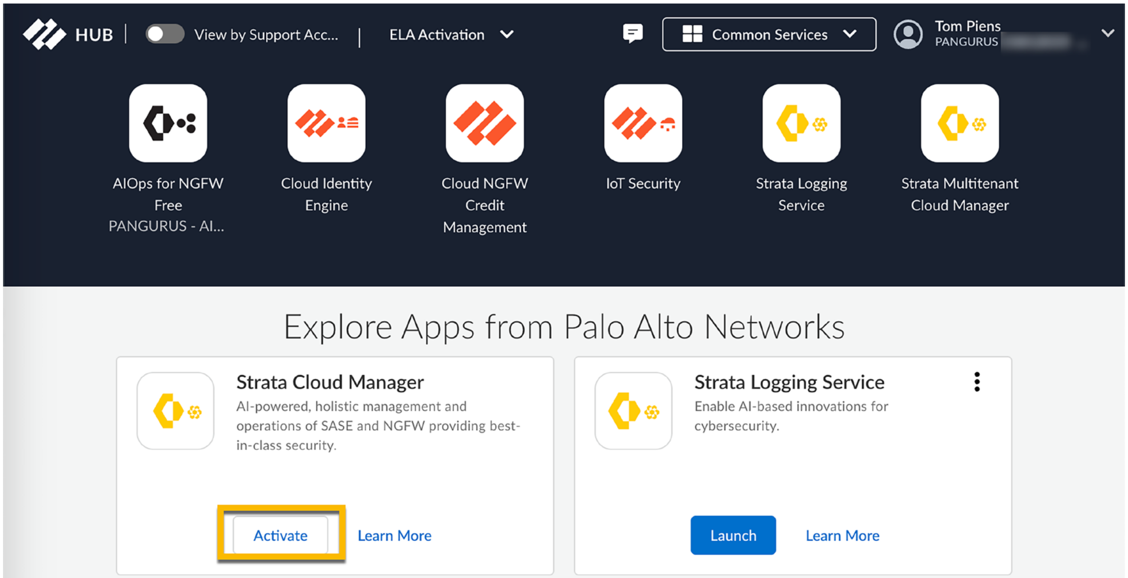


Figure 8.1: Activating Strata Cloud Manager from the hub

Before you continue, there are a few things to consider: Palo Alto provides the hub environment as a single-tenant environment with the ability to create subtenants.

This can come in handy if you need to split up different environments into their own subtenants. This can also help prevent conflicts, for example, a SLS instance used for Cortex XDR that should not share data with a set of firewalls in a different subtenant, or, as you may notice in Figure 8.1, there is a free version of AIOps for NGFW active that may conflict with SCM, so if you have AIOps Free deployed in your tenant, it's better to create a new or a subtenant. If AIOps Free has not been activated before, you can deploy SCM Essentials in your main tenant.

Let's look at how to create a subtenant.

Creating a subtenant

To create a subtenant, log in to `https://apps.paloaltonetworks.com` and, from the **Common Services** menu dropdown, select **Tenant Management:**

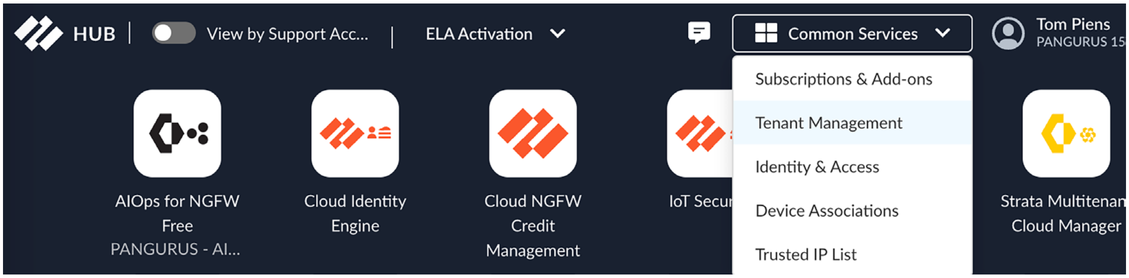


Figure 8.2: Tenant Management

From there, you will have the option to **Add Tenant** in the upper-right corner (the + sign):

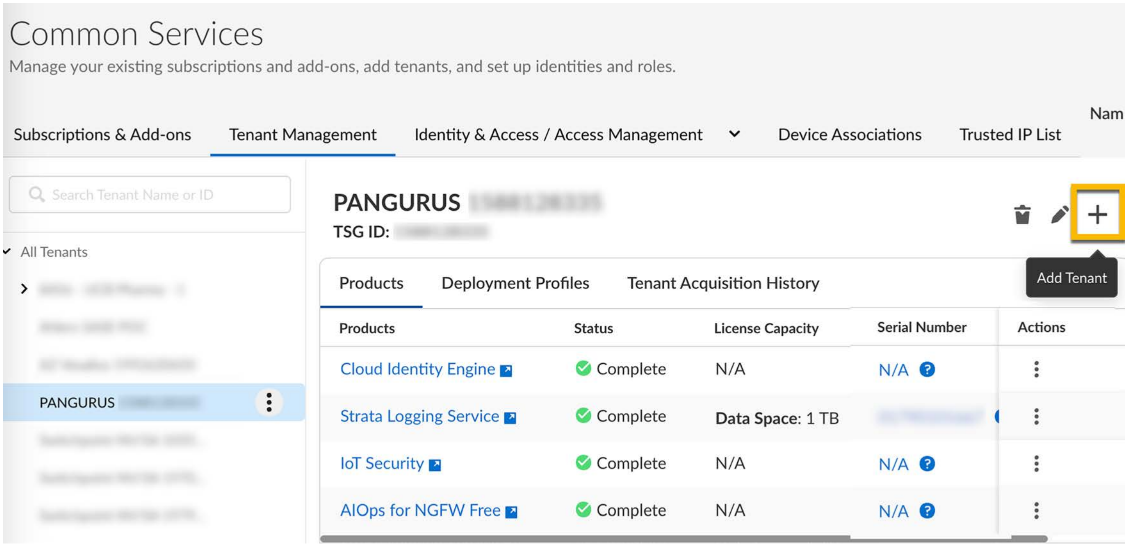


Figure 8.3: Add a new tenant

Next, you'll need to name the new subtenant, select a vertical if you want to share that information, inherit or set a new support contact, and choose the **User Inactivity Timeout** time.

The default inactivity timeout is 60 minutes and can be set as short as 10 minutes. Set the timeout to comply with company policy for sensitive management interfaces:

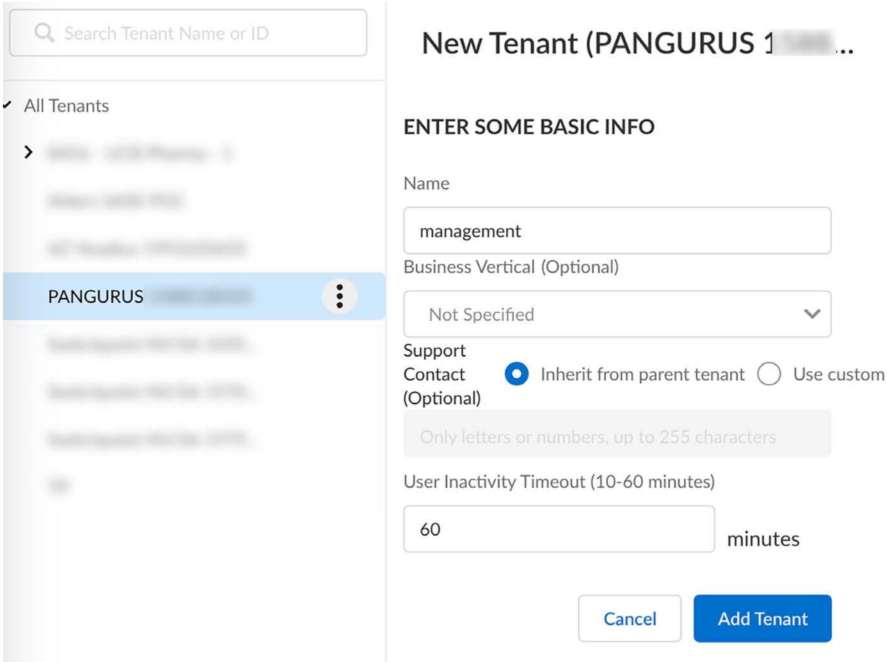


Figure 8.4: Adding a tenant

All the information from the master/root tenant will be shared with the subtenant, so, for example, an existing Cloud Identity application in the master tenant can be used in the subtenant.

Now that you've added the new tenant, you can select your new subtenant by returning to the hub, clicking your name in the upper-right corner, and selecting the subtenant. You can now activate new products in this subtenant.

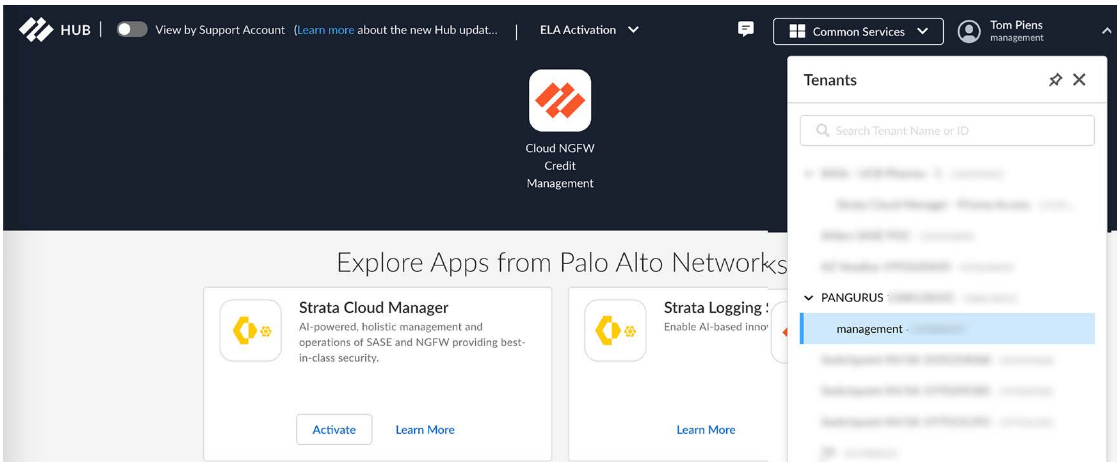


Figure 8.5: Selecting the new subtenant

Now, let's return to SCM activation.

Activating Strata Cloud Manager from the hub

When you start the activation process, regardless of whether it's via an email link or the hub activation page at `https://apps.paloaltonetworks.com/activation/scm-essentials`, you'll come to the same activation process:

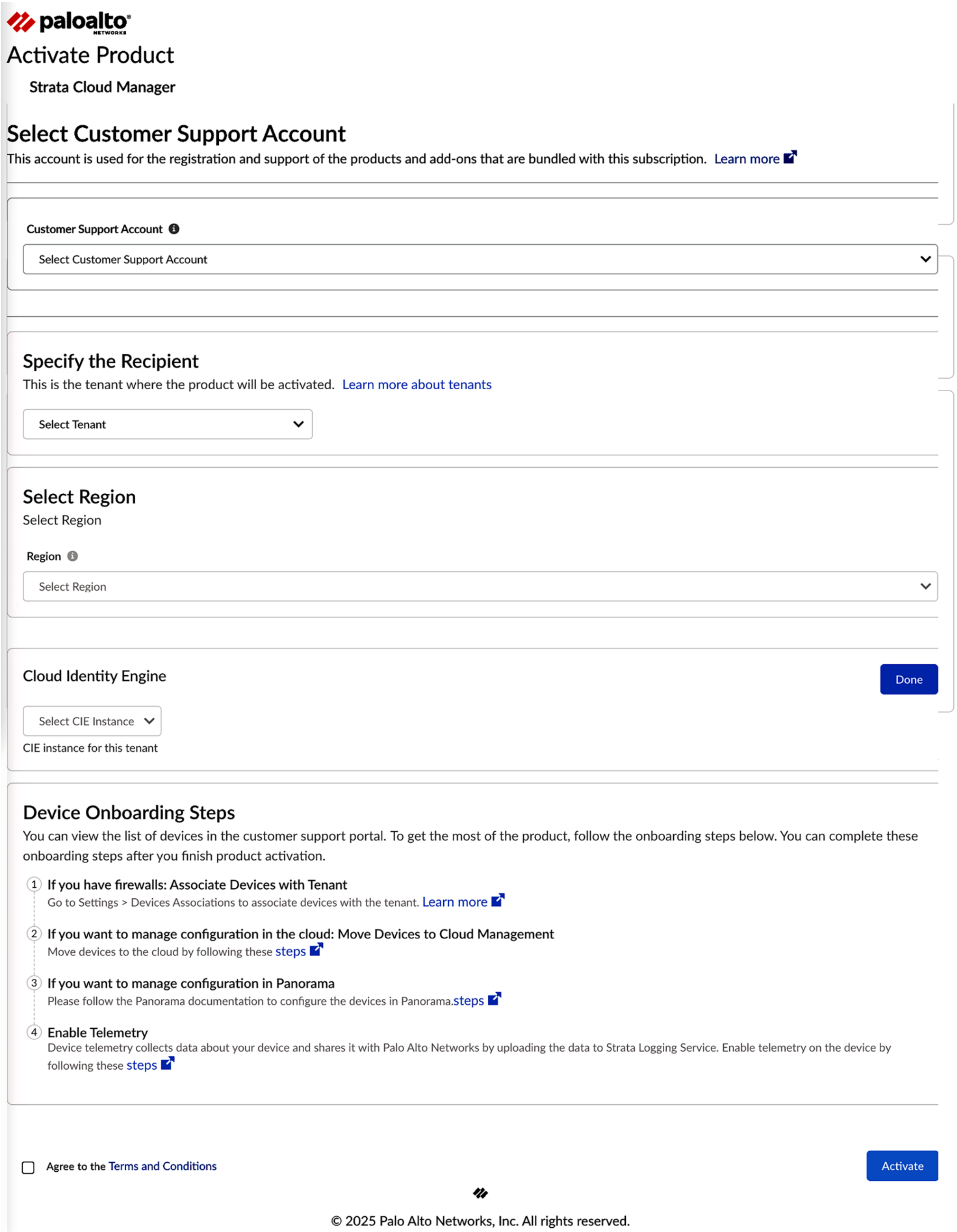


Figure 8.6: SCM activation

You'll need to select a few things before you can carry on:

1. **Customer Support Account** is the CSP account that contains the appropriate licenses. If your support account is attached to multiple CSP accounts, there may be several support accounts you can select or there may just be one.
2. **Specify the Recipient** is the tenant you want to deploy SCM into. This can be your master tenant or a subtenant.
3. **Select Region** lets you select the region in which this tenant will be deployed. It is very important you select the same region as the SLS. Failing to select the same region may cause your firewalls managed by SCM to log to the SLS, or vice versa.

   If the tenant has an active SLS subscription, the region will automatically be selected. If the tenant does not have an SLS active yet, consider activating that first before proceeding with the SCM.

4. If a CIE instance already exists, you can select it in the **Cloud Identity Engine** section to use in conjunction with SCM; if not, you can select to create a new one or select **None**.

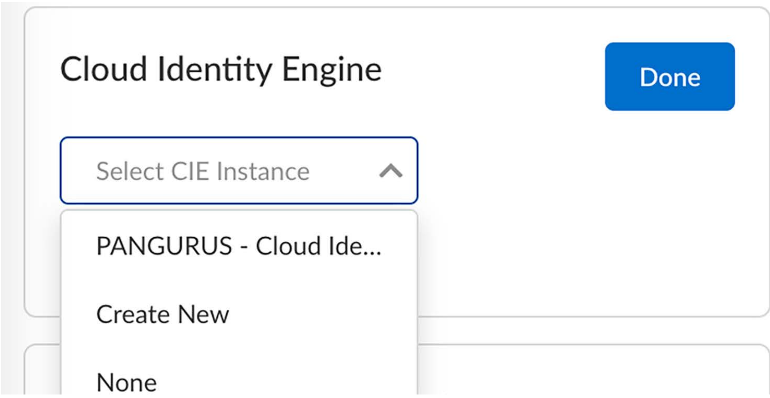


Figure 8.7: Select CIE Instance

Lastly, agree to the terms and conditions before clicking **Activate**:

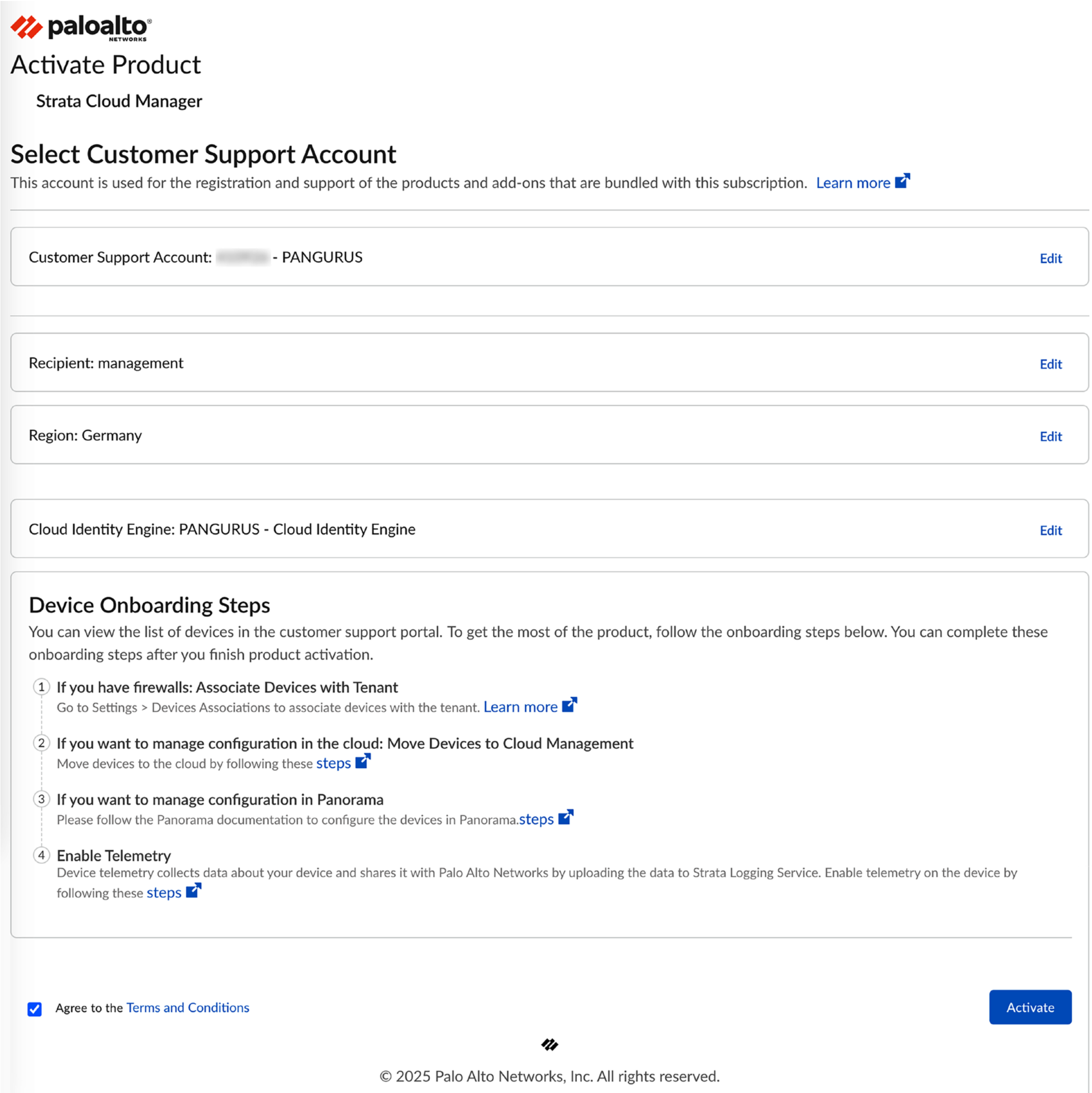


Figure 8.8: Completed activation form

When you click **Activate**, you'll be taken back to the **Tenant Management** page and a progress bar will show the SCM being provisioned:

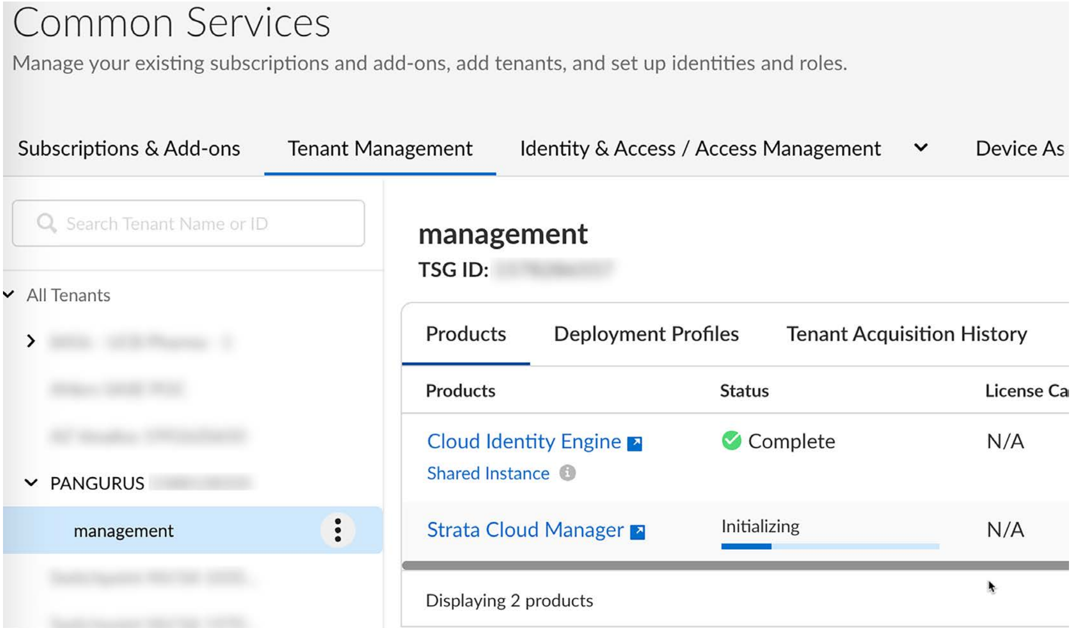


Figure 8.9: SCM provisioning

Activating AIOps or Strata Cloud Manager for NGFW

The activation email for AIOps will look similar to the screenshot below:

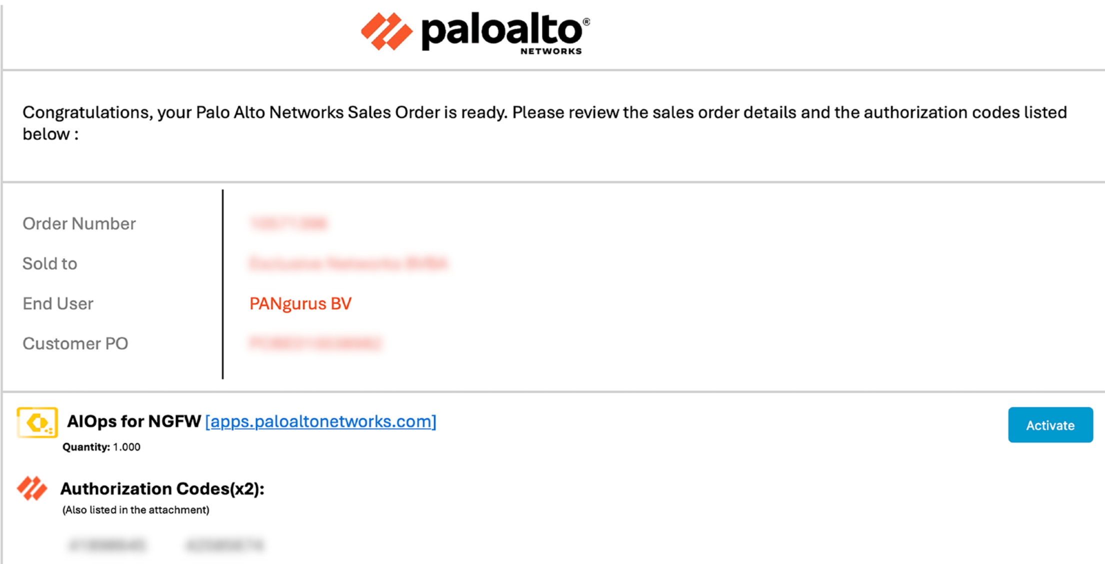


Figure 8.10: AIOps for NGFW activation email

Using the activation link from the email (the process will be identical for both AIOps and Prisma Access) will bundle all related licenses in the activation form, so the SLS will be listed as well:

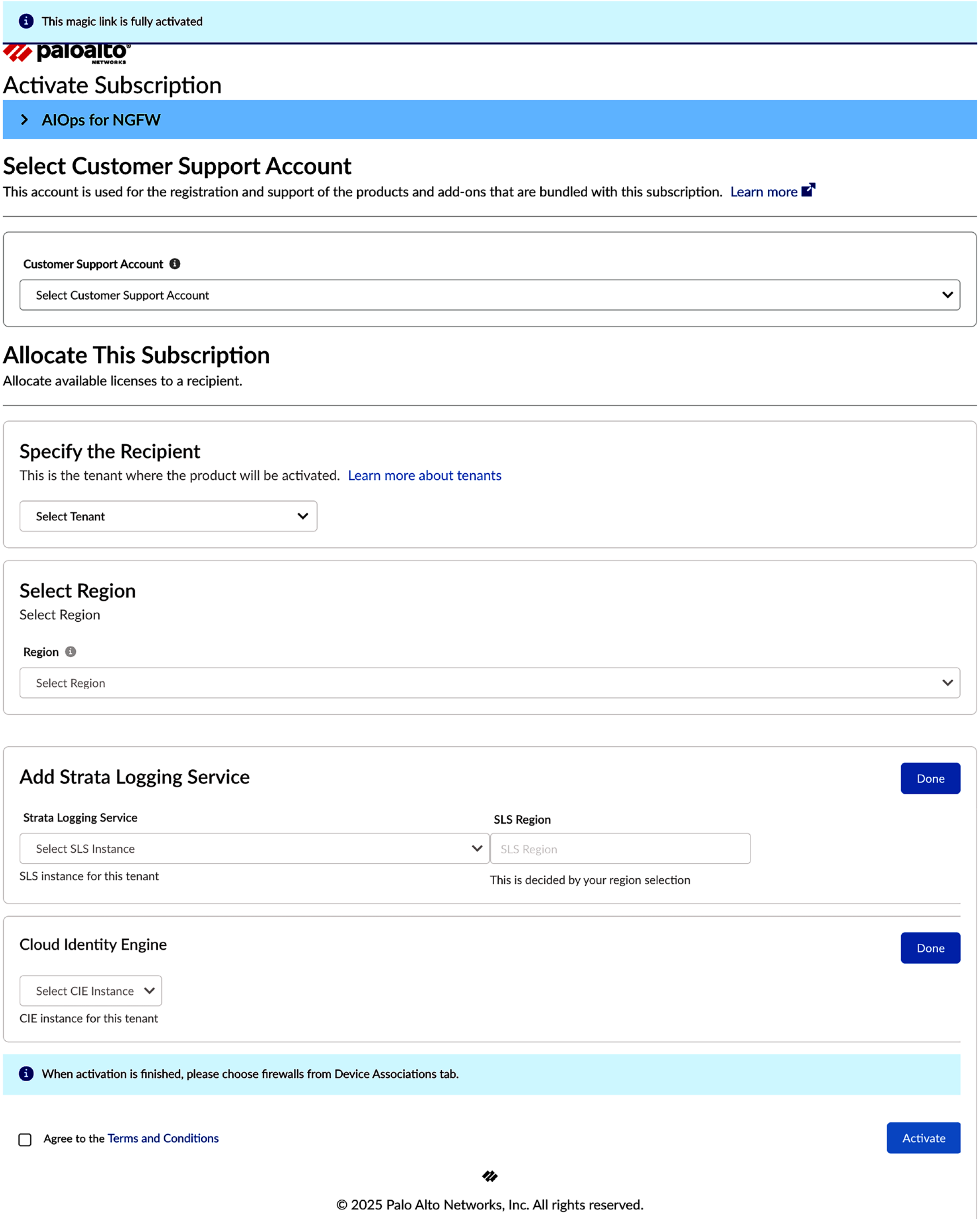


Figure 8.11: Activating AIOps Pro for NGFW from the "magic link"

You'll need to fill out these fields:

1. **Customer Support Account** is the **Customer Support Portal** (**CSP**) account that contains the appropriate licenses. If your support account is attached to multiple CSP accounts, there may be several support accounts you can select or there may just be one.
2. **Specify the Recipient** is the tenant you want to deploy SCM into. This can be your master tenant or a subtenant.
3. **Select Region** lets you select the region in which this tenant will be deployed.
4. In the **Add Strata Logging Service** section, either select an existing SLS or assign a volume in terabytes (TB) and a region for the SLS.
5. If a CIE instance already exists, you can select it in the **Cloud Identity Engine** section to use in conjunction with SCM, or select to create a new one or select **None**.
6. Once you select the **Agree with the terms and conditions** checkbox and then click **Activate**, the tenant will be spun up.

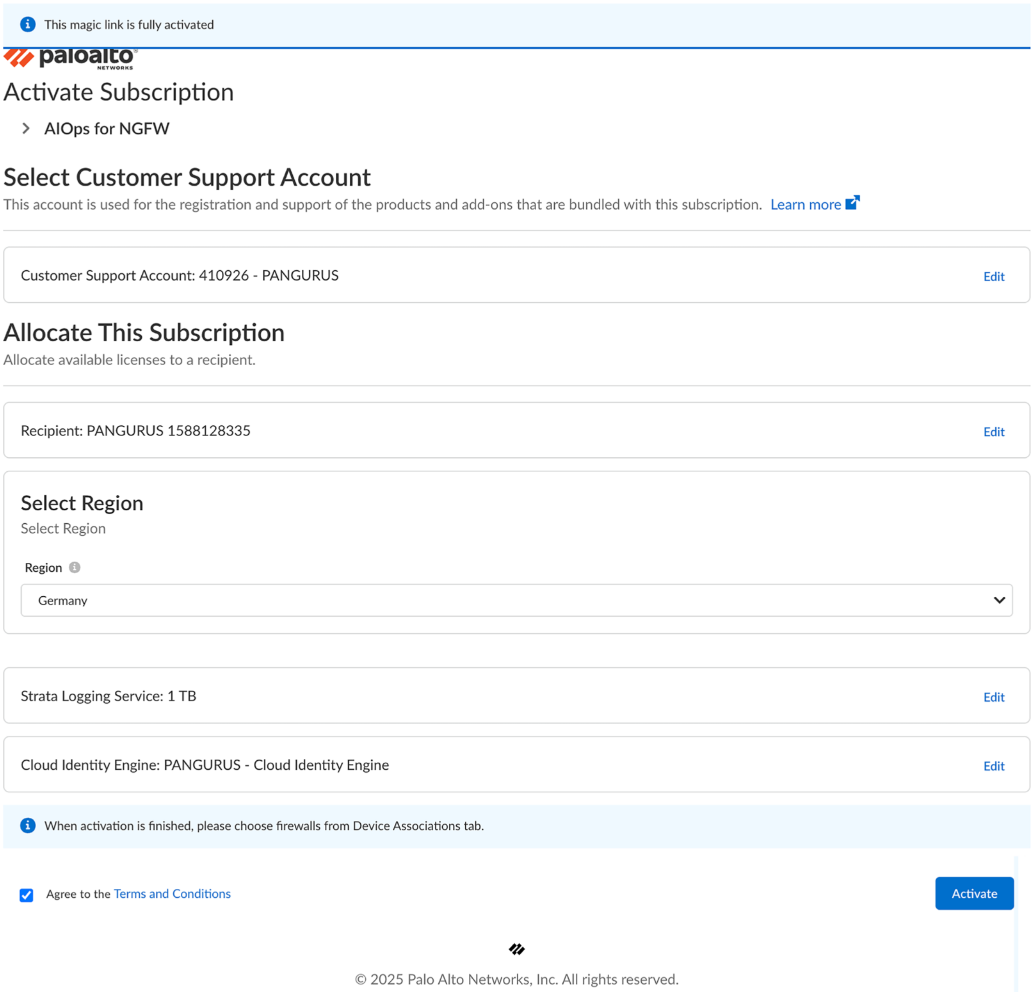


Figure 8.12: Activating AIOps for NGFW from the "magic link"

You are now able to access SCM via the Strata Cloud Manager tile at `https://apps.paloaltonetworks.com` or directly via `https://stratacloudmanager.paloaltonetworks.com/`.

Configuring Strata Cloud Manager

Since we're starting from a fresh configuration, the Dashboard page will not show you much at this point.

We'll first want to go to the **Manage** tab.

Starting with the Manage tab

In the **Manage** tab, you will be able to configure all the rulebases and all the associated objects that will be used in the policies. You can commit policies and review all jobs, run a security posture check, and change the hierarchy of the Configuration Scope (similar to Device Groups in Panorama).

We'll first need to activate the configuration service. To do that, we need to click **Start Managing Configs** in the **Manage** tab.

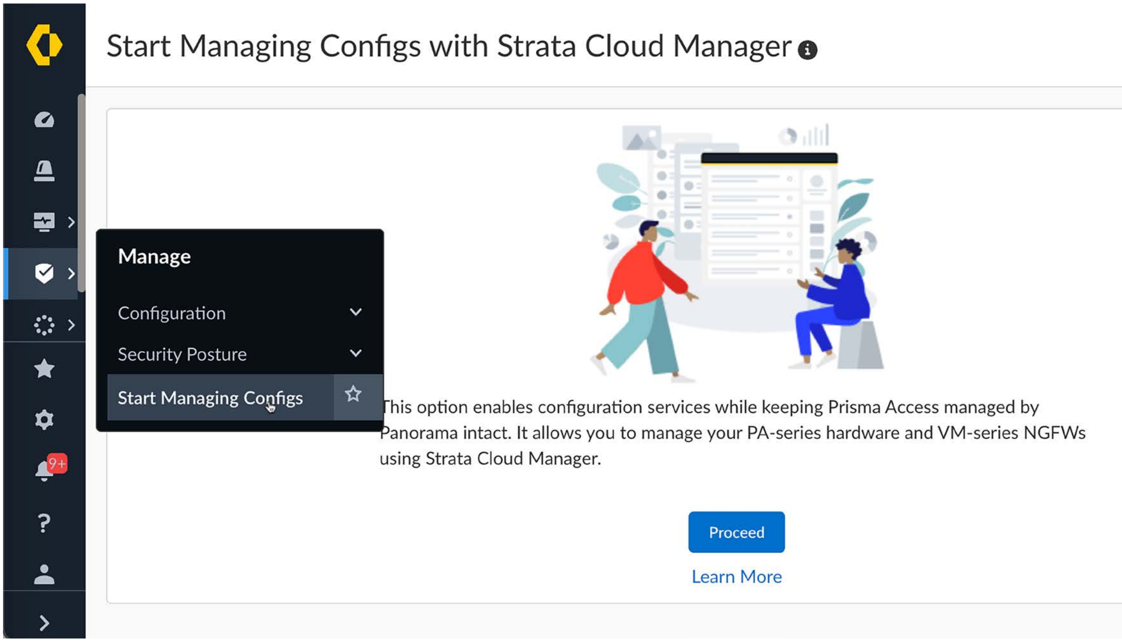


Figure 8.13: Start Managing Configs with Strata Cloud Manager

On the next page, click **Enable the Configuration Service** and, because this process can take up to five minutes, you can take a little break:

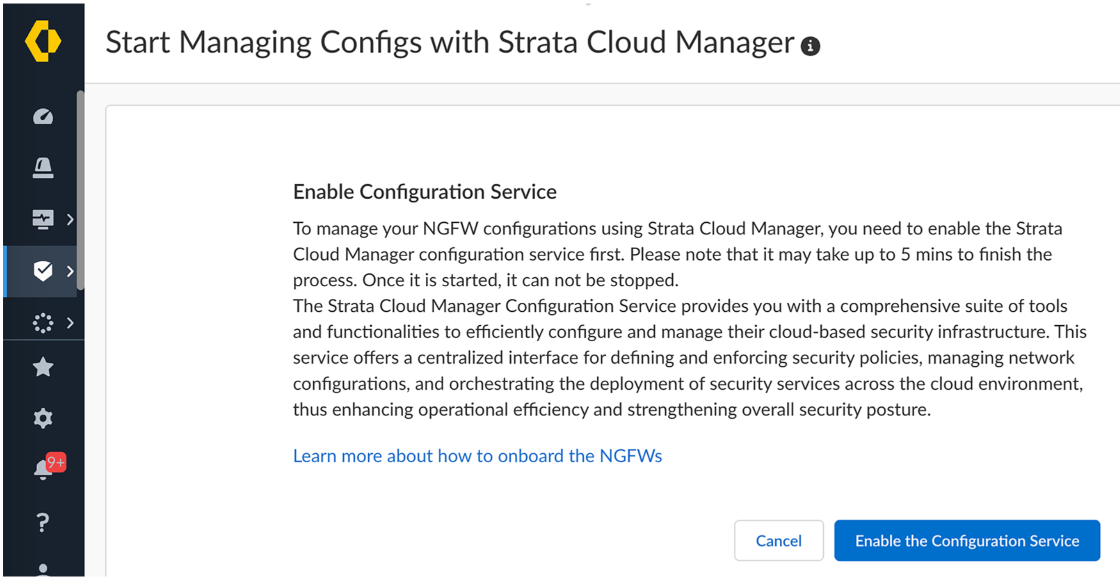


Figure 8.14: Enable Configuration Service

Once the process is complete, we can access the **Configuration** page.

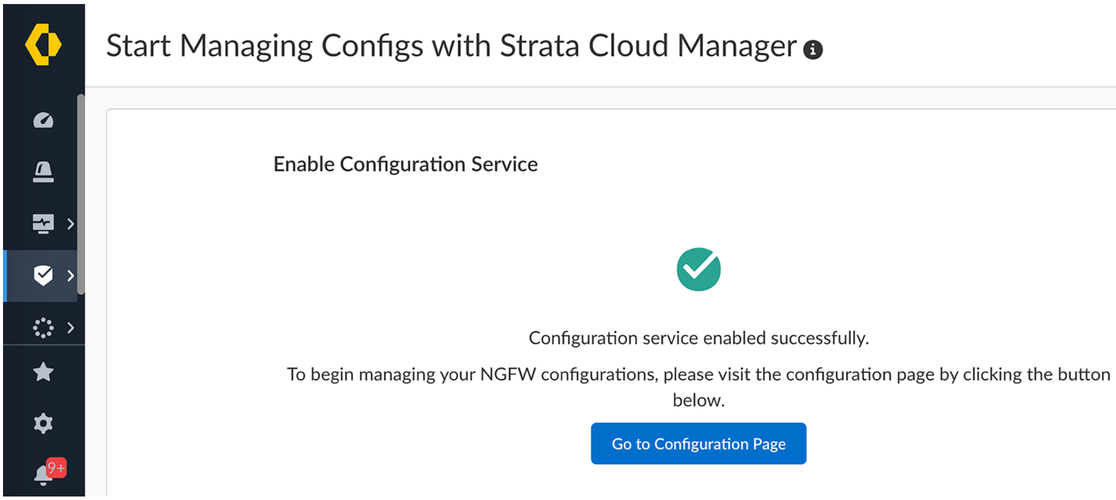


Figure 8.15: Configuration service enabled successfully

Clicking **Go to Configuration Page** will take you to **Manage** | **Configuration** | **NGFW and Prisma Access**.

NGFW and Prisma Access

The **NGFW and Prisma Access** section contains all the rules and objects you would typically find in the **Policies** and **Objects** tabs in the normal PAN-OS web interface. Once you access this section, you will see the following drop-down options available at the top of the screen:

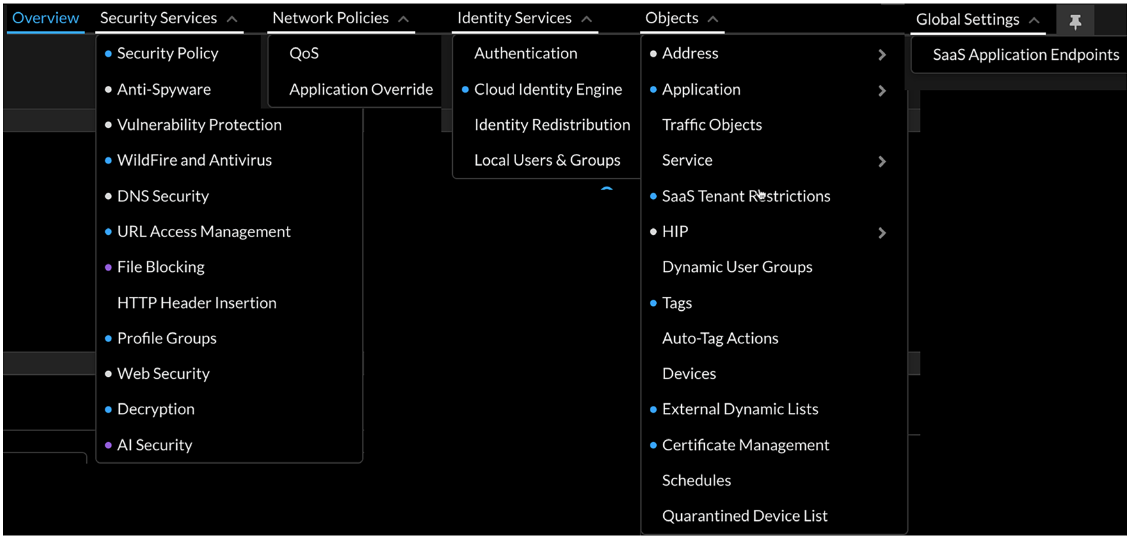


Figure 8.16: NGFW and Prisma Access menus

The dots next to each menu item mean that something (rules, objects, etc.) is configured in them. Some default rules and objects already exist before you make your first config changes.

Let's have a quick look at these options.

Overview will give you a high-level view of your overall deployment. You can assign snippets (small sets of security rules – see the *Snippets* section later in the chapter) at each level of the configuration scope (which is similar to device groups in Panorama), as well as a link to create a tech support file in case you need to reach out to Support.

Security Services contains all of your rulebases and security profiles:

- **Security Policy** is where your firewall rulebase is maintained.
- **Anti-Spyware** is where the anti-spyware security profiles are configured.
- **Vulnerability Protection** is where the vulnerability protection profiles are configured.
- **WildFire and Antivirus** is where WildFire and antivirus security profiles are configured.
- **DNS Security** is where DNS security profiles are configured.
- **URL Access Management** is the URL filtering security profiles.
- **File Blocking** is where the file-blocking security profiles are configured.

- **HTTP Header Insertion** is now a separate section; in the traditional web interface, this was part of the URL filtering profile. It allows you to set sanctioned SaaS applications that may also be used for private use so that users are only able to access the corporate tenant (Office365, Gsuite, etc.).
- **Profile Groups** is where you can make groups of security profiles.
- **Web Security** is where the Prisma Access browser policies are created.
- **Decryption** is the decryption rules.
- **AI Security** is a new section dedicated to AI tools. This feature requires the Cloud Access Security Broker (CASB) or AI Access Security license, so this option may not be available if you have neither. These features help protect AI tools from prompt injection, sensitive data leakage, insecure output, and model denial of service (DoS).

Network Policies contains a few low-level security rulebases:

- **QoS** is where QoS rules are configured.
- **Application Override** is where the app override rules can be created.

Identity Services contains all the configuration related to identity and authentication:

- **Authentication** is where the authentication portal rules are configured.
- **Cloud Identity Engine** is where you can configure the CIE integration.
- **Identity Redistribution** is where user ID redistribution can be configured.
- **Local Users & Groups** contains the local user accounts (for authentication against the firewall, not for accessing SCM).

Objects is where all the objects are configured:

- **Address** contains addresses, address groups, and regions.
- **Application** contains applications, application groups, and application filters.
- **Traffic Objects** is where Kubernetes clusters with the PAN-CNI plugin can be controlled.
- **Service** contains the services (ports) and service groups.
- **SaaS Tenant Restrictions** allows you to restrict access for public SaaS applications to specific tenants (e.g., only allowing the corporate Office365 apps, but not private).
- **HIP** contains the **host information profile (HIP)** objects and HIP profiles.
- **Dynamic User Groups** is where you can configure the dynamic user group match profiles.
- **Tags** contains the tags used in security rules and dynamic objects.
- **Auto-Tag Actions** now has its own section; in Panorama and the regular web UI, this is part of the **Log Forwarding** profile. Auto-Tag can add tags to source users or IP addresses when they trigger an event. The tag can then be used to temporarily restrict access to sensitive resources. For example, if an endpoint triggers Auto-Tag due to a vulnerability detected, a security rule can then restrict access for the tagged source to only remediation servers, or require additional authentication for any connections to sensitive resources.

- **Devices** is where you can configure a device match profile that can be used in security rules to determine which IoT devices match the security rule (this requires the IoT security license).
- **External Dynamic Lists** is where the **external dynamic lists** (**EDLs**) are configured.
- **Certificate Management** is where all the certificates used by the systems are stored. Trusted root certificates, forward trust certificates and (web) service certificates can be uploaded here. In the regular Web UI, this can be found in the **Device** tab.
- **Schedules** contains the schedule profiles used to control when security rules are active.
- **Quarantine Device List** contains the quarantined devices. Devices can be manually released from quarantine here.

Global Settings contains overarching configuration:

- **SaaS Application Endpoints** allows you to indicate which SaaS applications are being used by your organization and in which region they are active.
- As part of the Enterprise Data Loss Prevention (EDLP) license, there is the **User Coaching Notification Templates** option, in which user coaching templates can be created to pop up as part of the GlobalProtect deployment to coach users when they share sensitive or restricted data.

Similar to Panorama, SCM also has a hierarchy where policies can be applied at a global level, a regional level, or a subset. This is called the c**onfiguration scope**. The configuration scope hierarchy is as follows (**Global** contains rules and objects that apply to the entire install base, physical or virtual):

- **All Firewalls** (which may not be visible if you don't have any managed firewalls) contains all the managed firewalls' configuration (see the *Workflows* section further in this chapter).
- **Prisma Access** (which may not be visible if you don't have Prisma Access) applies to all components of Prisma Access:
    - **Mobile User Container** applies to all mobile users (GlobalProtect or Explicit Proxy if it is enabled):
        - **GlobalProtect** applies only to GlobalProtect users
        - **Explicit Proxy** (not visible in Figure 8.17) applies to clientless proxy endpoints
    - **Remote Networks** applies to all remote networks.
    - **Service Connections** has some components that can be found in the NGFW and Prisma Access section, such as QoS rules, certificate management, and user ID redistribution, even though service connections do not apply any security rules.

This is the default Prisma Access hierarchy, but if you manage additional firewalls, you can create more hierarchy levels to suit your needs (as you would in Panorama):

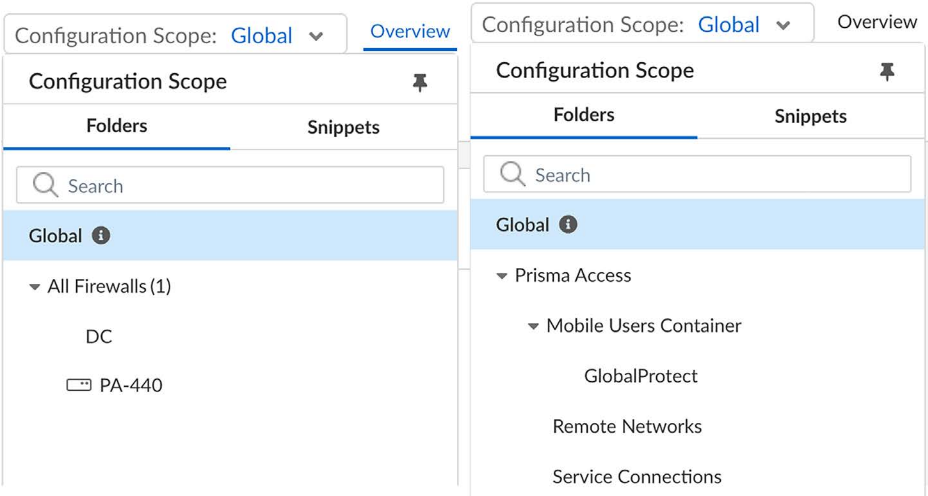


Figure 8.17: NGFW hierarchy and Prisma Access

This means that, just like in Panorama, any rules created in the GlobalProtect configuration scope section will only apply to GlobalProtect users, while any rules created at the Prisma Access level will apply to GlobalProtect users, Explicit Proxy users, and remote networks, and objects created at the Service Connection level will only be available to service connections. Objects created at the Global level, however, can be used by all Prisma Access components, and all onboarded NGFW, physical and virtual.

Security rules

Creating security rules in SCM is very similar to Panorama: you first need to select the configuration scope (device group) hierarchy level where you want to create the rule to determine where the rule will be applied (**Global**, **All Firewalls**, **Prisma Access**, or one of the subsets). The higher up the hierarchy, the more subfolders will inherit the configuration.

Next, you click **Add Rule** to create a new rule:

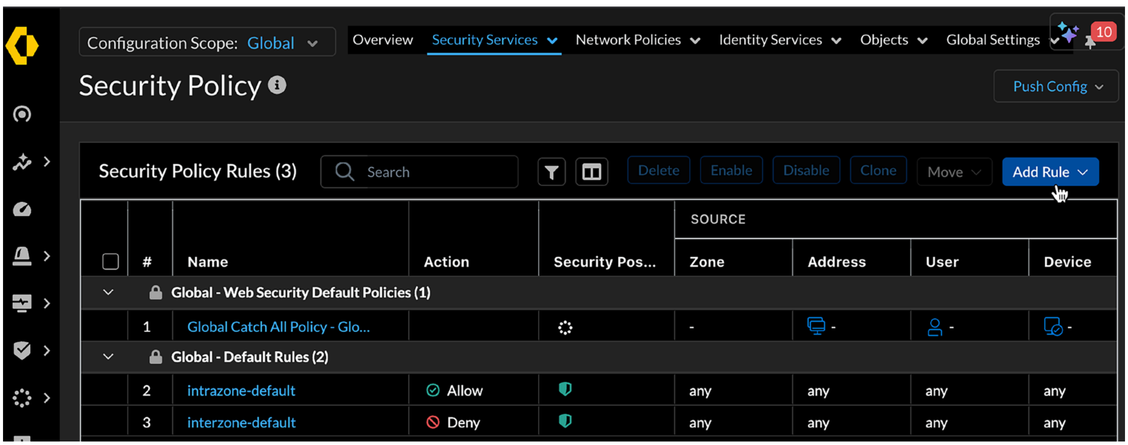


Figure 8.18: Creating a new rule

In any of the parent scopes, you can create pre- and post rules that will be applied before or after rules created in the child scope; in a child scope, you can only create regular rules. A pre- rule created in Global will be placed at the top of the rulebase for all lower-hierarchy configuration scopes, and post rules in Global will be placed at the very bottom. (Any post rules created in the Prisma Access configuration scope will be placed at the bottom of **Mobile Users** and **Remote Networks** rules but above the Global post rules.)

A new type of security rule, **Internet Access Rule**, is specifically designed to address more granular controls to internet access by providing the ability to select URL categories and apply different file blocking, **data loss prevention (DLP)**, and even decryption policies to each category independently.

In the following screenshot, this is illustrated as an example of the **ChatGPT** application, in the **AI Code Assistant** URL category where different file blocking and DLP profiles are applied from the default ones. These types of rules can be added to the security rules and will act like any other security rule, in the order they are placed.

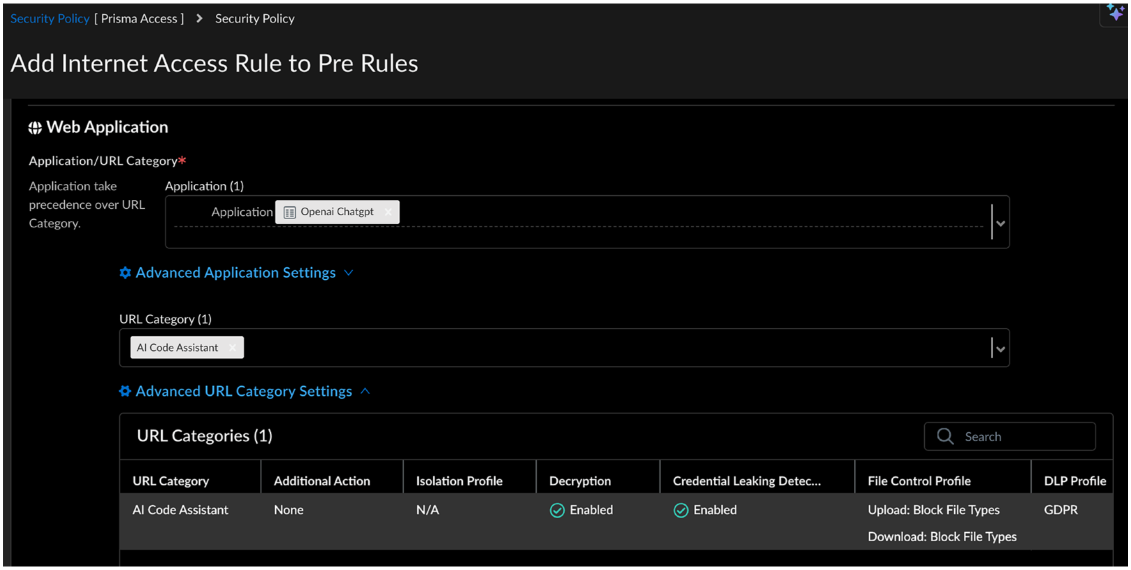


Figure 8.19: Internet Access Rule

A new concept in SCM security policies are Snippets.

Snippets

Snippets are little sets of security rules. They allow you to build or edit common sets of rules that may need to be used in many different locations (e.g., across different sets of firewalls and Prisma Access) without placing them at the **Global** configuration scope level.

There are a couple of preconfigured sets available to address very common use cases, such as **O365-best-practices** and the more generic **Recommended-Best-Practice**, that are added to the configuration scope as you can see in the following screenshot.

In this screenshot, we are in the **Prisma Access** configuration scope, on the **Overview** tab, and we have four configuration snippets attached to this scope. These snippets will need to be added or deleted according to your needs:

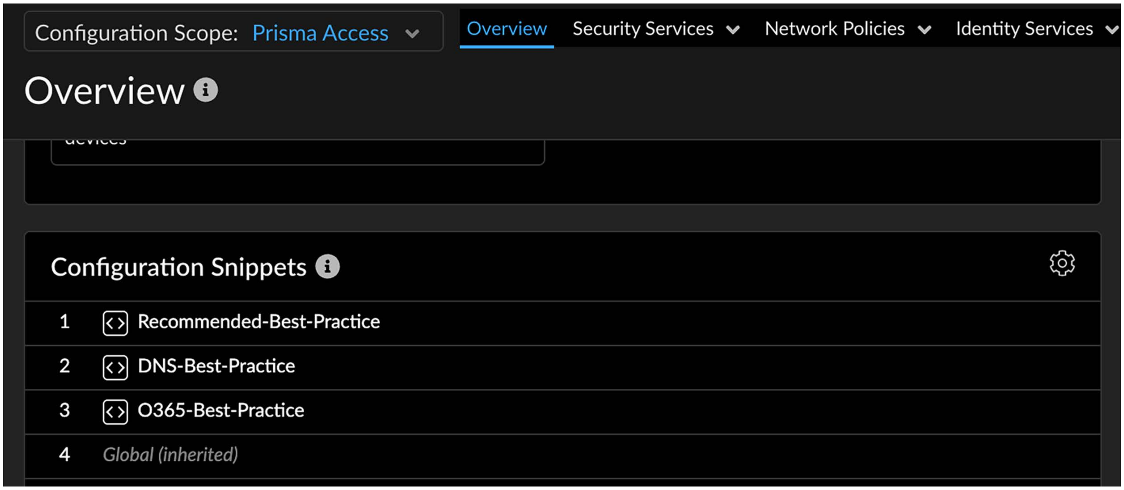


Figure 8.20: Adding snippets to a branch

You can add and remove snippets for each scope by clicking the cog wheel to the right. Snippets can be rearranged in a different order as well (this is how they will show up in the security rulebase). You can also rearrange them in the rulebase by clicking and dragging them to the desired spot in the rulebase:

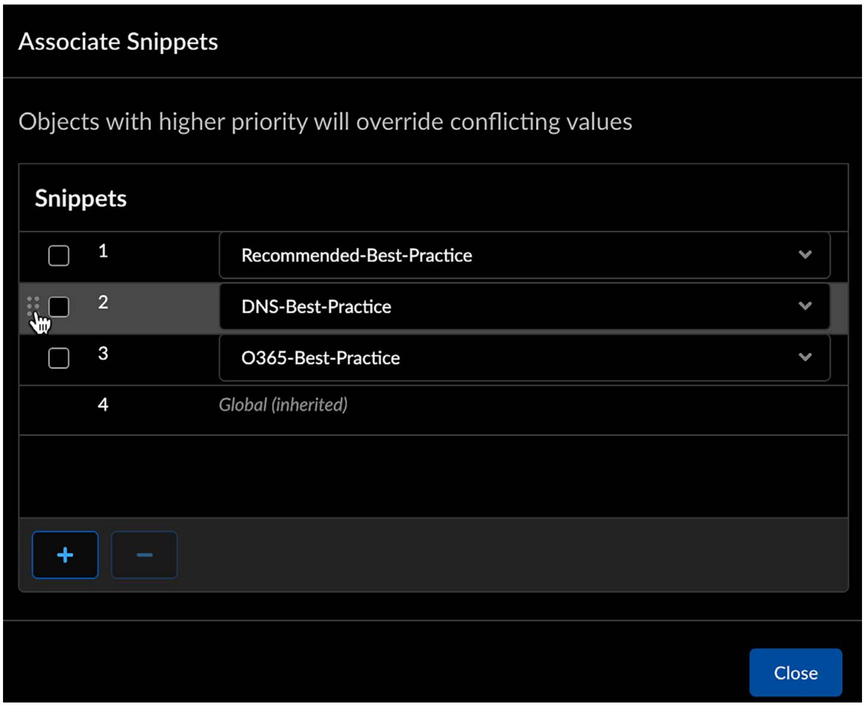


Figure 8.21: Rearranging snippets on a branch

To edit or add new snippets, you change the configuration scope from folders to snippets by selecting the **Snippets** tab. You will then see all the available snippets. You can click one to go into the configuration to make changes, click **Add Snippet** to create a new one, or click the kebab menu (the three vertical dots to the right of the snippets, when selected) to clone an existing snippet.

When you add a new snippet, it will create an entire folder with all the security rules, security profiles, and objects available to add to the snippet. This allows you to create unique profiles for each snippet without cluttering your configuration scope folders with the additional profiles and objects.

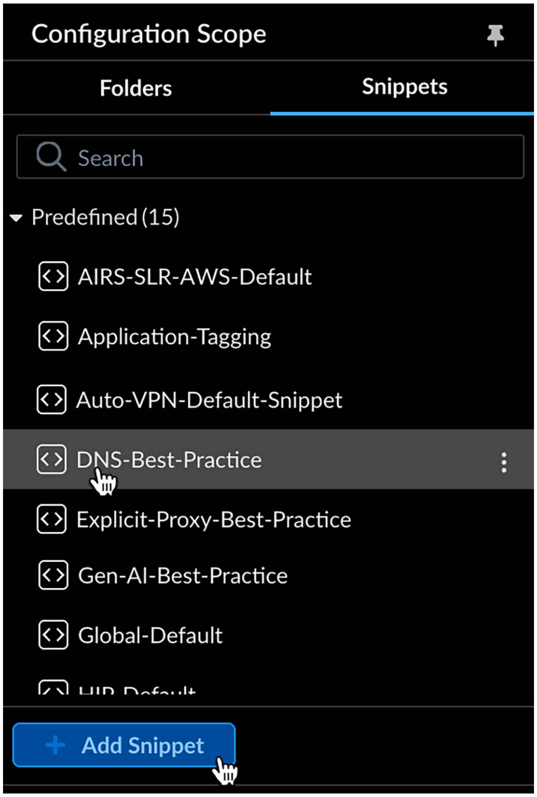


Figure 8.22: Edit or add snippets

Once you are in an existing or new snippet, you have all the same menu items available as any other configuration scope, but in this case, your configuration changes only apply to the snippet.

If we open the **DNS-Best-Practice** snippet we highlighted in the previous screenshot, we'll see that there is one rule preconfigured:

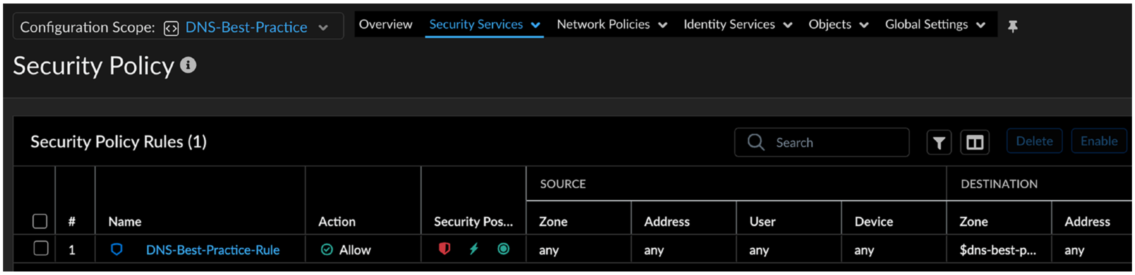


Figure 8.23: DNS-Best-Practice snippet

If we go back to the configuration scope we want to edit, we will see all the snippets that are placed in the selected place – in this case, **Prisma Access - Pre Rules**:

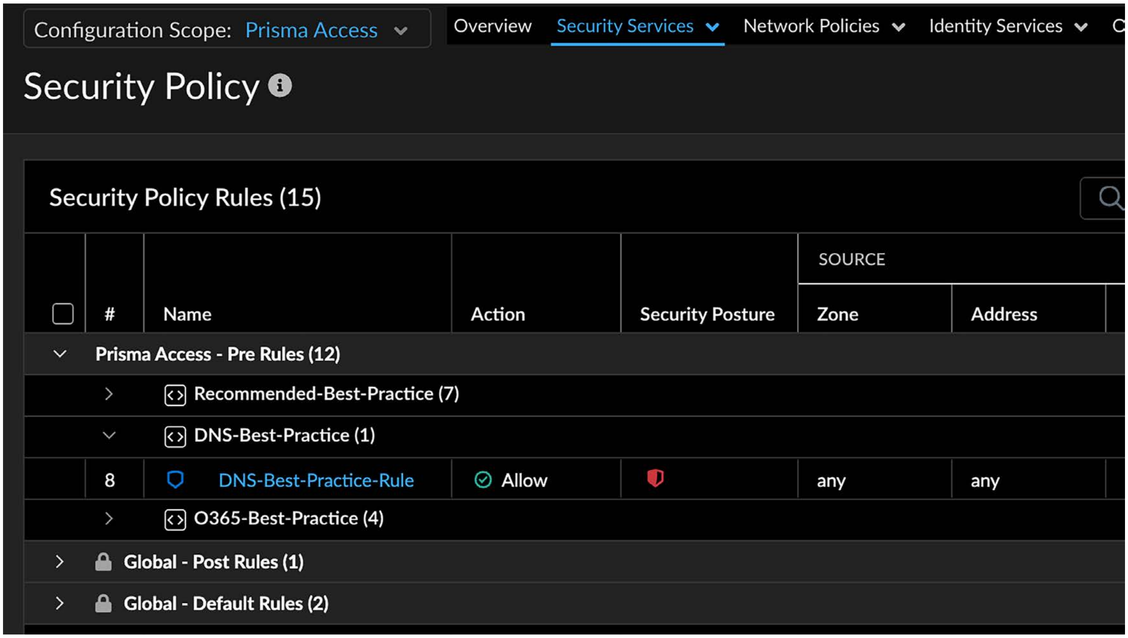


Figure 8.24: Snippets in Security Policy

Security profiles also follow a structured hierarchy, similar to Panorama.

Security profiles

All of the security profiles (**Anti Spyware**, **Vulnerability Protection**, **WildFire and Antivirus**, **DNS Security**, **URL Access Management**, and **File Blocking**) come with a **best-practice** profile preconfigured and set at the **Global** configuration scope. These default profiles are "locked," meaning you can't edit them:

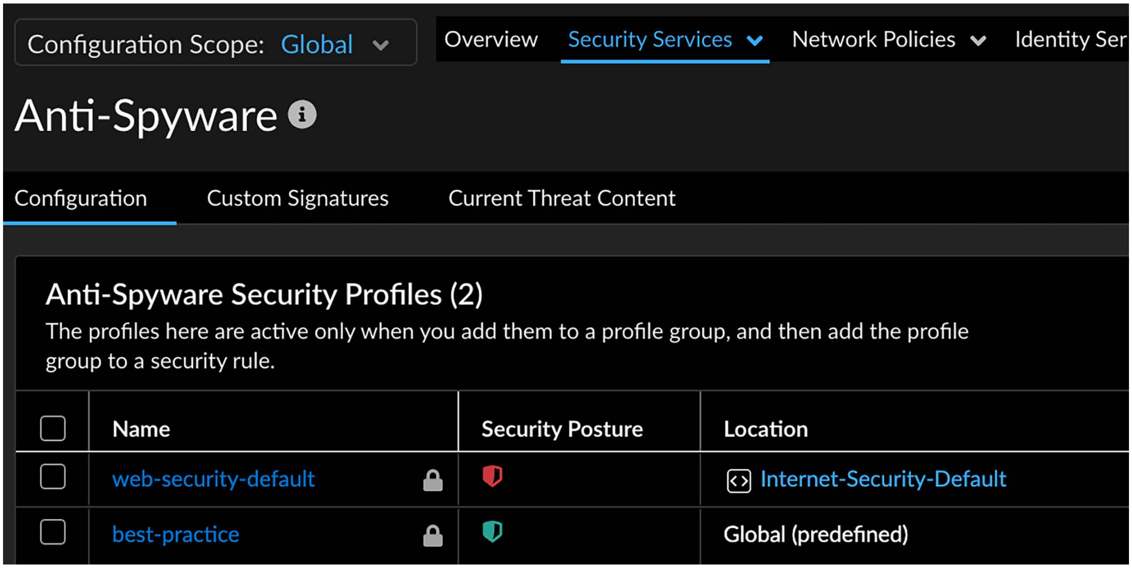


Figure 8.25: Default best-practice profile, as seen in the Anti-Spyware security profile

If you need to edit the default **best-practice** profile, you can select the profile and click the **Clone** button above it. If you keep the original name (**best-practice**), you will overwrite the locked profile and the cloned profile will be unlocked so you can edit it. This step can be repeated at every level in the configuration scope to customize the profile for each different level.

The advantage of cloning and editing the default **best-practice** profile rather than creating a new profile is that each time a new security rule is created, the **best-practice** profile group is automatically selected, which contains references to all of the **best-practice** profiles, making sure that the preferred configuration is always applied.

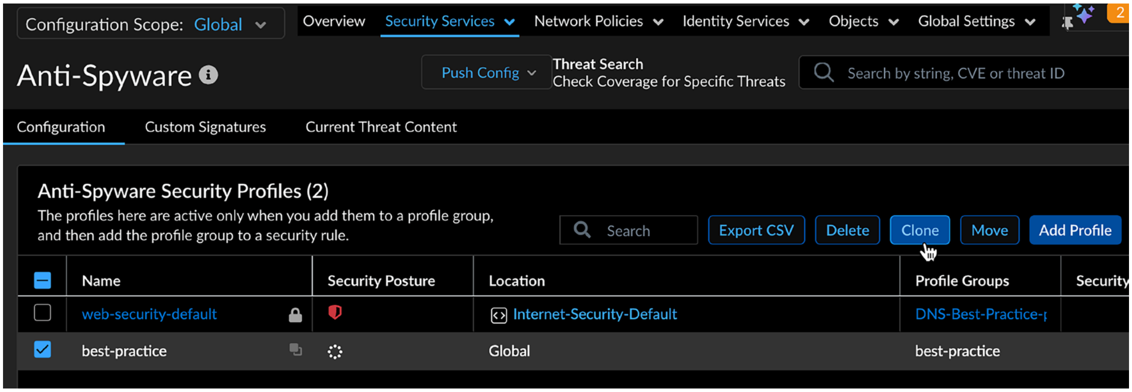


Figure 8.26: Unlocked default best-practice profile

You can, of course, make different profiles and different profile groups to suit your needs.

Access management

We can control who has access to certain sections of a scope by setting access management in **Manage** | **Access Control** | **Scope Management**:

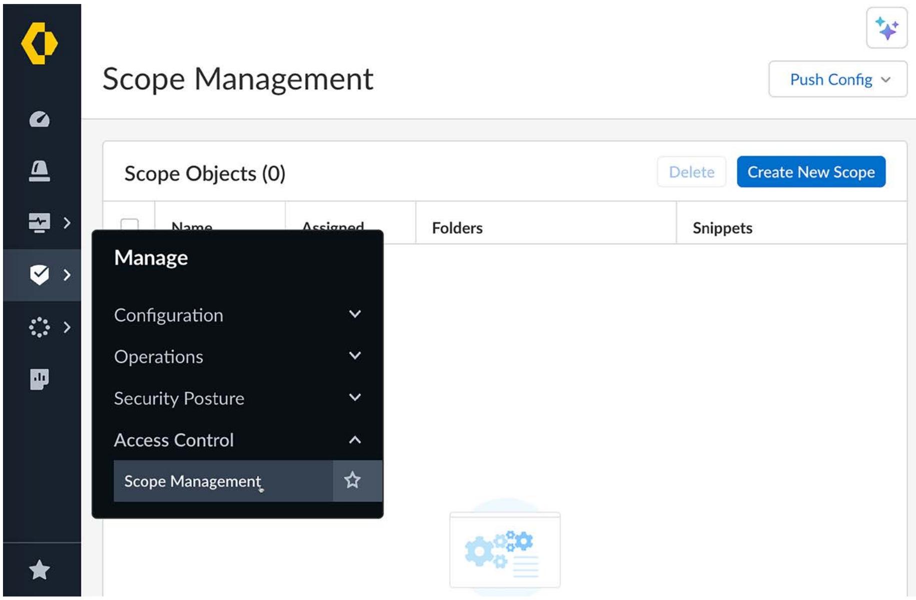


Figure 8.27: Scope Management

Click **Create New Scope** to start a new scope management profile. Provide a descriptive name and select all the configuration scopes you want to include in this profile:

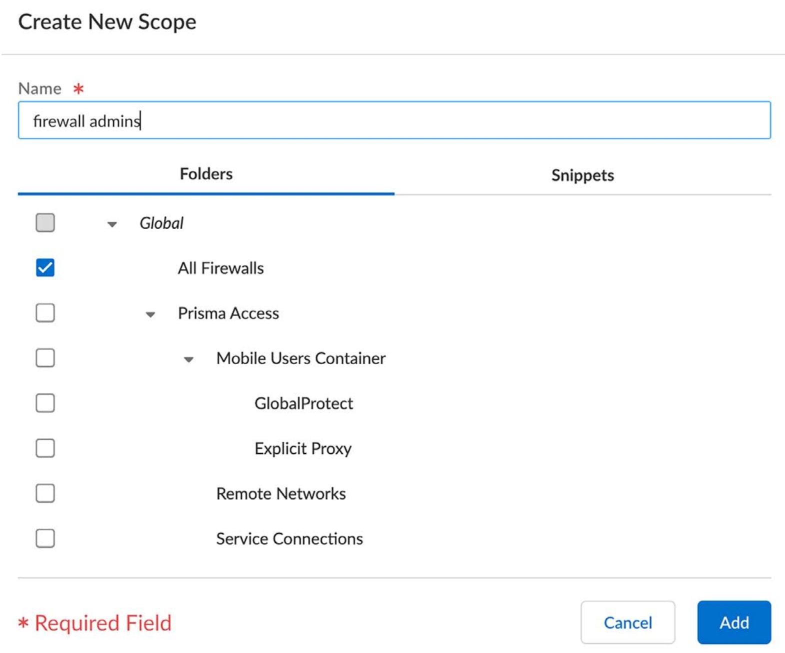


Figure 8.28: Create New Scope

Once the profile is created, you can add users and assign them a role. This will determine what level of access users get to a specific section of the configuration scope:

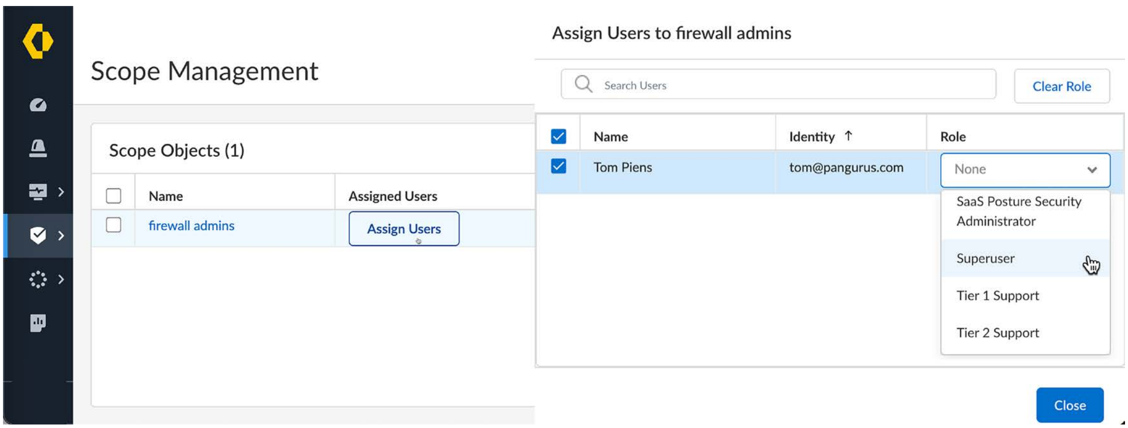


Figure 8.29: Adding users to a scope

Before we can get started pushing configuration to devices, we first need to associate them to SCM. We look at how to do this next.

Associating devices to Strata Cloud Manager

Although we first need to associate devices to SCM before pushing configuration to them, note that the configuration can be created before the devices are associated, however, as we've done in this chapter.

To associate devices:

1. Attach the devices to your tenant via the hub (`apps.paloaltonetworks.com`). To be able to do so, make sure your firewalls are registered in the support portal (`support.paloaltonetworks.com`) and that you have assigned all the licenses to them.

   Next, log in to the hub, and from **Common Services**, select **Device Associations**:

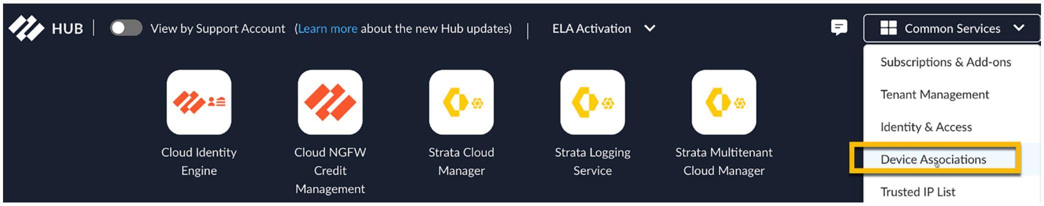


Figure 8.30: Device Associations

2. Make sure you select the right tenant, if there are multiple, and click **Add Devices**:

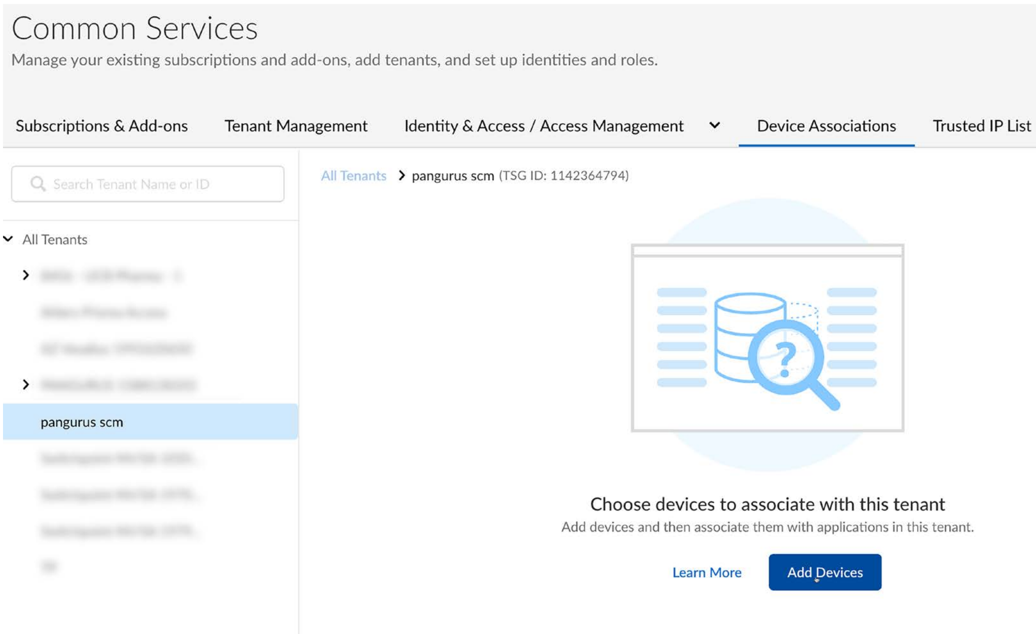


Figure 8.31: Add Devices

3. Select all the serial numbers that should be managed:

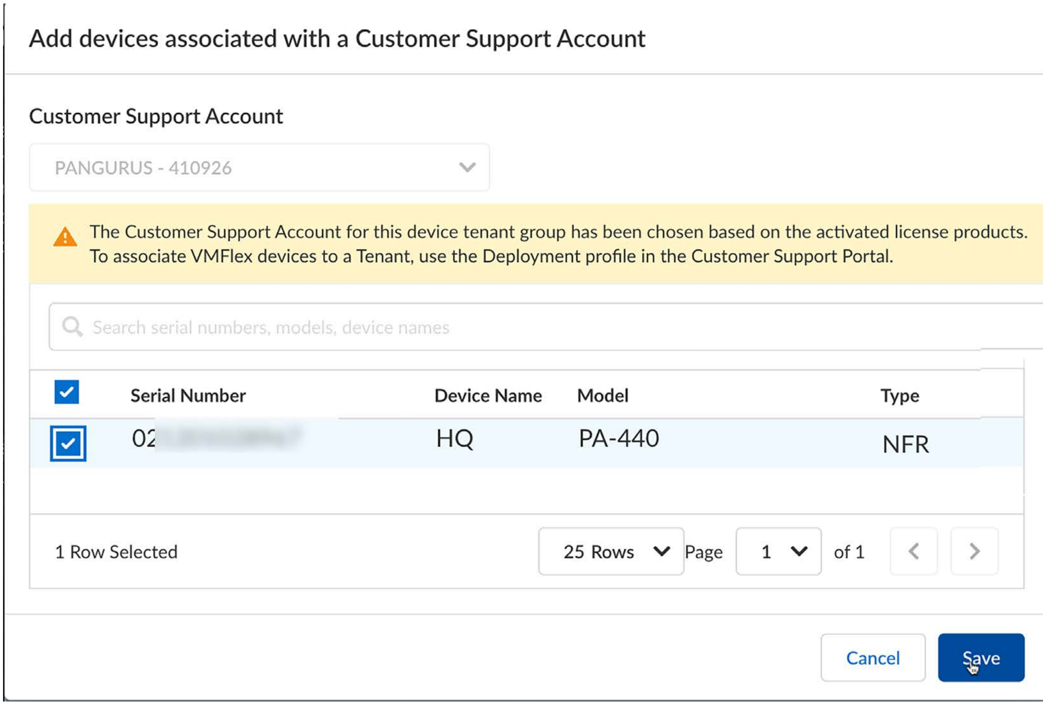


Figure 8.32: Select devices

4. If you have additional products, such as SLS, you can associate the device(s) to these services after they've been associated with the tenant by clicking **Associate Products:**

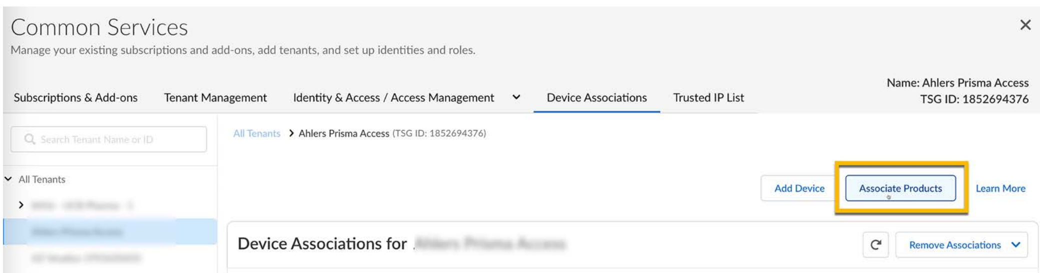


Figure 8.33: Associate Products

5. Then, select the product and serial numbers and click **Save:**

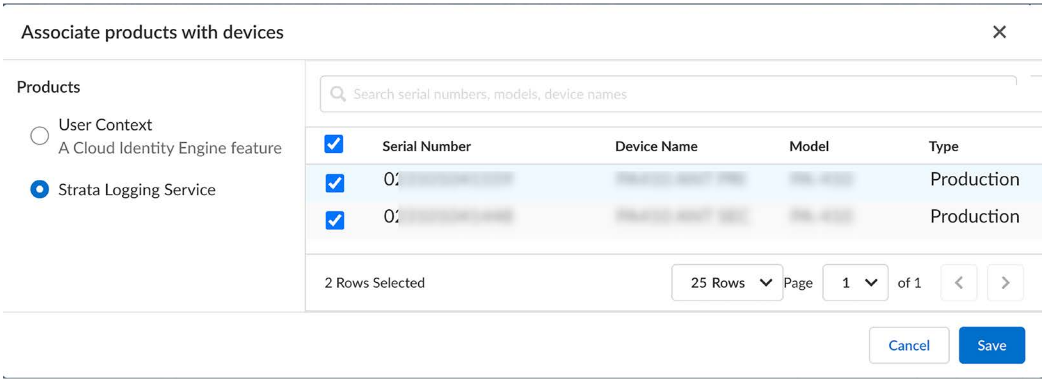


Figure 8.34: Associate products with devices

6. Next, make sure the firewall is set to be managed by the Cloud Service rather than Panorama. Go to **Device | Setup | Management** and, in **Panorama Settings**, switch the toggle from **Local Panorama** to **Cloud Service**. Commit the change to the firewall.

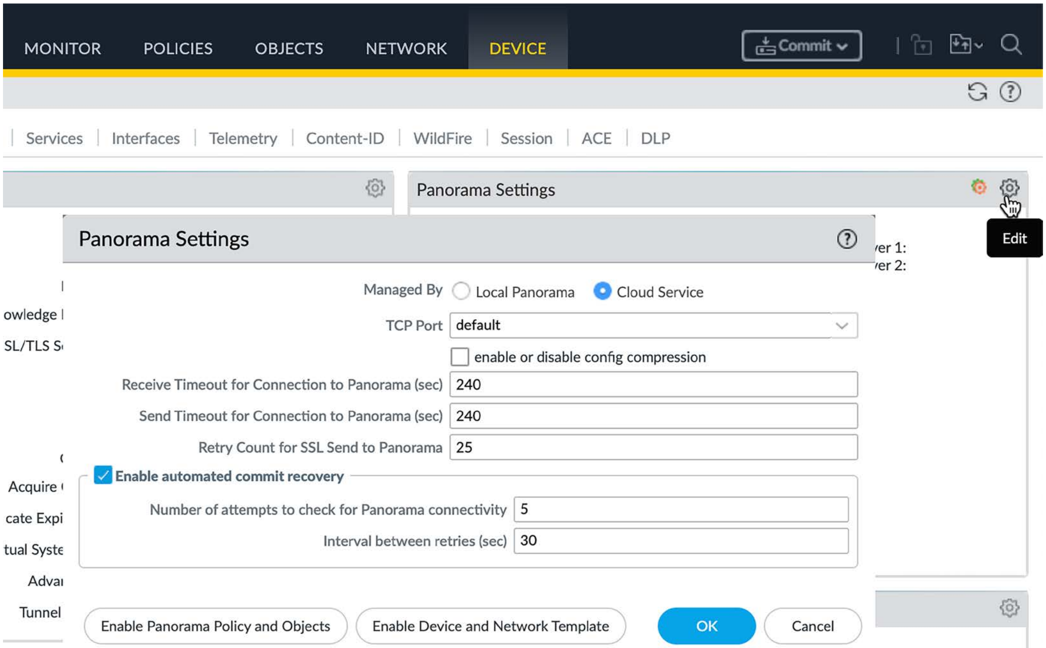


Figure 8.35: Firewall managed by Cloud Service

We can now onboard the firewall into SCM.

Managing devices and device configuration through Workflows

In the **SCM Workflows** tab, we can add and manage firewalls. (In Panorama, this would be the **Templates** and parts of the **Panorama** tabs.)

Before we add new firewalls, there are a few things we can do to facilitate the onboarding process by creating **Device Labels** and **Device Onboarding** profiles.

Device Onboarding

In **Workflows | NGFW Setup | Device Onboarding | Device Labels**, we can create labels and label groups. Label groups can be used to assign **Zero Touch Provisioning (ZTP)** (which allows "fresh" firewalls to be auto-provisioned with basic configuration the moment they come online) and automatic onboarding of devices. This allows you to quickly deploy new devices in remote offices where there are no local technical staff. All new devices are automatically in ZTP mode, so they can automatically be onboarded into your SCM.

Follow these steps to prepare to onboard new out-of-the-box firewalls:

1. Register the device serial number to the support portal, and assign all licenses.
2. Associate them to your SCM tenant, as we saw in the previous section.
3. In **Workflows | NGFW Setup | Device Onboarding | Device Labels**, set up a new label group, ensuring the ZTP **Enable** checkbox is ticked.
4. Add a new label.

   These steps should look similar to the screenshot below:

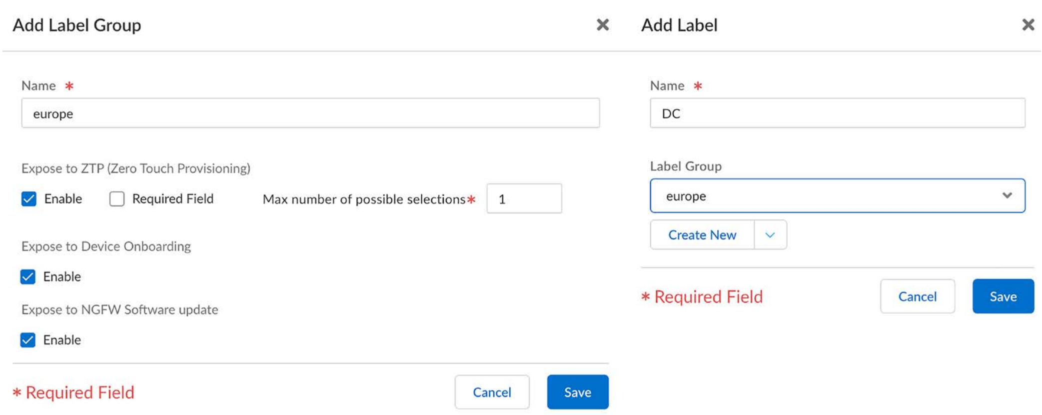


Figure 8.36: Adding label groups and labels

5. In **Workflows | NGFW Setup | Device Onboarding | Device Onboarding**, create a new onboarding profile and add the tag you just created in **Labels**.
   - You can set **Device S/N** (Serial Number) to **Any**, or a specific pattern to determine which firewalls will be captured by this profile.
   - You can also select which folder firewalls caught by this profile will be placed in – in the below case, the folder is **All Firewalls**, which is the default folder.
   - You can set a **Target OS Version** if you want the firewall to be upgraded to the target OS immediately when it joins SCM.

- If you have a Palo Alto SD-WAN deployment, you can join the firewall to it by enabling **VPN Onboarding.**
- The firewall can also be attached to the CIE attached to the SCM tenant, if there is one active, via **User Context Onboarding**.

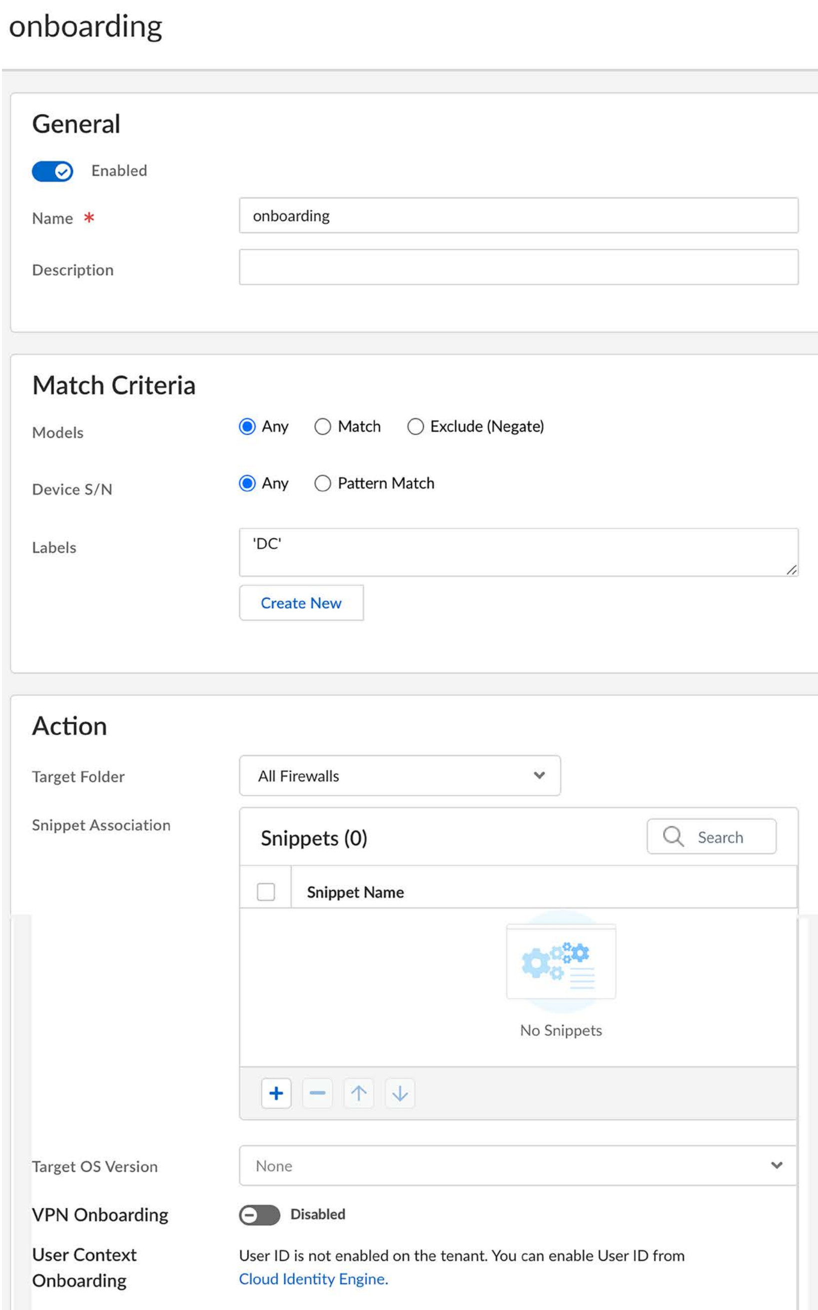


Figure 8.37: Onboarding profile

Folder Management

In **Workflows | NGFW Setup | Folder Management**, we can create device folders, sort of like device groups or templates in Panorama. We saw these earlier in this chapter as the configuration scope hierarchy in the *Security rules* subsection. These folders allow us to group similar devices and set up inheritances.

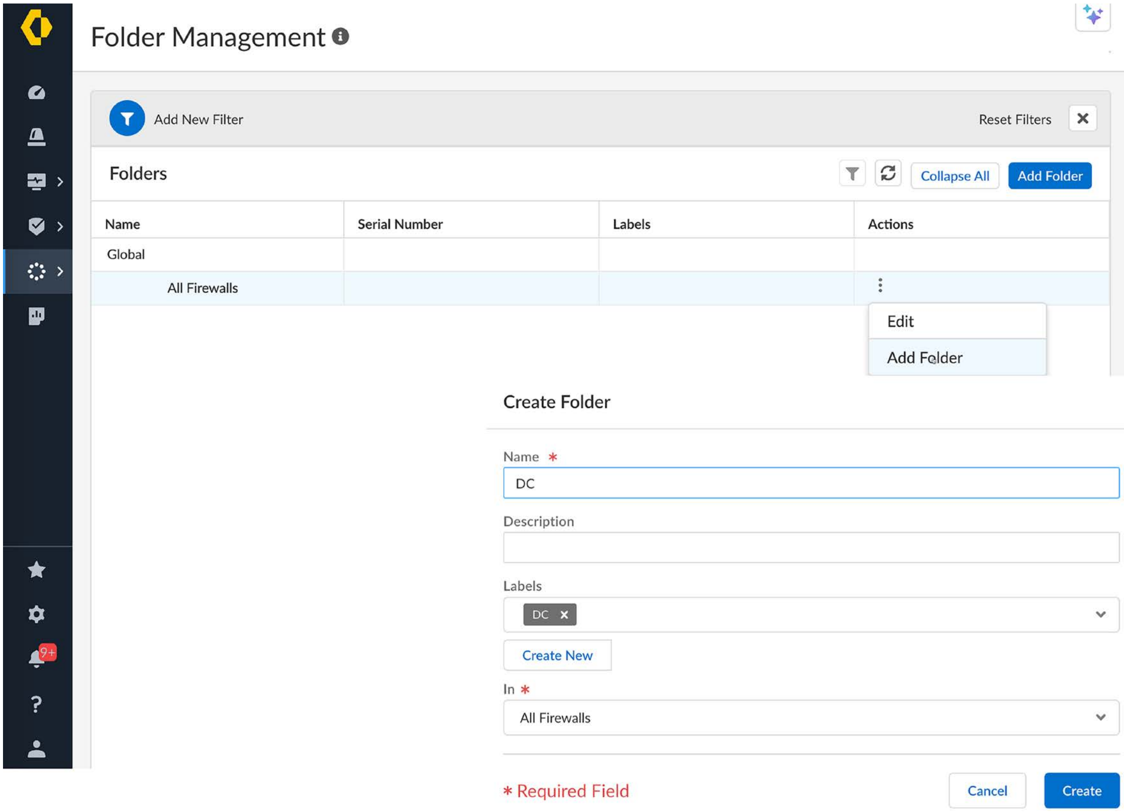


Figure 8.38: Folder Management

Global would be where all shared configuration goes, such as embargo rules or administrative access. **All Firewalls** or **Prisma Access** would be a top layer folder for either all NGFW or Prisma Access folders. Additional regional folders can be added and then individual clusters or locations could be added.

Device Management

In **Workflows | NGFW Setup | Device Management**, we can review onboarded devices.

To manually add new firewalls, select the **Available Devices** tab, select the available devices (remember, they first need to be associated to the tenant), and then click **Move to Cloud Management**:

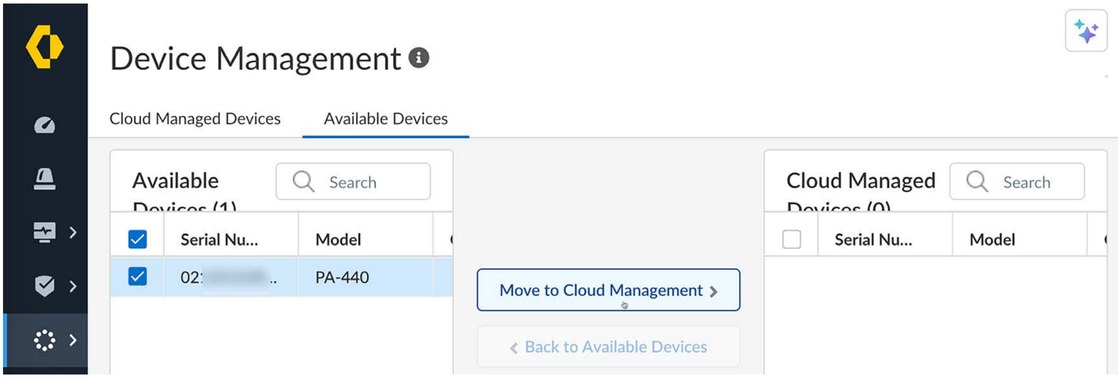


Figure 8.39: Manually adding new firewalls

Then, select the appropriate label (if any) and click **Move to Cloud Management** again:

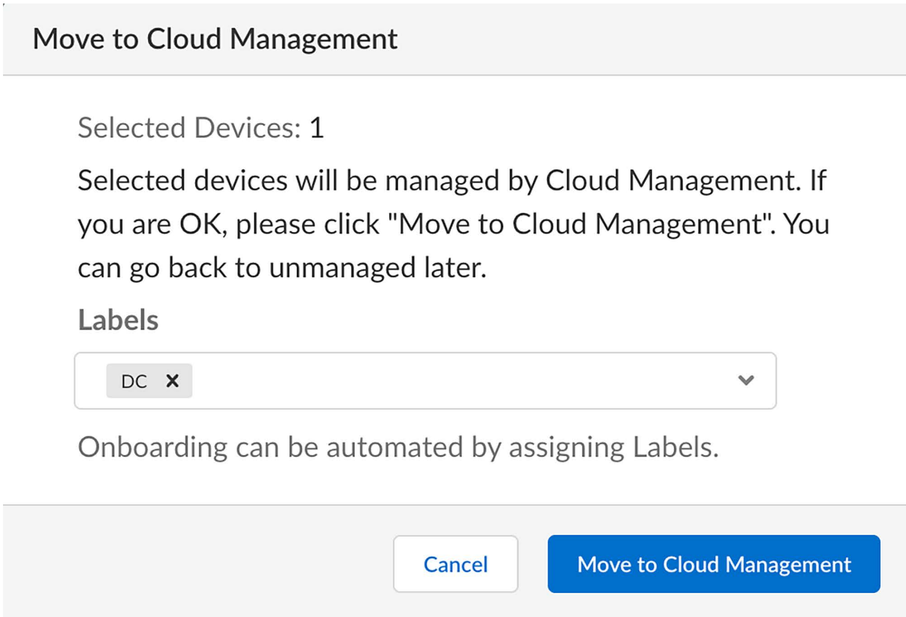


Figure 8.40: Move to Cloud Management

The device will now show up in the Cloud Managed Devices tab:

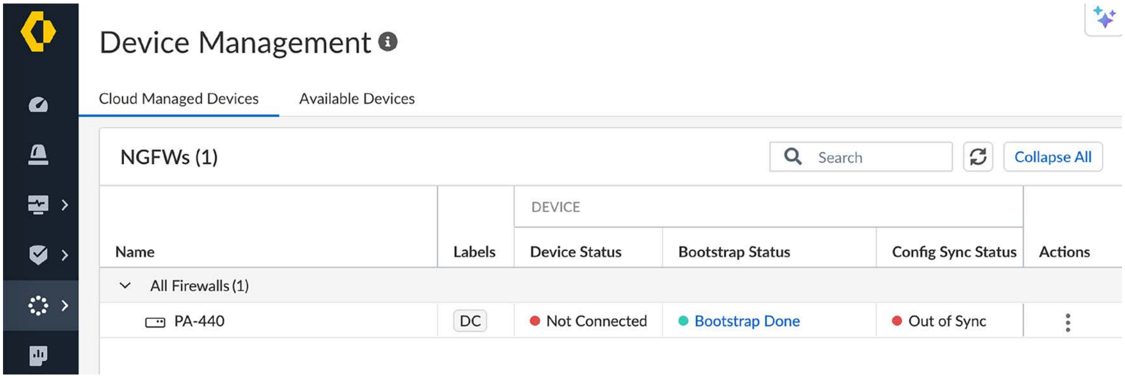


Figure 8.41: Cloud Managed Devices

Lastly, the device itself needs to be set to be managed by SCM (device local configuration is needed to let it connect to the cloud and be onboarded into SCM). In **Device | Setup | Management | Panorama Settings** in the firewall web interface, switch **Local Panorama** to **Cloud Service**.

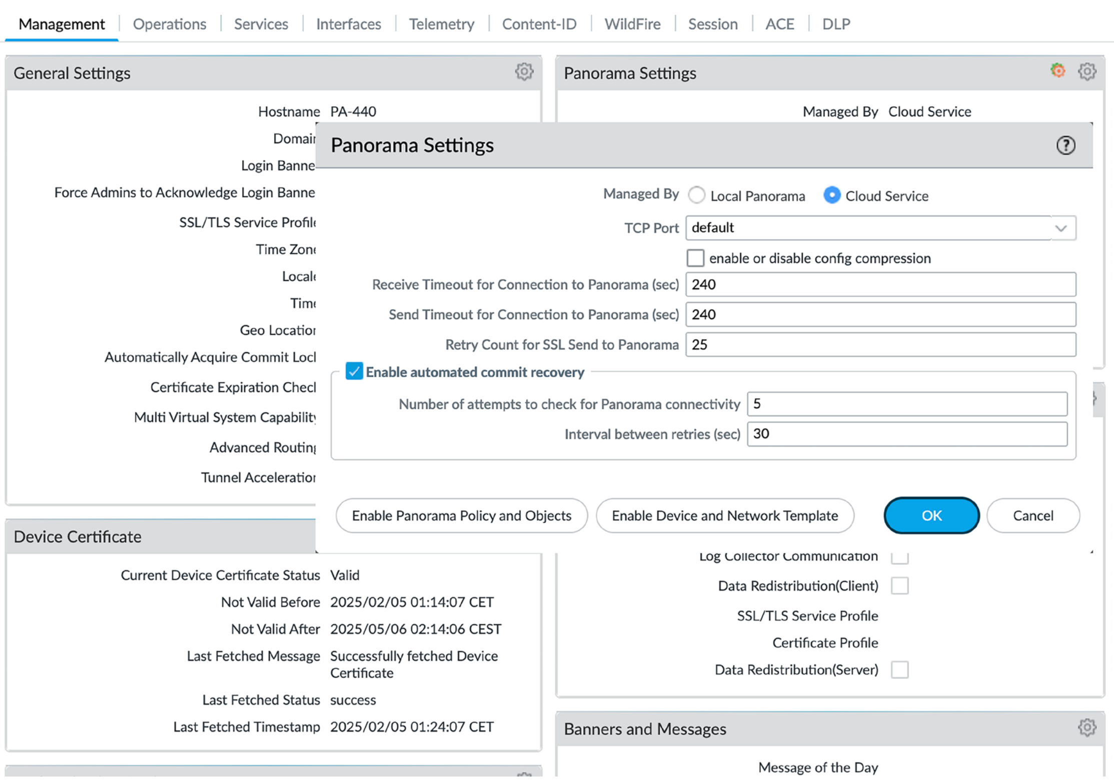


Figure 8.42: Setting the device to be cloud service managed

Make sure the firewall is allowed to make outbound connections on ports 443 and 3978, and that the device certificate is valid. The firewall will now show up as **Connected** in the SCM **Device Management.**

Device Settings and Global Settings

Now that devices are onboarded, additional tabs become available under **Manage | Configuration | NGFW and Prisma Access** if you select the **All Firewalls** configuration scope or one of its children:

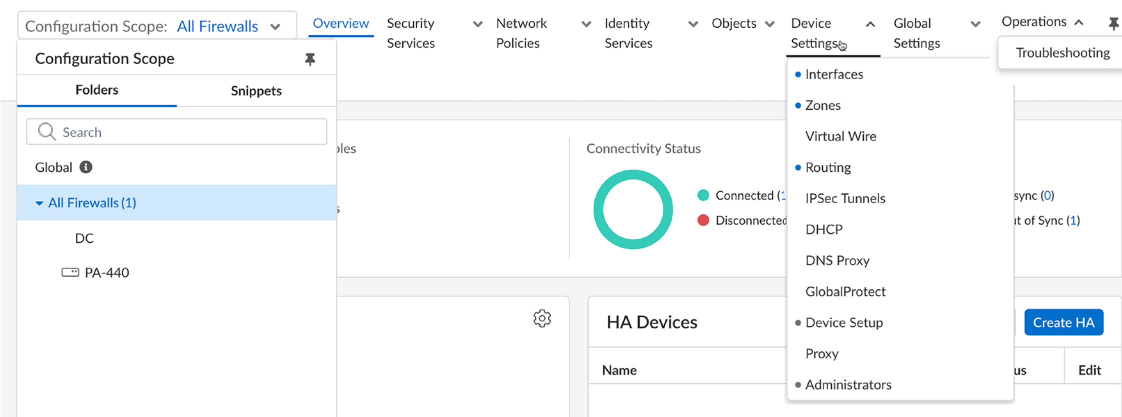


Figure 8.43: Device Settings and Global Settings

Under the **Device Settings** tab, we get the following options:

- **Interfaces** allows you to add interfaces and set IP addresses.
- **Zones** lets you configure the zones you'd assign to interfaces or use in security policies.
- **Virtual Wire** lets you bundle two interfaces in a virtual wire configuration.
- **Routing** allows you to configure the virtual router on managed devices.
- **IPSec Tunnels** lets you configure VPN tunnels. This section bundles the **IKE Gateways** and **IPSec Tunnels** you'd see separately in Panorama or the firewall web interface in one single profile.
- **DHCP** lets you configure the DHCP server of the DHCP relay.
- **DNS Proxy** contains the DNS proxy profiles you can deploy to firewalls.
- **GlobalProtect** is where you can configure both GP portals and GP gateways deployed to firewalls.
- **Device Setup** contains the device configuration you would normally find under the **Device** tab, such as service routes, admin authentication, management interface settings, dynamic updates schedules, session settings, TCP settings, and so on.
- **Proxy** is where you can configure the explicit or transparent proxy configuration.
- **Administration** is where you can configure local administrator accounts and password profiles.

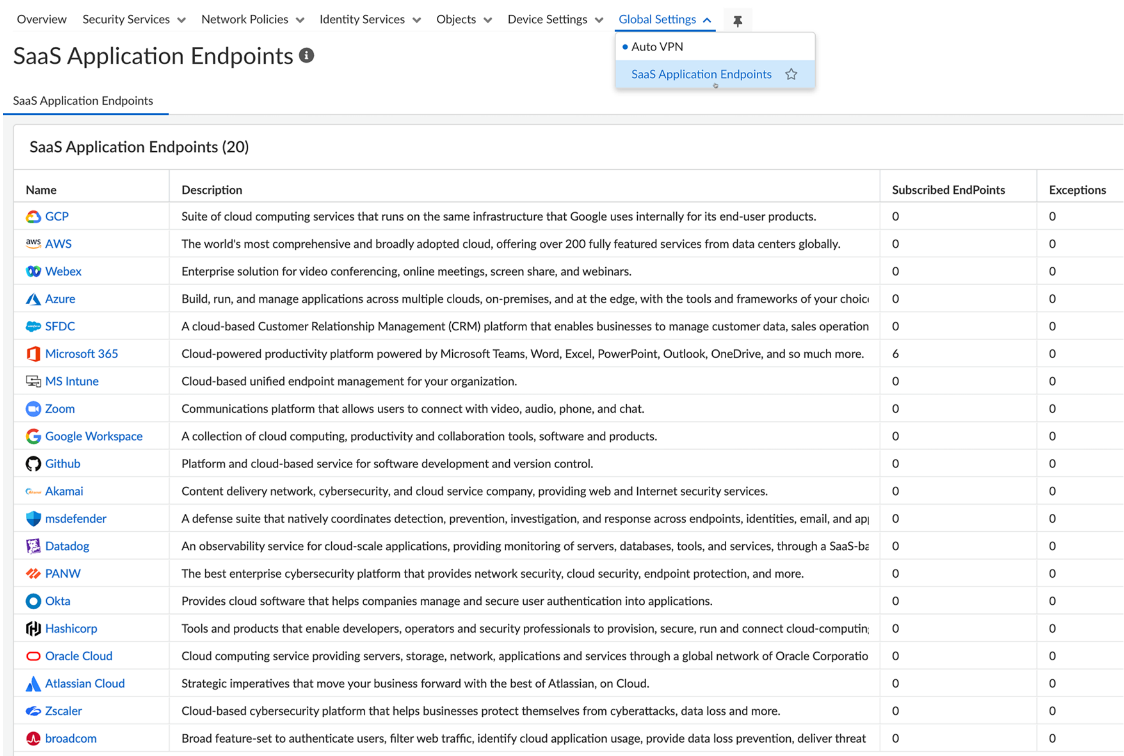
All these contain the same options you have available under the **Network** and **Device** tabs on the Panorama or firewall local web interface.

Under the **Operations** tab, you have the following options:

- **Troubleshooting** is similar to the troubleshooting options found on the firewall web interface. You can test several functions on the firewall to ensure things work as expected. The following tests are available:
    - **DNS Proxy** shows the DNS proxy cache of the firewall.
    - **User Group** shows the user groups on the firewall.
    - **User IP** shows the user-to-IP mapping on the firewall.
    - **Routing** shows the routing table on the firewall.
    - **Dynamic Address Group** shows all the DAGs on the firewall.
    - **Dynamic User Group** shows all the DUGs on the firewall.
    - **EDL** shows all the external dynamic lists on the firewall.
    - **NAT** fetches the running IP pool from the firewall.
    - **Session Browser** fetches the active sessions from the firewall. You can add filters to limit the output to just the information you're looking for.

In the **Global Settings** tab, the following options become available:

- **Auto VPN** allows you to set up SD-WAN hub-and-spoke configuration so newly added devices immediately join your SD-WAN fabric.
- **SaaS Application Endpoints** lets you onboard sanctioned SaaS applications:



Name	Description	Subscribed EndPoints	Exceptions
GCP	Suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products.	0	0
AWS	The world's most comprehensive and broadly adopted cloud, offering over 200 fully featured services from data centers globally.	0	0
Webex	Enterprise solution for video conferencing, online meetings, screen share, and webinars.	0	0
Azure	Build, run, and manage applications across multiple clouds, on-premises, and at the edge, with the tools and frameworks of your choic	0	0
SFDC	A cloud-based Customer Relationship Management (CRM) platform that enables businesses to manage customer data, sales operation	0	0
Microsoft 365	Cloud-powered productivity platform powered by Microsoft Teams, Word, Excel, PowerPoint, Outlook, OneDrive, and so much more.	6	0
MS Intune	Cloud-based unified endpoint management for your organization.	0	0
Zoom	Communications platform that allows users to connect with video, audio, phone, and chat.	0	0
Google Workspace	A collection of cloud computing, productivity and collaboration tools, software and products.	0	0
Github	Platform and cloud-based service for software development and version control.	0	0
Akamai	Content delivery network, cybersecurity, and cloud service company, providing web and Internet security services.	0	0
msdefender	A defense suite that natively coordinates detection, prevention, investigation, and response across endpoints, identities, email, and ap	0	0
Datadog	An observability service for cloud-scale applications, providing monitoring of servers, databases, tools, and services, through a SaaS-ba	0	0
PANW	The best enterprise cybersecurity platform that provides network security, cloud security, endpoint protection, and more.	0	0
Okta	Provides cloud software that helps companies manage and secure user authentication into applications.	0	0
Hashicorp	Tools and products that enable developers, operators and security professionals to provision, secure, run and connect cloud-computin	0	0
Oracle Cloud	Cloud computing service providing servers, storage, network, applications and services through a global network of Oracle Corporatio	0	0
Atlassian Cloud	Strategic imperatives that move your business forward with the best of Atlassian, on Cloud.	0	0
Zscaler	Cloud-based cybersecurity platform that helps businesses protect themselves from cyberattacks, data loss and more.	0	0
broadcom	Broad feature-set to authenticate users, filter web traffic, identify cloud application usage, provide data loss prevention, deliver threat	0	0

Figure 8.44: SaaS Application Endpoints

Variables can be created in all of the above sections where variables are typically used (for example, in interfaces):

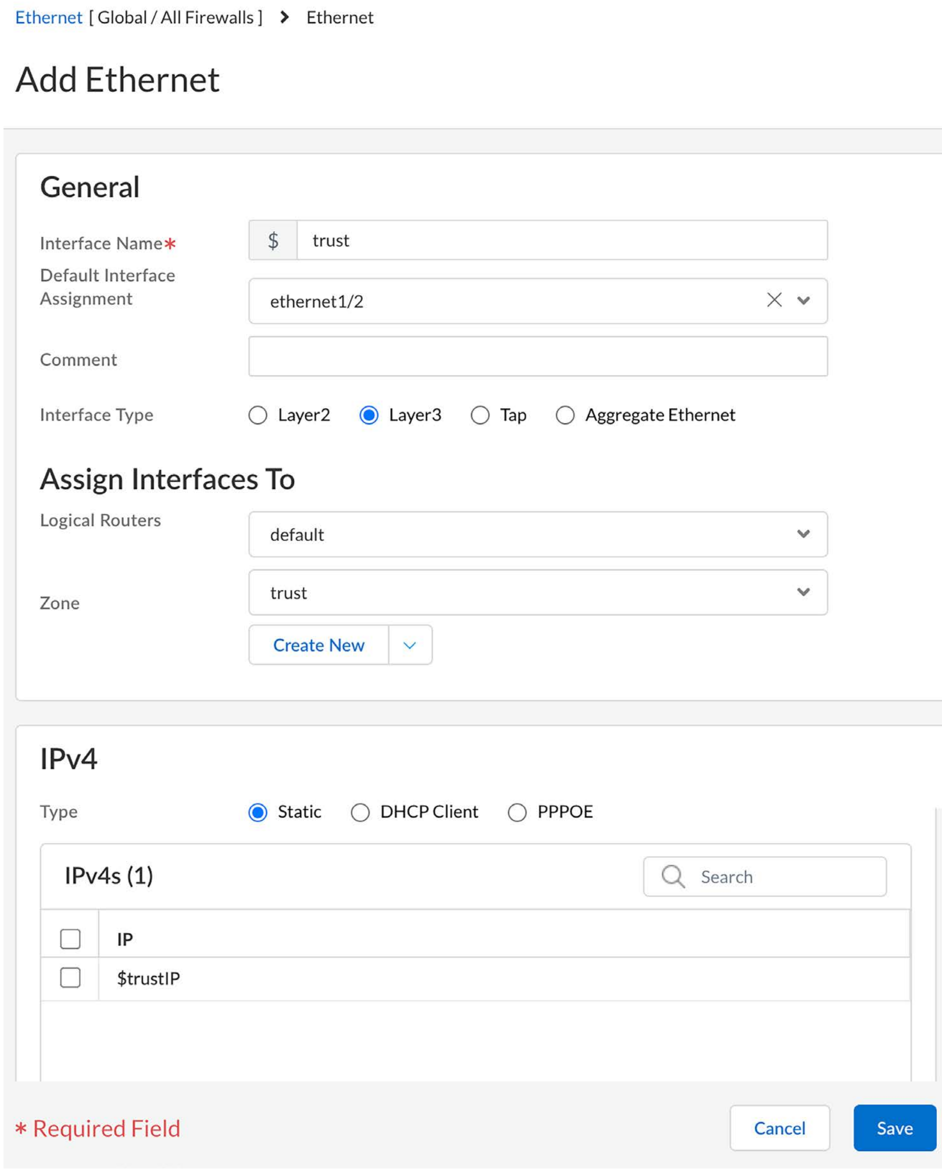


Figure 8.45: Adding a variable in interfaces

They can also be managed from **Overview** (make sure you're in the **All Firewalls** folder):

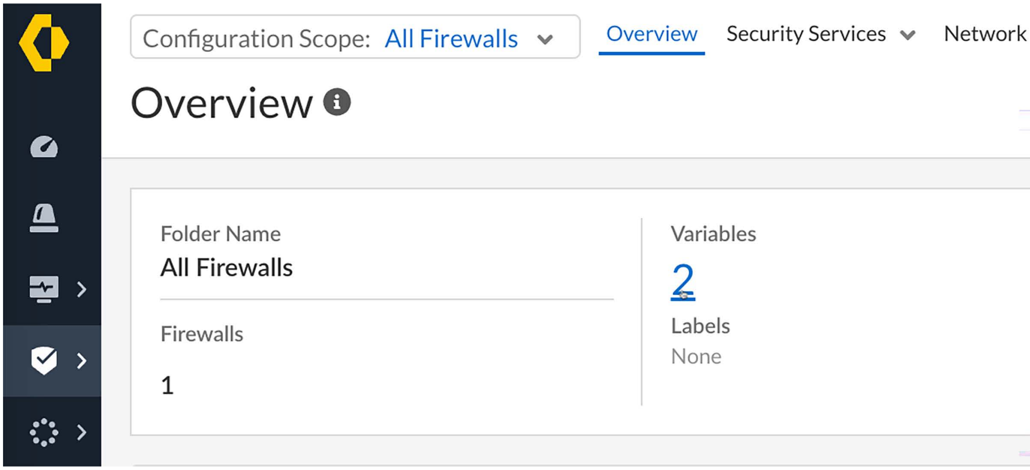


Figure 8.46: Variables

Exploring dashboards

Now that we have the ability to deploy configuration to firewalls, we can start taking inventory of how our deployments are faring.

In the **Dashboards** page (the very first page you see when you open SCM), we can get an overview of all sorts of things, including device health, but we can also activate additional dashboards that give us more insights. To do this, click the **More Dashboards** button at the top right, as you can see in the following screenshot, to select and activate additional dashboards.

One additional dashboard you can add, as an example, is **Security Posture Insights**, which gives us a time-lapse view of the security posture of our devices. If a new configuration is introduced that improves or decreases our overall posture, it will be reported so appropriate action can be taken.

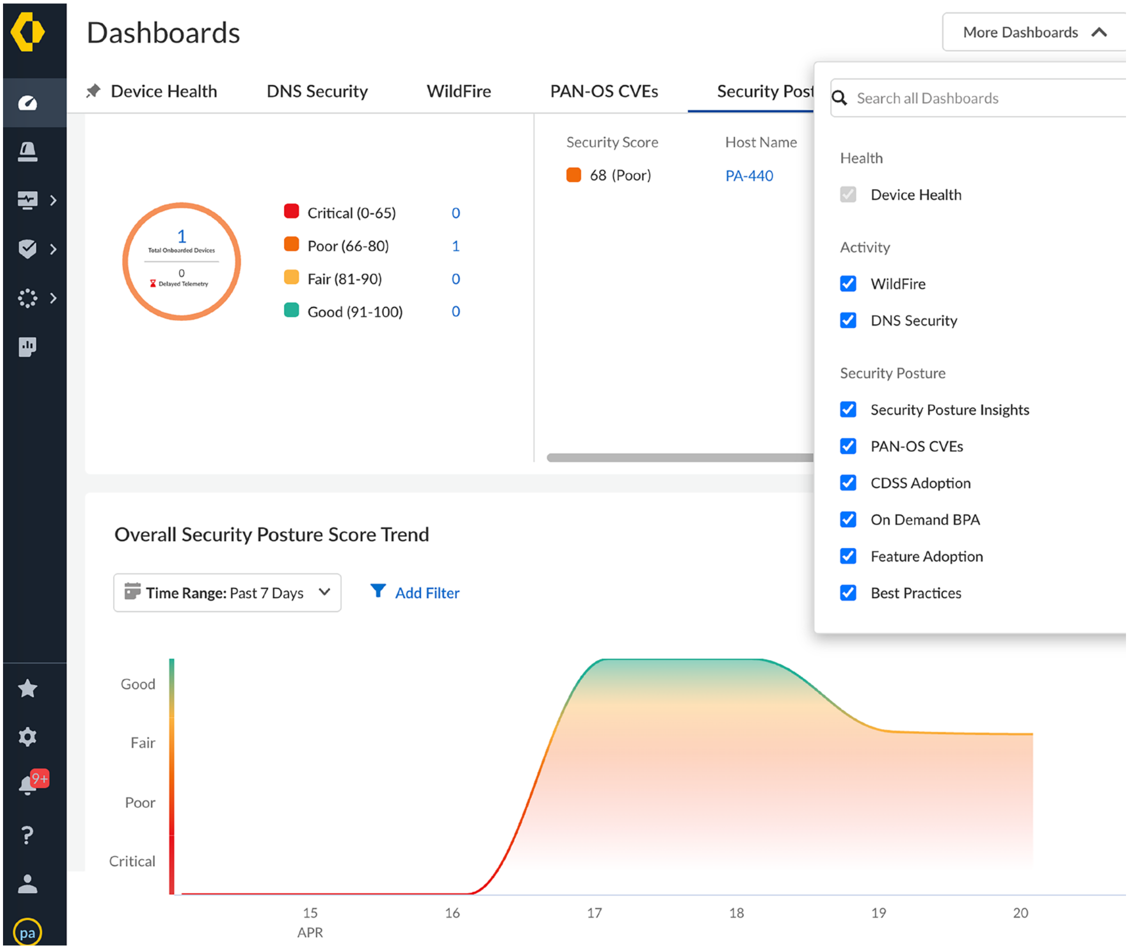


Figure 8.47: Security Posture Insights

All the dashboards give you an immediate overview of what's actively going on in your environment and allow you to drill down into actionable data. **DNS Security**, for example, helpfully gives you a quick overview of all the malicious and potentially concerning DNS lookups that were detected on your network or from your remote users:

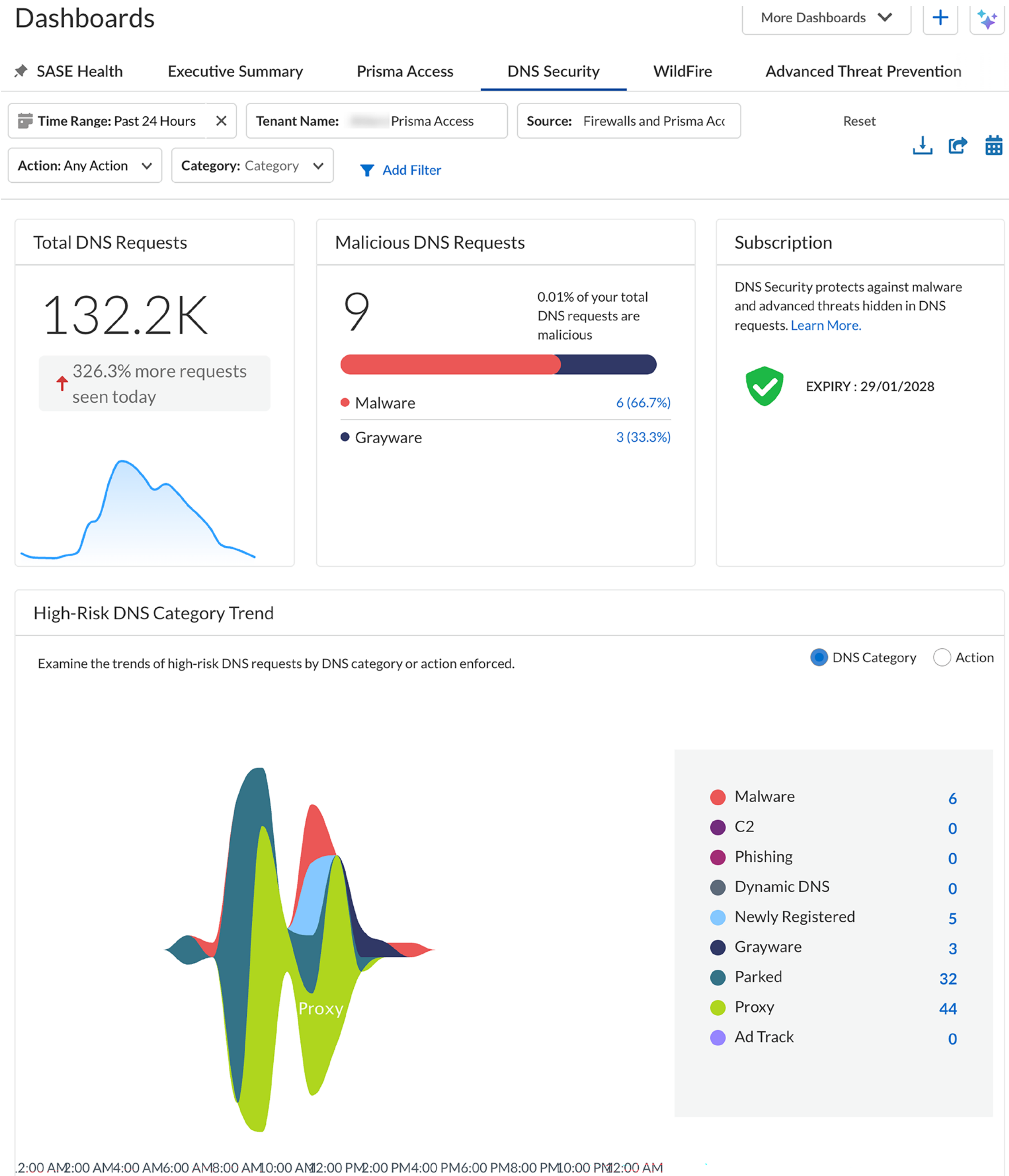


Figure 8.48: DNS Security dashboard

Best Practices and **On-Demand Best Practices** (depending on your license; Pro licenses will have even more options) give you an immediate overview of your configuration compliance and best practices hits and misses so you can easily keep track of any shortcomings or improvements that can be made to upscale your security posture:

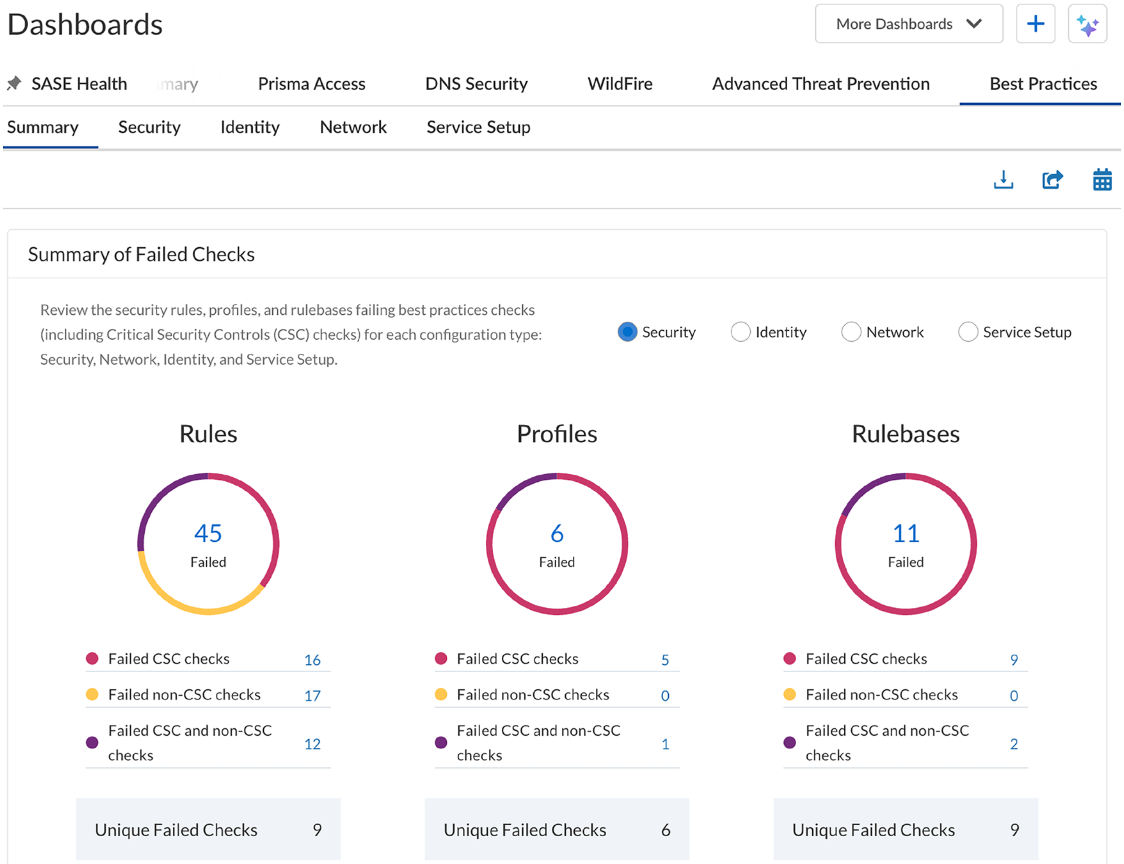


Figure 8.49: Best Practices dashboard

The **Monitor** tab also provides additional insights that help improve or remediate any configuration shortcomings or new threats that need to be addressed:

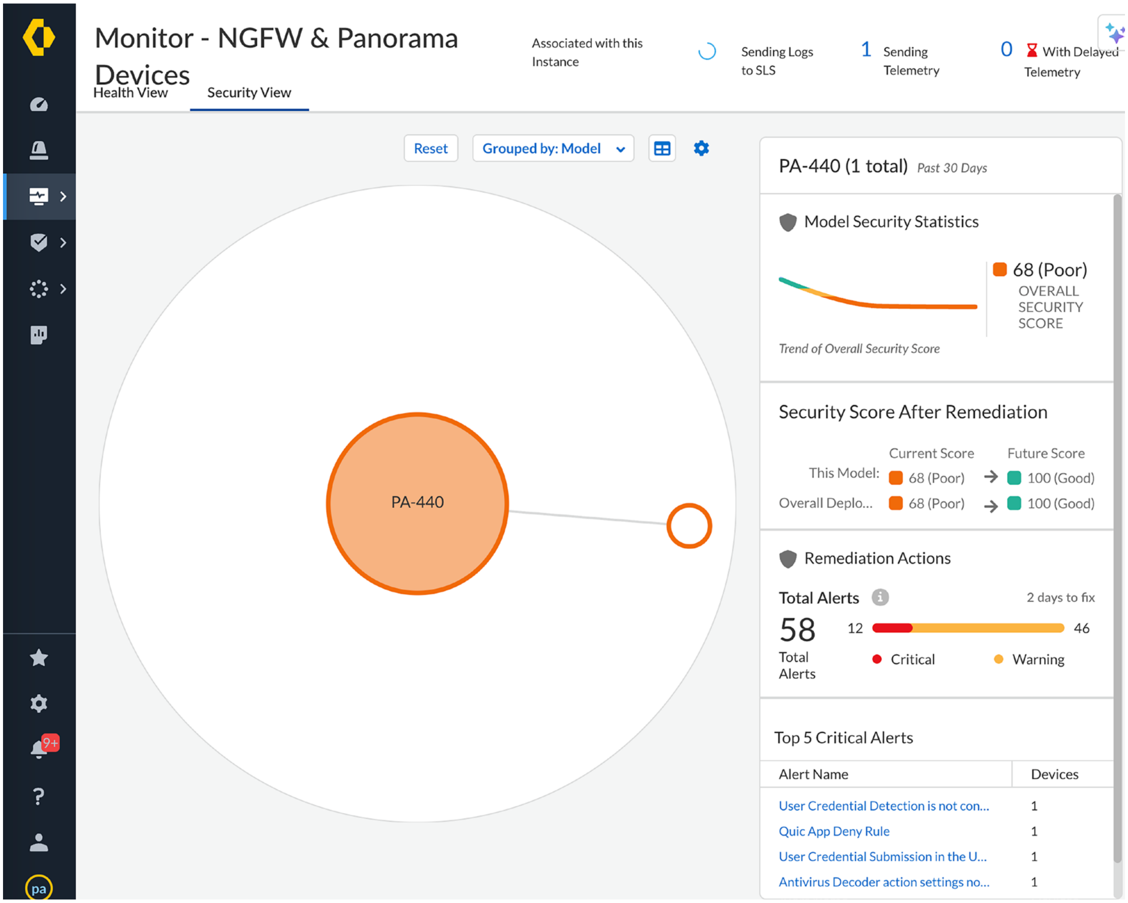


Figure 8.50: Device monitoring

With the information we covered in this chapter, you should be able to deploy your own SCM and onboard firewalls for simplified and accessible central management. The main advantage of SCM is that you can manage and provide management access from anywhere without needing to set up complex out-of-band access, retaining control and visibility even if local infrastructure is difficult to access from outside.

Summary

In this chapter, you learned about the Strata Cloud Manager central management platform and how it can be leveraged to make managing groups of firewalls much less complex. You learned how device configuration scope and folders can be used to simplify and make configuration consistent across all of your managed devices.

In the next chapter, we will review the best practices for upgrading firewalls and Panorama.

9

Upgrading Firewalls and Panorama

Just like any other operating system, bugs are sometimes found in PAN-OS, which could cause all kinds of issues. These bugs need to be fixed, and so update packages, called **maintenance releases**, that customers can install to improve the resilience and stability of their systems are made available. New features are also introduced through new major releases of the operating system.

In this chapter, we will learn how to upgrade firewalls, Panorama, and **High Availability** (**HA**) pairs. We will review what steps need to be taken to prepare for an upgrade and how to ensure continuity throughout the upgrade process, as well as any limitations that may apply, any issues that may arise, and the steps that need to be taken to upgrade.

In this chapter, we're going to cover the following main topics:

- Documenting key aspects
- Preparing for the upgrade
- The upgrade process (for standalone and HA firewalls, Panorama, and log collectors)
- The rollback procedure
- The downgrade procedure
- Special case for older hardware

Technical requirements

This chapter assumes that you have a working knowledge of testing system functionality, so you should be able to ascertain that everything is working normally after an upgrade has taken place.

If you are going to upgrade a cluster, you should first get comfortable with how it is configured before proceeding.

Documenting key aspects

Before you can start the upgrade procedure, you should first take some time to document the key aspects of the network surrounding the firewall or Panorama. This information will need to go into a test plan that you can execute immediately after you have performed the upgrade as you will need to quickly ascertain whether the device is up and running and passing traffic as expected. It is important to identify key production applications and, if possible, identify personnel who can assist in testing application functionality post-upgrade.

It may be prudent to make an upgrade checklist so that you don't forget any important caveats, as well as a contingency plan in case something goes wrong, which includes at which point fallback is required. Set this point so that you have plenty of time to troubleshoot for minor oversights, but not so long that it impacts the business. Arrange for an appropriately sized maintenance window beforehand. Here's a checklist to help you get things organized:

- Map out key application data flows
- Identify personnel who can assist in verifying application functionality
- Document the upgrade plan beforehand
- Document a contingency or rollback plan
- Ensure out-of-band connectivity is available to the device

Depending on the PAN-OS version that you start with, some steps in the upgrade path may require the installation of a minimum version content update, or there may be other considerations. Always check the release notes of every version you plan to upgrade to (`https://docs.paloaltonetworks.com/search.html#q=pan-os-release-notes`).

Now, let's take a look at some important considerations you need to be aware of before starting the upgrade process.

Upgrade considerations

PAN-OS comes in a major version (`X.y.z`), a feature release (`x.Y.z`), and a maintenance release (`x.y.Z`). On average, a new major version is released every year, with a feature release following about half a year later, containing some new and updated features. Both versions get their own maintenance releases, which mostly contain bugfixes.

Maintenance packages are usually released every 6 to 8 weeks, with occasional hotfix versions (`x.y.z-h*`) to address critical issues and system vulnerabilities sooner.

For example, 11.0.0 is the base version of major release 11. After several months of being publicly available, some bugs were found and several consecutive maintenance releases were made available to address them. By the time a new feature release, 11.1.0, was made available, the previous major version was already on maintenance release 11.0.3. By the time 11.2.0 was released, the previous releases were on 11.0.6 and 11.1.1.

Moving forward, all code trains will receive their own maintenance releases. Once 12.0 is made available, the previous major versions will still keep receiving updates until their respective end-of-life dates.

For the purposes of this chapter, I will refer to both major releases and feature releases as *major* if the intention is to upgrade or downgrade from one code train to the next.

Each major and feature release has a **base image**, which is the install medium for the whole release. This version always needs to be **downloaded** before later maintenance versions can be added. It does not usually need to be installed (see the *Special case for upgrading older hardware* section later in this chapter for an exception to this).

> In most circumstances, a release can be considered **mature** when it reaches a minimum maintenance release version of `x.y.4` or later. Carefully weigh the need to upgrade if `x.y.4` or a later version is not available. Also review any recent hotfixes that may merit upgrading sooner, or defer the upgrade for a little while longer.

If your environment has Panorama, you should plan to upgrade it first, as Panorama is **backward-compatible** with almost any version of PAN-OS running on the firewall. However, it should not support firewalls that are more than two maintenance versions higher than itself (that is, Panorama should always be upgraded first and should be on the highest PAN-OS version of the entire installation base before considering any other upgrades).

When upgrading HA pairs, upgrading one member to anything higher than two major versions over its peer could cause session-sync issues during failovers. If you want to ensure as little disruption on the network as possible during the upgrade process, consider upgrading the firewalls **in lockstep**, rather than upgrading one member by several versions before starting on its peer.

For example, while upgrading a cluster from PAN-OS 9.1 to 10.1, if one member is upgraded to PAN-OS 10.0, session sync will continue to work and the upgraded peer will fall into a non-functional state. If the lagging peer is rebooted, the non-functional peer will assume an active role with minimal impact on the network.

If one member is upgraded to PAN-OS 10.1 before the peer is upgraded and rebooted into PAN-OS 10.0 (still running on PAN-OS 9.1), the newly upgraded peer will become suspended and will not assume an active role once the peer is rebooted, causing severe network impact.

> When only one peer is upgraded in a cluster, the lowest PAN-OS version peer will become the active member of the cluster even if the upgraded peer has a lower priority, leaving the upgraded member in a non-functional state (faulty but participating in a forced passive capacity).

When upgrading to the latest maintenance release from an earlier maintenance version (for example, from 11.2.1 to 11.2.5), you do not need to install any intermediary maintenance versions unless explicitly indicated in the release notes.

Prior to 10.1, when upgrading across multiple major versions, you were required to upgrade to the next major version before moving on to the one after it; you could not skip a major version (for example, you should go from 9.1 to 10.0 to 10.1). It is wise to install the preferred maintenance release instead of the base, even for the "middle" major version, as this will prevent any bugs from making an appearance in the middle of an upgrade process.

Starting from 10.1, you can skip up to 3 software versions when upgrading.

Starting from 10.2, a simplified software upgrade feature has been introduced that allows you to automatically verify, download, and install any dependencies there may be before moving forward to the target version.

Next, we'll take a closer look at the steps you should take before starting an upgrade.

Upgrade path

Depending on your currently installed major version, and whether your firewall is standalone or in a HA cluster, the upgrade path may be different.

Starting from a major release prior to PAN-OS 10.1, upgrades must happen one major version at a time, i.e., 9.1 to 10.0 to 10.1, etc.

Starting from PAN-OS 10.1 upgrades can skip up to 3 versions so you can upgrade from 10.1 directly to 11.1, skipping 10.2 and 11.0. If you want to go to 11.2, you first need to upgrade to 10.2 and then skip to 11.2.

Beware: Skipping versions in an HA cluster will cause a disruption in traffic as session synchronization only works between firewalls that are no more than one major version apart.

Preparing for the upgrade

Before we get started on the upgrade process, there are a few precautions we should take to ensure we are properly prepared and have everything set so that the upgrade process goes smoothly:

1. Go to **Device** | **Setup** | **Operations** for an unmanaged firewall or **Panorama** | **Setup** | **Operations** for Panorama, then click **Save named configuration snapshot** and name the configuration file for the device name, date, and time (for example, `HQmember1-02032025-10h05.xml`).
2. Next, click on **Export named configuration snapshot** and save the file somewhere where you can find it if you need it.
3. If the firewall is managed by Panorama, click **Export Device State** and store the downloaded file safely. This file contains the runtime information of the firewall, including the Panorama-pushed device group and template configuration. If a recovery is needed, this file will help you get back up and running without needing to recombine the local and Panorama configuration.

If you have Panorama, you should already have the scheduled backup configured under **Panorama | Scheduled Backup**, but it doesn't hurt to have a fresh backup just in case.

4. Go to **Device | Dynamic Updates**, click on **Check Now**, and make sure the device has installed the latest content packages available to your system.
5. Some PAN-OS versions require a minimum version of content packages to be installed before the OS can be installed. If a newer content package version is available, download and install it before the PAN-OS upgrade takes place.
6. If your system is already on PAN-OS 10.2 or later, in **Device | Software** you can click the **Validate** button to check whether any requirements are missing. You must download the desired target version first.
7. In PAN-OS 11 and above, you can validate a target upgrade without downloading the image first. The output will immediately give you the option to download any requirements, including the base image, and the target release if it has not been downloaded already:

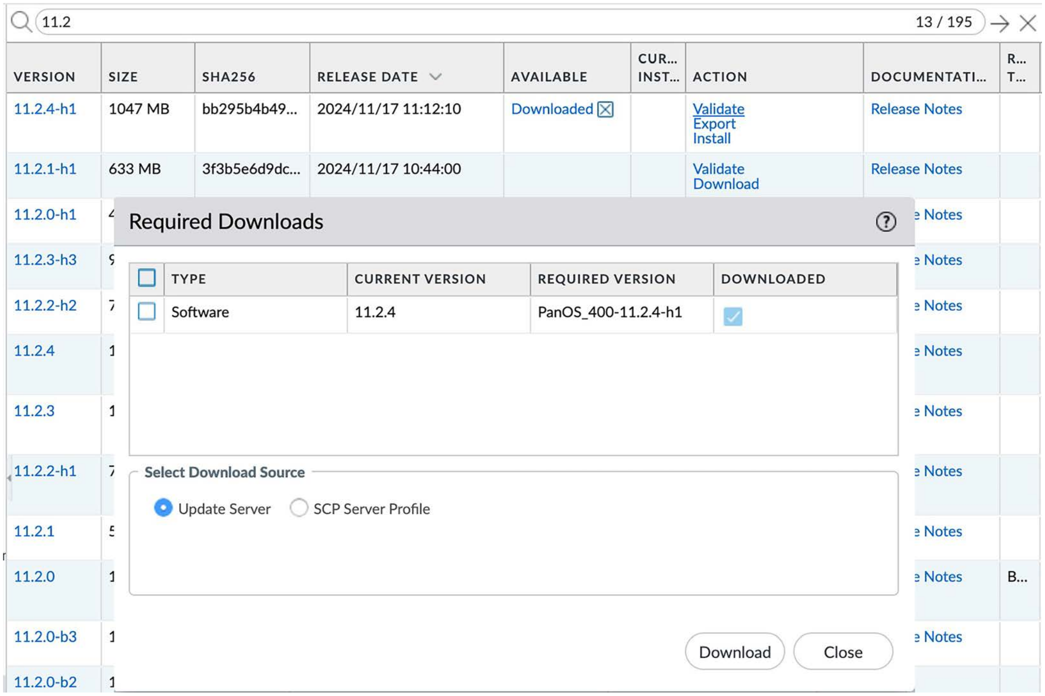


Figure 9.1: Validate upgrade path

If the device is running in HA mode, perform the validation on the peer as well.

8. For HA pairs that have **Device | High Availability | Election settings | Preemptive** enabled, disable it on the primary member. A preemption could cause unexpected automated failovers during the upgrade process, which you will want to prevent (preemption must be enabled on both members to function properly, so disabling one member is sufficient to break the mechanism).

9. Determine which maintenance version you should reach by the end of the upgrade process by reviewing the security advisories at `https://security.paloaltonetworks.com/`. Take note of which versions are marked as **preferred** at `https://live.paloaltonetworks.com/t5/Customer-Resources/Support-PAN-OS-Software-Release-Guidance/ta-p/258304` as I will refer to them as the **preferred maintenance release**.
10. To save time, download all the required base images and preferred maintenance versions needed for the upgrade process to the device from the **Device** | **Software** or **Panorama** | **Software** pages. When upgrading manually, store them in a local repository by going to `https://support.paloaltonetworks.com` then **Update** | **Software Updates**. If you intend to skip ahead by more than one major version and you are currently on a version lower than PAN-OS 10.1, you may not be able to download the latest code train directly onto the device as the software manager may not be able to understand these software packages. You can download those versions on local storage or wait until after the first stage to then download them from the update server.

From the CLI, you can use the following commands to refresh the available software repository, download, and eventually install the PAN-OS images:

```
> request system software check
> request system software download version x.y.z
> request system software install version x.y.z
```

It is generally a good idea, if you are not already using the latest version of your current code train, to install and reboot the latest version in your current PAN-OS environment before moving on to the next major release. Follow these steps to prepare yourself for the upgrade event:

1. Download the **preferred** version of the currently installed major or feature release.
2. Download the base image to the next major release you want to go to.
3. Download the preferred maintenance release version of the major release.
4. Prepare your maintenance window(s)—schedule to upgrade Panorama (both members of the cluster) first and then schedule the firewalls in another maintenance window. This will give you more time to focus on a single objective and will make troubleshooting, should anything unexpected happen, easier since you will only need to focus on one area.

> Make sure you allow plenty of time for the upgrade to complete, check the connectivity, and troubleshoot and roll back if needed, even if the upgrade itself is not expected to take long.

5. When the installation is complete, you will need to reboot. After the reboot, it may take several minutes for the management server to return to full functionality. The system comes back online in stages, so there will be a period of time where you are able to reach a login prompt but it will appear as if your password is incorrect. If this is the case, your reboot may simply need a few more minutes to fully set up all of its services. If you do manage to log in but the usual prompt is not visible, just give it a few more minutes—**don't panic** and **do *not* reboot**.

6. Once you are logged in after the upgrade, the system will need to commit its configuration to the data plane and perform some post-upgrade jobs. These are performed during the `AutoCommit` process. Once the `AutoCommit` process is complete, indicated by `FIN OK`, the system will be up and running. Track the progress with the following command:

```
> show jobs all
Enqueued Dequeued ID PositionInQ Type Status Result Completed
--------------------------------------------------------------
2024/12/31 23:01:33 23:01:33 1 AutoCom FIN OK 23:02:04
```

   Some devices may have two consecutive `AutoCommit` cycles; the first one is a regular `AutoCommit` and the second is used to synchronize the `idmgr` process between HA devices. `idmgr` maintains IDs on objects, network elements, and policies on the firewall. These IDs need to match for both members for the session failover to work flawlessly. You can verify whether `idmgr` is synced with the following command:

```
> debug device-server dump idmgr high-availability state
```

7. You can also verify whether the system is ready to process traffic with the following command:

```
> show chassis-ready
```

8. From the CLI, run the following command and take note of any deviating settings that need to be verified and potentially reset post-upgrade:

```
> show session info
```

9. Prepare a checklist of services to check post-upgrade. Refer to the upcoming *After the upgrade* section for a baseline and add additional or more specific checks as needed for your environment.

Now that we have made all the preparations and the maintenance windows have been set, it is time to perform the upgrade.

The upgrade process

When you start the upgrade process, quickly recheck each of the nine steps in the preceding section and reach out to your stakeholders to let them know that the maintenance window is about to start, and to wait for your signal to test whether all the applications and processes are running smoothly. Do this so that they have time to shut down any processes that do not handle interruptions very well. The first step is to upgrade Panorama.

In the first section, we'll upgrade a standalone Panorama instance followed by an HA cluster. Next, we'll review log collectors, and then we'll go for standalone firewalls, followed by firewall clusters. Skip ahead to the section you need (but remember to upgrade Panorama first).

> If you need to download the images manually, be mindful of which version you download. For Panorama and log collectors, there are two flavors: `Panorama_pc-x.x.x` for VM-installed Panorama/collector, and `Panorama_m-x.x.x` for chassis-installed Panorama/collector. For firewalls, the file name also contains the VM or chassis reference to ensure you select the right image: `PanOS_vm-x.x.x`, `PanOS_400-x.x.x`, and so on.

Upgrading a single Panorama instance

While upgrading Panorama introduces the least risk in terms of network impact, as it doesn't process sessions, a standalone Panorama instance does require an appropriate amount of precaution as a failed upgrade could introduce some difficulties with managing the firewalls. Be extra sure that an up-to-date configuration save is stored in a secure location, then follow these steps:

1. If any configuration changes have not been committed, review and save them, then collect a `running-config.xml` backup. Otherwise, discard the changes.
2. Go to **Panorama** | **Software** to install the preferred maintenance version of the currently installed major release, or run the following command in the CLI:

   ```
   > request system software install version x.y.z
   ```

   Keep track of the installation process from the CLI by using the following command:

   ```
   > show jobs all
   ```

3. When the upgrade is complete, the dialog window will request you to reboot. Click **Yes** to do so. If this dialog does not appear, go to **Panorama** | **Setup** | **Operations** and click on **Reboot Device**, or execute the following command in the CLI:

   ```
   > request restart system
   ```

   Press *Y* to confirm.

4. After the reboot, make sure the next major release base image and maintenance release are both downloaded on Panorama. If not, download them now, starting with the base image.
5. Install the preferred maintenance release package. When the installation is complete, click **OK** on the **reboot** dialog or manually initiate the reboot.
6. If you need to upgrade to another major version, repeat *steps 4* and *5*. Download or manually upload them, starting with the base image.
7. Reboot Panorama.

If you have a Panorama HA cluster, the procedure is a little different.

Upgrading a Panorama HA cluster

In a Panorama cluster that is still set in legacy mode and has an NFS volume set for log storage, only the primary Panorama instance receives logs, so there may be an interruption in log reception when upgrading the primary member. Firewalls will retain logs and forward them when Panorama comes online again.

If your Panorama environment is still in legacy mode, consider transitioning to **Panorama** or **Management-Only**, which offloads the log reception responsibility to log collectors. You can check your Panorama instance's current mode with the following command:

```
> show system info | match system-mode
```

Follow these steps to upgrade the cluster:

1. Verify if there is still uncommitted configuration. If there is, save and collect a fresh copy of `running-config.xml`; otherwise, discard it.
2. Start the upgrade process on the **Secondary-Passive** device first. Install the **preferred maintenance version of the current code train.**
3. When the installation task is complete, reboot the secondary device.
4. Verify that Panorama is up and running.
5. Suspend the **Primary-Active** member from the cluster by going to **Panorama | High Availability | Operational Commands** and clicking on **Suspend Local Device**. Alternatively, run the following command from the CLI:

   ```
   > request high-availability state suspend
   ```

6. Verify that the peer is now set to **Secondary-Active** and is operating normally.
7. On the **Primary-Suspended** device, install the **preferred maintenance release of the current code train.**
8. Reboot the device after the installation process is complete.
9. When the reboot is complete and the device is up and running, unsuspend the device by going to **Panorama** | **High Availability** | **Operational Commands** and clicking on **Make Local Panorama Functional**, or by using the following CLI command:

   ```
   > request high-availability state functional
   ```

Now that both members have been upgraded to the preferred maintenance version and we are certain that the cluster performs failovers as expected, we are ready for the major upgrade. Verify that the next major version base image and preferred maintenance release are present on both members. Download or manually upload them if needed:

1. On the secondary member, install the preferred maintenance release of the next major version.
2. Reboot the secondary device and wait for it to return to full functionality.
3. Suspend the **Primary-Active** member.
4. Verify that **Secondary-Active** is working as expected.
5. Install the preferred next major version maintenance release.
6. Reboot **Primary-Suspended**.
7. Unsuspend the primary device.
8. On the primary device, save to Panorama and manually execute **sync to remote** via **Dashboard | High Availability Widget**, or by using the following CLI command:

   ```
   > request high-availability sync-to-remote running-config
   ```

Next on the list is upgrading the log collectors to the same version as Panorama.

Upgrading log collectors (or firewalls) through Panorama

If Panorama is running in Panorama mode, the log collector will have been upgraded when Panorama was upgraded. Any additional log collectors will need to be upgraded afterward.

All of the log collectors in a collector group need to be upgraded at the same time as they all need to be on the same operating system for them to successfully pair up. During the upgrade process, logs will not be forwarded to the collector group and firewalls will store logs locally until the collector group comes back online, at which point the backlog is uploaded to the collector group. Since multiple devices need to be upgraded at the same time, it is best to perform the upgrade from Panorama. Upgrade Panorama to the same or a newer version before you upgrade the log collectors, then follow these steps:

1. In **Panorama** | **Device Deployment** | **Dynamic Updates**, ensure that the latest content updates are installed on all members of the collector group. Click **Check Now**, then download and install the latest updates if needed.
2. In **Panorama** | **Device Deployment** | **Software**, click **Check Now** and download the *preferred* maintenance release of the currently installed major version.
3. After the download is complete, click **Install** and select all the members of the collector group. Check the **Reboot device after install** checkbox and click **OK**.
4. Download the base image and the preferred maintenance release for the next major version.
5. Monitor **Panorama** | **Managed Collectors** for the log collectors to re-establish a connection to Panorama.
6. In **Panorama** | **Device Deployment** | **Software**, click **Install** next to the new base image, but select **Upload only to device** (**do not install**).
7. Then, click **Install** next to the preferred maintenance release and select **Reboot device after install**.
8. Repeat *steps 4* through *7* until you reach the desired major and maintenance release.

When upgrading log collectors to a new major version, there may be changes to the log database that take more time to complete than a regular upgrade. To monitor the process, you can log on to the CLI of the log collector and check the progress:

```
> debug logdb show-es-upgrade-time
```

After the log migration is complete, you can check the cluster health by issuing the following command:

```
> show log-collector-es-cluster health
```

Once the upgrade is complete, there should be sufficient time for aftercare. The next phase is to upgrade the firewalls.

Upgrading a single firewall

Upgrading a standalone firewall will cause interruptions to the network as all connections are dropped while the firewall reboots. Make sure all the stakeholders are notified and any critical processes are halted before commencing the reboot phase of the upgrade process:

1. Verify whether there are any uncommitted changes. Save and collect a fresh `running-config.xml` file; otherwise, discard it.
2. From **Device** | **Software**, install the latest maintenance release for the currently installed major version:

```
> request system software check
> request system software download version x.y.z
> request system software install version x.y.z
```

3. Keep track of the installation process from the CLI with the following command:

```
> show jobs all
```

4. When the upgrade is complete, the dialog window will request you to reboot. Click **Yes** to do so. If this dialog does not appear, go to **Device** | **Setup** | **Operations** and click on **Reboot Device**; otherwise, run the following in the CLI:

```
> request restart system
```

   Press *Y* to confirm.

5. After the reboot, the firewall will need to perform an `AutoCommit`, which is a job that pushes the newly upgraded configuration down to the data plane. Right after a reboot, the data plane is blank and will not start processing traffic until it has received its first configuration after booting up. It may take longer than a regular `commit` job to complete since all of the data plane processes need to get their configuration, whereas a regular commit can skip unchanged configuration. You can follow the `AutoCommit` by issuing the following command in the CLI:

```
> show jobs all
Enqueued Dequeued ID Type    Status Result Completed
----------------------------------------------------
23:22:18 23:22:18 1  AutoCom ACT    PEND   5%
```

6. Verify that the next major base image and preferred maintenance release packages are present in **Device** | **Software**. Download or manually upload the software package if needed, starting with the base image:

```
> request system software check
> request system software download version x.y.0
> request system software download version x.y.z
> request system software install version x.y.z
```

7. Install the preferred maintenance version.
8. Click **Yes** if the dialog window asks you to reboot; otherwise, reboot the firewall manually.

   ```
   > request restart system
   ```

   Press *Y* to confirm.

9. Wait for `AutoCommit` to finish and then test the firewall functionality.

When upgrading a firewall HA cluster, take the following steps to ensure a smooth upgrade.

Upgrading a firewall cluster

When upgrading a cluster, few to no sessions should be lost as the transition between peers is seamless. It is important to pace the process so that the firewalls have enough time to sync session states before the other member is suspended. Performing the following command on both members in between upgrades should result in roughly identical session counts:

```
> show session info
```

Follow these steps to start the upgrade:

1. Verify whether there are any uncommitted changes. Run `commit` and `sync` on the peer and collect a fresh `running-config.xml` file from both members; otherwise, discard the changes.
2. Remember to disable preemption in **Device** | **High Availability** | **General** | **Election Settings** if it is currently enabled.
3. On the secondary device, in **Device | Software**, click **Check Now**, and download the latest maintenance release for the currently installed major version. You will be prompted to sync the download to the peer. Select **Sync to Peer** and continue with the download. Install the latest maintenance release, or run the following via the CLI:

   ```
   > request system software check
   > request system software download version x.y.z
   > request system software install version x.y.z
   ```

4. Keep track of the installation process from the CLI with the following command:

   ```
   > show jobs all
   ```

5. When the upgrade is complete, the dialog window will request you to reboot. Click **Yes** to do so. If this dialog does not appear, go to **Device** | **Setup** | **Operations** and click **Reboot Device**, or run the following from the CLI:

   ```
   > request restart system
   ```

   Press *Y* to confirm.

6. After the reboot, the firewall will need to perform an `AutoCommit` process, which is a job that pushes the newly upgraded configuration down to the data plane. It may take longer than a regular `commit` job to finish. Track the progress by issuing the following command in the CLI:

```
> show jobs all
Enqueued Dequeued ID Type Status Result Completed
---------------------------------------------------------
23:22:18 23:22:18 1 AutoCom ACT PEND 5%
```

7. Wait until the secondary device is in the `Passive` state (older PAN-OS versions may go into the `NonFunct` state at this time, but the functionality is the same). Check whether the session table is being synced:

```
> show session all
```

8. On the primary device, go to **Device** | **High Availability** | **Operational Commands** and click **Suspend local device**, or run the following on the CLI:

```
> request high-availability state suspend
```

9. On the primary device, install the *preferred* maintenance release of the current code train.
10. When the upgrade completes, reboot the firewall.
11. Wait for the `AutoCommit` process to complete and the firewall to go into a `Passive` state.
12. To return the primary device back to an `Active` state, run `suspend` and then `unsuspend` for the secondary device:

```
> request high-availability state suspend
> request high-availability state functional
```

13. Verify that the next major base image and preferred maintenance release packages are present in **Device** | **Software** on both devices. Download or manually upload the software package if needed, starting with the base image:

```
> request system software check
> request system software download version x.y.0
> request system software download version x.y.z
> request system software install version x.y.z
```

14. On the secondary device, install the preferred maintenance version of the next major release:

```
> request system software check
> request system software download version x.y.0
> request system software download version x.y.z
> request system software install version x.y.z
```

15. Click **Yes** if the installation dialog window asks to reboot, or reboot the firewall manually:

```
> request restart system
```

Press *Y* to confirm the reboot.

Wait for the `AutoCommit` process to be complete and for the secondary device to go into the `Passive` state (older PAN-OS versions may go into a `NonFunct` state).

16. Verify whether the state table is being synchronized:

```
> show session all
```

17. On the primary device, click on **Suspend local device** in **Device** | **High Availability** | **Operational Commands**, or run the following in the CLI:

```
> request high-availability state suspend
```

18. In **Device** | **Software**, install the preferred maintenance release of the next major version, or run the following via the CLI:

```
> request system software install version x.y.z
```

19. When asked to reboot in the **Install** dialog, click **Yes**, or manually reboot via **Device** | **Setup** | **Operations**. Alternatively, run the following in the CLI:

```
> request restart system
```

    Press *Y* to confirm rebooting the firewall. Wait for the `AutoCommit` process to finish.

20. In the **Dashboard** | **High Availability** widget, verify that the configuration is in sync. If not, sync the configuration to the secondary device with the widget or through the CLI:

```
> request high-availability sync-to-remote running-config
```

21. Verify that the primary device is in a passive state. If it is still suspended, unsuspend it:

```
> request high-availability state functional
```

22. To return the primary device to an active state, suspend and then unsuspend the secondary device:

```
> request high-availability state suspend
> request high-availability state functional
```

23. Re-enable **Preemptive** on the primary device if it was enabled before, and save the changes.

> **Important Note**
>
> If one cluster member is upgraded more than one major version higher than its peer, for example, the primary is on version 10.2.11 and the secondary is upgraded to 11.1.5, the secondary will go into the `NonFunct` state, which is not recoverable until either the primary is upgraded to an appropriate version or the primary member becomes unavailable. Simply suspending the primary member will not make the secondary member become **Active**.

When the upgrade is complete, the work is not yet done; ensure you have the right contacts and procedures ready for aftercare, making sure everything still works.

After the upgrade

During the aftercare phase, you should go over your checklist to ensure all the applications are up and running and any other critical infrastructure or business processes are fully functional and not being blocked by the firewall. Use this list as a template, adding your own checks as needed:

- Reach out to your stakeholders to run tests and verify that everything is working as expected. Monitor their tests in the traffic log to verify that all the allowed sessions are allowed, and blocked sessions are still being blocked.
- Verify whether any deviating session settings have been included in the upgrade and reset them if needed.
- For firewalls managed by Panorama or a Panorama upgrade, verify whether pushing configuration from Panorama works as expected.
- Check the VPN and GlobalProtect connections.
- Verify whether the dynamic routing protocols are picking up routes as expected.
- Check the system logs for unexpected error messages.

You should now be able to upgrade Panorama, log collectors, and firewalls and have the appropriate procedures in place to fully prepare and perform aftercare once you are done. In the next section, we'll review what to do if the upgrade fails and you need to get back to a previous situation quickly.

The rollback procedure

If the upgrade causes unexpected issues and troubleshooting is unable to clarify why, the last resort is to roll back to the previous deployment.

If you find yourself in this situation, make sure you do the following:

- Write down all the symptoms
- Note down which troubleshooting steps were taken
- In **Device** | **Support**, create a `Techsupport` file as you may need to reach out to Palo Alto Networks support if you are unable to find what went wrong
- Save any related files, the CLI output, troubleshooting files, packet captures, and so on in one location

Once you've documented your troubleshooting efforts, the easiest way to roll back is to switch the `sysroot` boot partition. The firewall has two system volumes that contain a fully installed PAN-OS, of which only one partition is active. The inactive partition either contains the previously installed version or the next version if you have just installed it but not rebooted yet.

From the CLI, you can query the status, which shows you which version is currently `RUNNING-ACTIVE` and which one is installed on the inactive partition, and can be reverted to the following:

```
> debug swm status
Partition       zState            Version
---------------------------------------------------------
sysroot0          REVERTABLE        11.2.4
sysroot1          RUNNING-ACTIVE    11.2.4-h1
maint             READY             11.2.4-h1
```

To roll back after an upgrade, you can simply activate the previous partition by executing the following command:

```
> debug swm revert
Reverting from 11.2.4-h1 (sysroot1) to 11.2.4 (sysroot0)
```

Then, reboot the system:

```
> request restart system
```

This will take you back to the previously installed version of PAN-OS.

If this procedure fails, reverse the steps of the installation procedure until you have installed the maintenance version that you started from:

1. If needed, download the base image of the previous major version, then download the preferred maintenance release or the maintenance release you were on when you started.
2. Install the maintenance release directly. The system will prompt you to choose a specific configuration file to download to. Pick your backup file. If you don't have a backup file, just pick `running-config.xml`.
3. Reboot.
4. If you need to go down another major version, download the base and maintenance release.
5. Install the maintenance release, pick the desired backup configuration file and reboot.
6. After the device has rebooted to the desired release, if you did not have a device-loaded backup config file, go to **Device** | **Setup** | **Operations** and click **Import named configuration snapshot** to load your backup config.
7. Then, click on **Load named configuration snapshot**, pick your backup configuration file, and click **Commit**.

If you want to, you can also downgrade to a previous version.

The downgrade procedure

There may be a time when you have upgraded to a newer version but feel you want to remain on the previous version for a while longer and need to downgrade. Rather than reverting to the previous version, which may be an older maintenance release, you can downgrade to the previous major release, but to the latest maintenance release in that build, following these steps:

1. Verify that both the base image and preferred maintenance release versions have been downloaded on both members when downgrading a cluster. If the images were removed, download the base image first, then download the maintenance version.
2. If you are downgrading a cluster, suspend and upgrade the primary device first.

   When you initiate a downgrade to a lower major version, the system will ask whether you want to load a configuration file that was saved just before the previous *upgrade*. This will ensure you revert to a configuration file that was used in the version you are downgrading to.

   Unless a lot of changes were implemented after the upgrade, it is a good idea to load the file, rather than relying on the conversion process of the current configuration to the lower major version.

3. If the primary device is still in a suspended state after the reboot, set it to **functional**. This will cause the primary device to become active and start processing sessions, regardless of whether `preempt` is enabled. In a cluster, the member with the lowest PAN-OS major release will gain priority over the other peer. During the upgrade process, this will prevent an uncontrolled failover after one peer has been upgraded, but in a downgrade, it may be a bit of a challenge:

   ```
   > request high-availability state functional
   ```

4. Downgrade the secondary device, making sure you make the same choice regarding the configuration file. Load the previous version or rely on the downgrade conversion.
5. After the downgrade completes on the secondary device, set it to the `functional` state:

   ```
   > request high-availability state functional
   ```

6. Sync the configuration from the primary to the secondary device:

   ```
   > request high-availability sync-to-remote running-config
   ```

Following these steps should bring you back safely to a previous version.

In the next section, we'll review a corner case when upgrading older hardware.

Special case for upgrading older hardware

Some older hardware may not have sufficient space on the hard drive to accommodate upgrading directly from one major version to the next. This will become apparent if you first download the base image and then download and install the maintenance release as you will receive an error message saying that the base image is missing. This is caused by the system trying to load the maintenance image by deleting any images that are not in use at the time, which in this case is the base image. For these special cases, follow these steps to upgrade successfully:

1. Delete any non-essential software images.
2. Download the base image of the next major version, install, and reboot.
3. After the reboot, download the maintenance version, install, and reboot.

Some old hardware may not support a new version of PAN-OS, or some form factors may suffer specific issues that should be described in the release notes under the known issues. Always review these notes briefly before moving forward to a new major release.

Summary

In this chapter, you have learned how to upgrade the Panorama management system, log collectors, and firewalls. You have also learned how to upgrade clusters in such a way that it will cause no or minimal impact on your business, and can plan in advance which steps will need to be taken to ensure that the upgrade goes smoothly. You have also learned how to roll back in the event of failure, or gracefully downgrade if needed. You also know which precautions need to be taken when upgrading an older or smaller form factor device.

If you're preparing for the PCNSE, keep in mind the difference between major, feature, and maintenance releases and how upgrades should be performed. A base image is needed before a maintenance release can be installed.

In the next chapter, we will learn how to set up log collectors and set them up redundantly, as well as how to create custom reports.

10

Logging and Reporting

In this chapter, we will learn how logs can be forwarded to log collectors, syslog servers or emailed. We'll also learn how to select which logs are sent to a specific destination, and what event trigger logs should be sent. We'll learn how to configure log collectors and create a log collector group to ensure redundancy and increase log capacity. We'll also learn about built-in reports and how custom reports can be created.

In this chapter, we're going to cover the following main topics:

- Log storage
- Configuring log collectors and log collector groups
- Leveraging **Strata Logging Service** (**SLS**), formerly Cortex Data Lake
- Logging to an external syslog
- Configuring log forwarding profiles
- Filtering logs
- Predefined reports and creating custom reports
- Using the Application Command Center

By the end of this chapter, you'll be able to ensure that the logs that matter most are stored in a safe location for as long as you need them. You'll be able to collect quick statistics on the types of applications and other data that traverses your network.

Technical requirements

In this chapter, we will be forwarding logs via syslog and sending out alerts via email. If you do not have access to a syslog server and an email relay, set them up so that you can test the topics we discuss. There are several free software packages available, like Kiwi Syslog Server and the SMTP server in Windows IIS, that can get you started.

Log storage

In its standalone configuration, a firewall has somewhere between a few terabytes of storage on high-end devices and a few gigabytes on low-end devices for logs (the PA-410 series doesn't have local log storage as it was designed to be managed by Panorama or Strata Cloud Manager and send its logs to a log collector or SLS). This space then has to be split up among all the different log databases, such as **Traffic**, **Threat**, **URL Filtering**, **WildFire**, and several others. This could cause a skewed perception of how much log storage is actually available, which, combined with high traffic volumes, could lead to the system having only enough storage for a couple of days' worth of logs.

To review the current log capacity and what percentage of the capacity has been assigned to individual databases, check **Device** | **Setup** | **Management** | **Logging and Reporting Settings**. You can change how much space is reserved for each log database by changing the percentage next to each log database. Keep in mind that changing the allocated space after the system has already been collecting logs for a while may purge some or all of the logs that were already stored. While assigning quotas, keep an eye on the total allocation near the bottom left of the screen, as illustrated in the following screenshot:

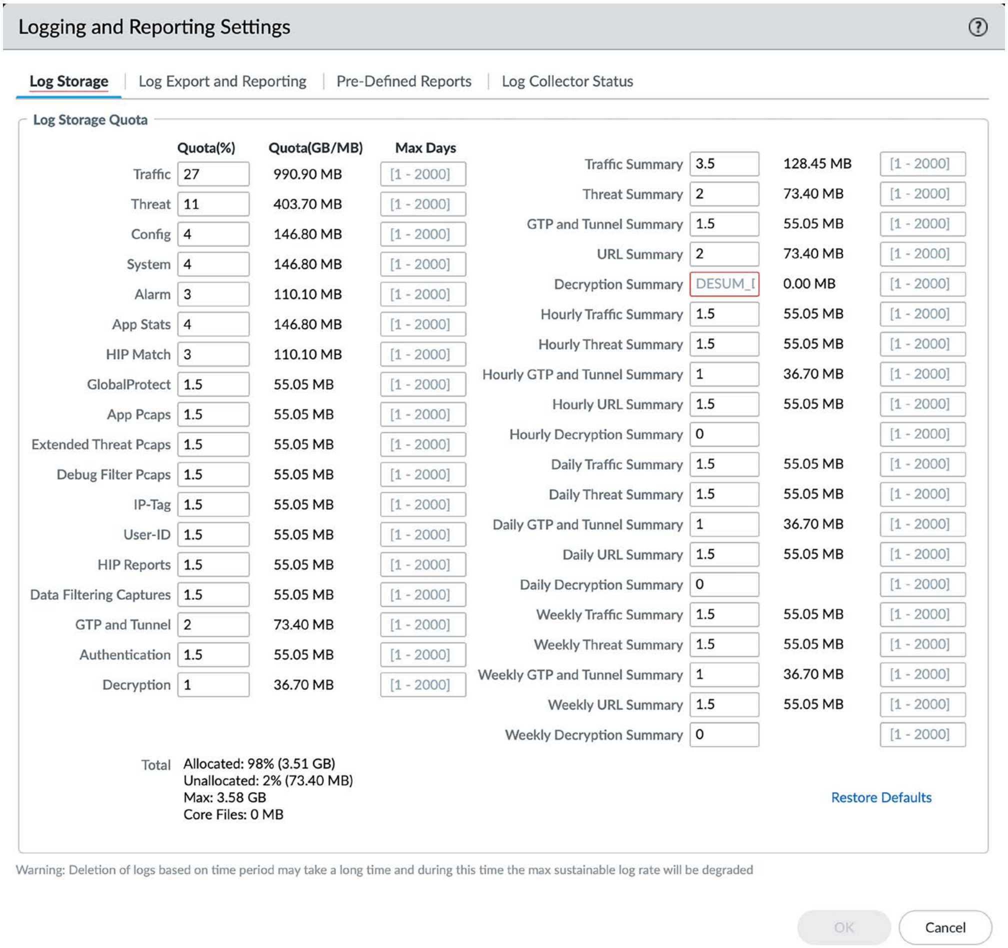


Figure 10.1: Log storage percentage

Note

In some cases, you may notice that one of the databases has no value and is highlighted in red, like the Decryption Summary in *Figure 10.1*. This can happen when a new database is added in a release that you've upgraded to from a previous version. You can assign it a quota to start using it.

The rule of thumb is that for an average log rate of `10` logs per second and a retention period of `30` days, you need around `60` gigabytes of storage. The average log rate on a mid-range firewall is estimated at around `400` logs per second, which requires nearly `2.5` terabytes of storage to save for `30` days.

There is a calculator available at `https://apps.paloaltonetworks.com/sls-sizing-estimator` that gives you a good estimate regardless of whether you're logging to SLS or log collectors.

Important note

On the local hard drive, logs are pruned on a *first-in-first-out* basis in accordance with their database quota. If the **Traffic** database is full, the oldest logs from this database will be pruned. Other databases are left alone.

Review the quota usage and retention estimate with the following command:

```
> show system logdb-quota
```

The output will look similar to what you see here:

```
admin@PANgurus> show system logdb-quota
Quotas:
system: 4.00%, 0.143 GB Expiration-period: 0 days
config: 4.00%, 0.143 GB Expiration-period: 0 days
alarm: 3.00%, 0.108 GB Expiration-period: 0 days
appstat: 4.00%, 0.143 GB Expiration-period: 0 days
hip-reports: 1.50%, 0.054 GB Expiration-period: 0 days
traffic: 27.00%, 0.968 GB Expiration-period: 0 days
threat: 11.00%, 0.394 GB Expiration-period: 0 days
...
Disk usage:
traffic: Logs and Indexes: 184M Current Retention: 44 days
threat: Logs and Indexes: 49M Current Retention: 44 days
system: Logs and Indexes: 148M Current Retention: 29 days
config: Logs and Indexes: 25M Current Retention: 46 days
alarm: Logs and Indexes: 32K Current Retention: 0 days
trsum: Logs and Indexes: 47M Current Retention: 44 days
...
```

You're now able to review how much storage is available on your system and decide how you may be able to better divvy up the available quota.

In the next section, we will learn how to set up log collectors and log collector groups.

Configuring log collectors and log collector groups

To ensure that logs can be stored for an extended period of time, as you may need to comply with certain standards that require lengthy log storage (regulations such as **SOX** and **HIPAA** and standards such as **ISO 27001** require several years' worth of logs to be stored), logs can be exported into a dedicated log management system (tools like Elastic Stack, LogRhythm, or Splunk).

You can create additional log collectors by setting up and licensing a second Panorama in Panorama mode and creating a high availability cluster, or by adding additional Panorama appliances and configuring them in Logger Mode. Both **virtual machine (VM)** and physical machine appliances can be used to achieve the aforementioned, but the cluster option requires both devices to be the same flavor (both physical machines or both VMs).

On the device you want to set to Logger Mode, run the following command from the CLI:

```
> request system system-mode logger
```

In Panorama, you can add multiple log collectors in **Panorama** | **Managed Collectors** and then add them to one or more groups in **Panorama** | **Collector Groups**.

To improve availability, you can select **Enable log redundancy across collectors** in the log collector group. This will create a second copy of every log entry, which is stored in a different log collector. This will ensure that logs are always available, even if a log collector is unreachable. This will consume additional disk space, so carefully weigh the need for availability over the retention period.

As you can see in the following screenshots, there are several different ways that log collectors can be deployed to best suit an organization's needs:

1. To split managed devices up into groups and set a preferred log collector.

2. In *Figure 10.2*, the top collector is preferred, and the next collector will be used if the primary one fails or is unreachable:

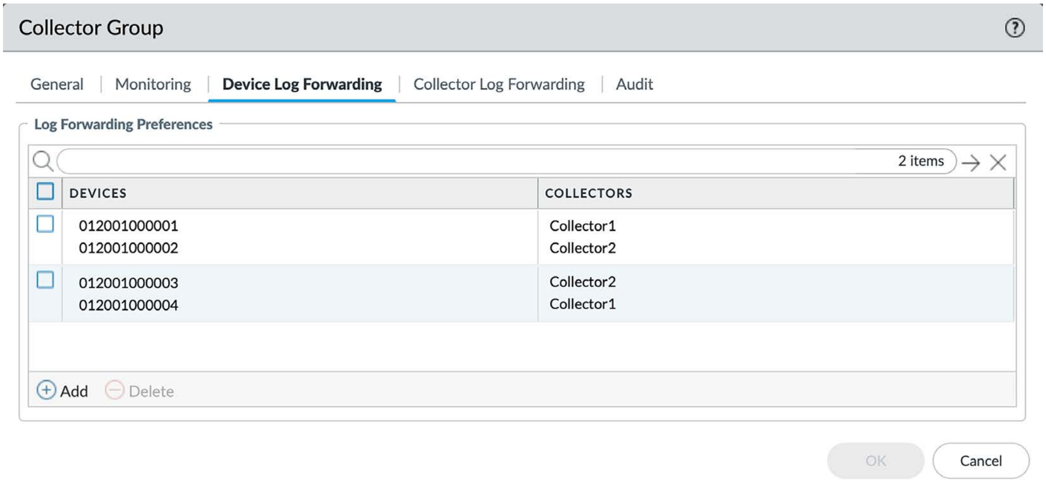


Figure 10.2: Different collector groups

3. To add all managed devices and collectors to one pool and have all devices send logs to the same collector:

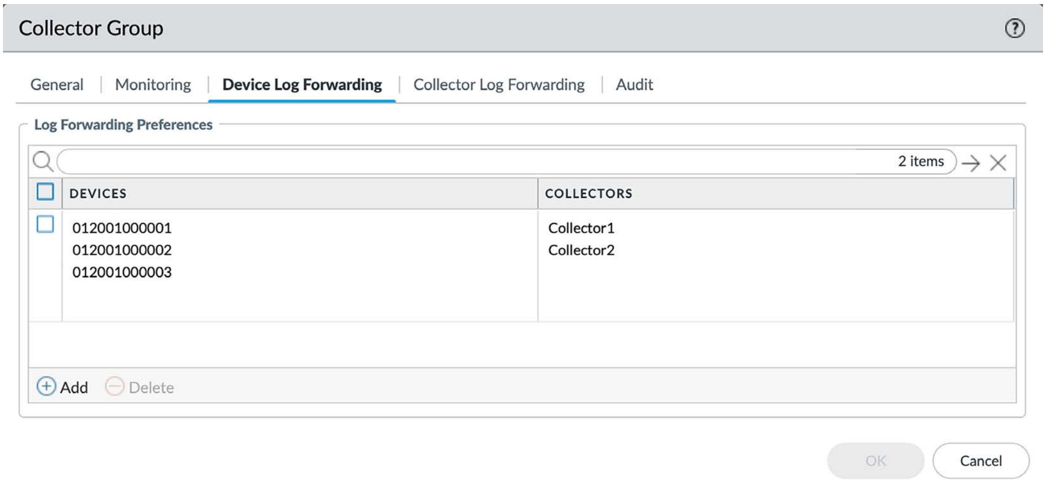


Figure 10.3: A single collector group

4. To limit the collectors available to managed devices. This could be helpful if devices and collectors are spread out geographically:

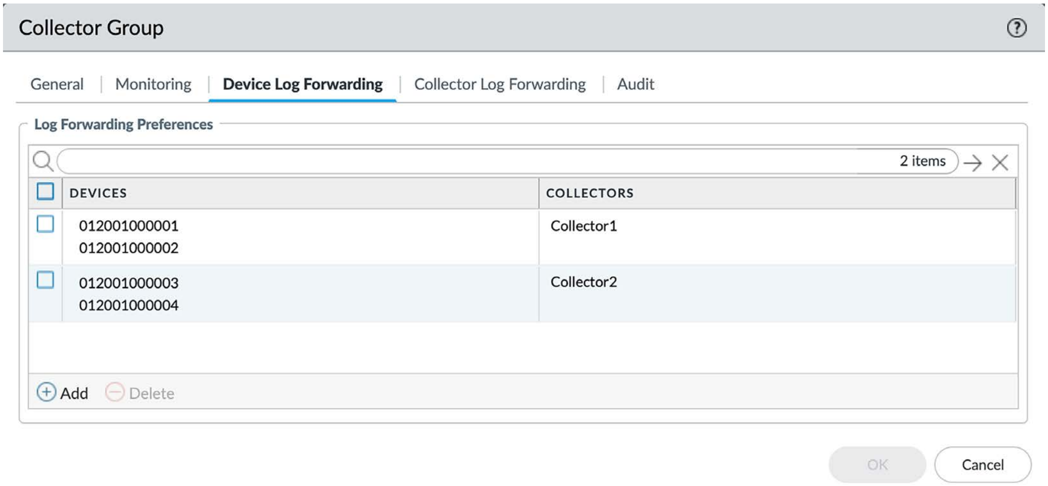


Figure 10.4: Limited availability of log collectors to devices

> **Important note**
>
> Even though logs are forwarded to the preferred log collector device, the log collector group will evenly distribute logs among all members of the collector group as the collector group is considered one logical unit.

If the bandwidth between geographical locations is too limited for a collector group to efficiently distribute logs among its peers, consider making multiple groups.

Panorama will push these settings out to the managed devices so that they are made aware of which exact destinations they are expected to log to. In the firewall, you can verify the preference list by executing the following command:

```
> show log-collector preference-list
```

The output of running this command should look similar to the following:

```
reaper@PANgurus> show log-collector preference-list
Log Collector Preference List
Forward to all: No
Serial Number: 000700001 IP Address: 192.168.27.10 IPV6 Address: unknown
```

You can also verify whether logs are being forwarded properly:

```
> request log-collector-forwarding status
```

As an alternative to deployed physical log collectors, you can also log to the cloud, which we'll see in the next section.

Leveraging Strata Logging Service

With **Logging Service**, currently called **Strata Logging Service (SLS)**, logs are no longer sent to Panorama or a collector group, but instead go up into the cloud through a secure connection. This is a licensed feature, so every firewall that should log to the cloud will need to be outfitted with a license. Once the licensing is in order and SLS is properly set up at `https://apps.paloaltonetworks.com/marketplace/strata_logging_service`, you can configure each firewall locally or through a Panorama template.

From **Device** | **Setup** | **Management** | **Logging Service**, you can select **Enable Logging Service**. PA-5200 and PA-7000 can have multiple (up to 20) simultaneous connections to SLS (and log collectors). This is achieved by selecting **Enable High Speed Log Forwarding** from **Device** | **Setup** | **Logging and Reporting Settings**. One important caveat is that when **High Speed Log Forwarding** is enabled, all local log storage is **disabled** so logs will only be visible from Panorama or SLS Explore:

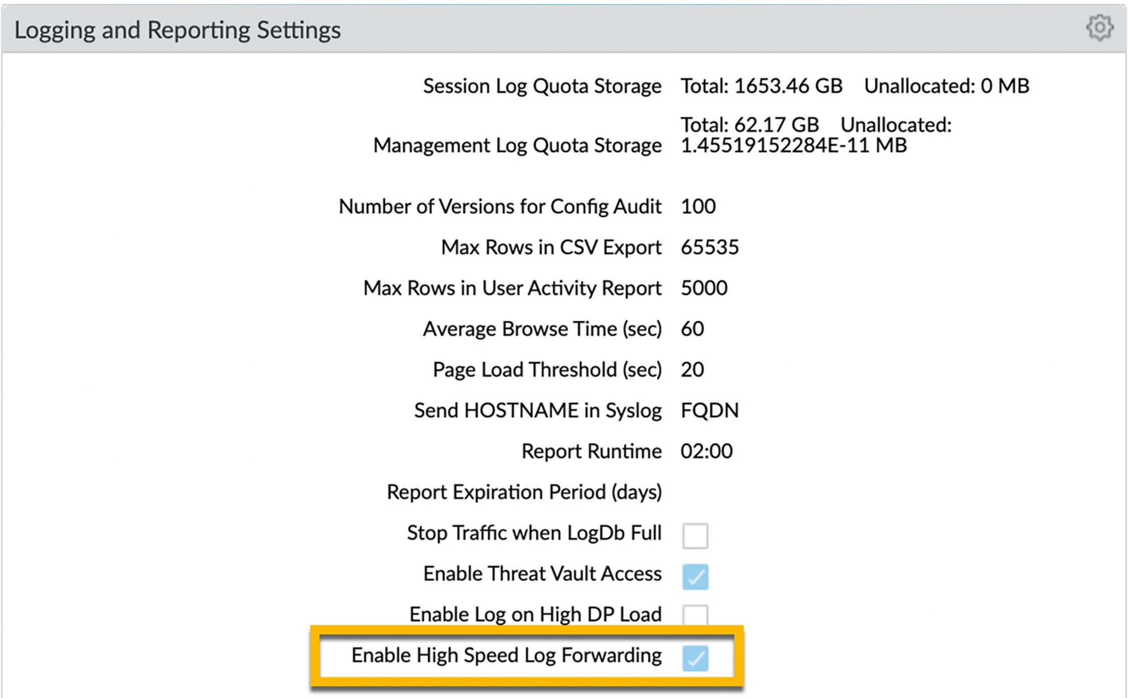


Figure 10.5: Enable High Speed Log Forwarding

In **Device** | **Setup** | **Cloud Logging**, the **Enable Enhanced Application Logging** option will increase the information gathered about applications and send it to SLS. These logs can only be used by Cortex applications and will not be visible to you. As the following screenshot shows, you can also use **Enable Duplicate Logging (Cloud and On-Premise)**, which writes logs to Panorama or the log collectors and sends a copy to SLS:

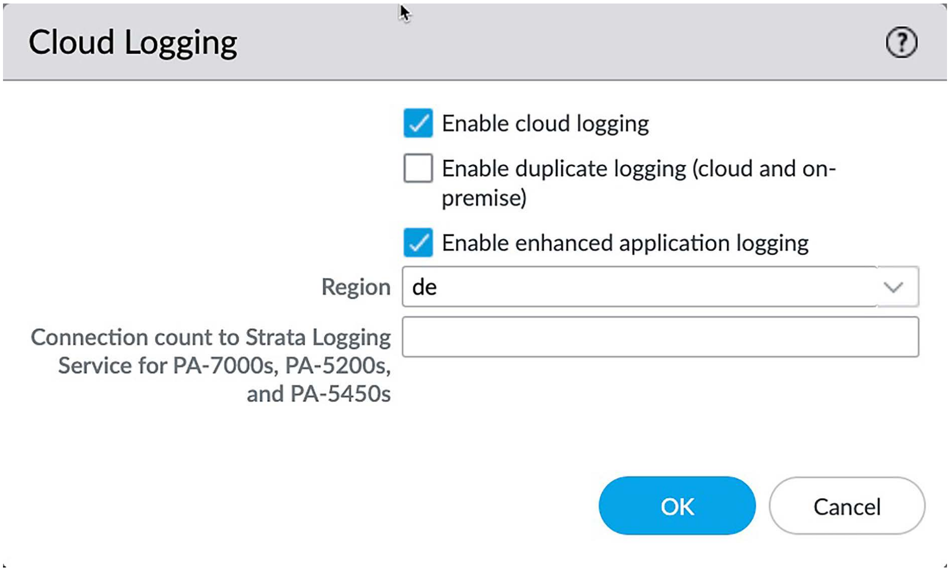


Figure 10.6: Logging service

Starting from PAN-OS 9.0.2, you can also connect firewalls that are not managed by Panorama to **Logging Service**. As you can see in the following screenshot, **Onboard without Panorama** has a **Connect** option, which lets you connect to **Logging Service** using a **Pre-Shared Key** (**PSK**), which you first configure in the SLS portal:

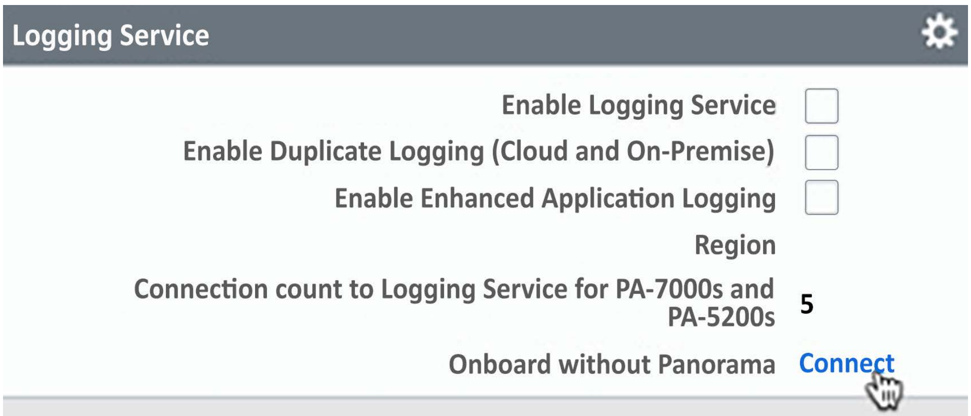


Figure 10.7: Onboard without Panorama

You can check whether the firewall is connected to **Logging Service** by issuing the following command:

```
> request logging-service-forwarding status
```

You now have a good grasp of the advantages SLS may have over local storage. In the next section, we'll review alternative options for sending logs out.

Logging to an external syslog

As well as *native* logging to Palo Alto Network products, you can also forward logs to syslog servers, email them out, send SNMP traps, or forward them to an HTTP server. To be able to forward logs, we will first need to create server profiles that we can later use when we set up forwarding.

For SNMP, we can create a new profile in **Device** | **Server Profiles** | **SNMP Trap**. Here, we can choose **V2c** or **V3** SNMP compatibility and provide connectivity details of the SNMP server. Unless absolutely necessary, avoid using **V2c** as this version is no longer a secure option. If **ENGINEID** is left blank, as in the following screenshot, the firewall will insert its serial number:

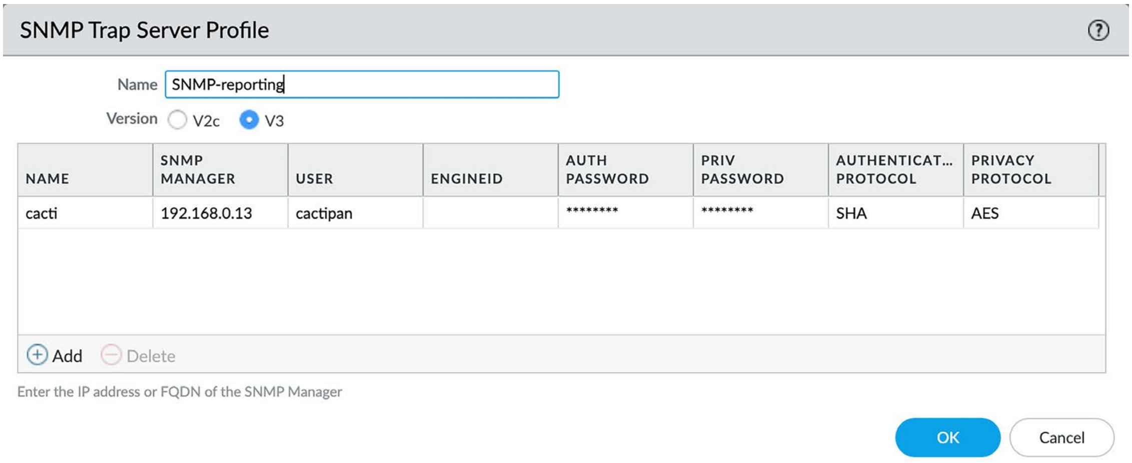


Figure 10.8: SNMP v3 server profile

For syslog, we can create a profile in **Device** | **Server Profiles** | **Syslog**. We have the option of forwarding over UDP, TCP, or SSL. If possible, select **SSL** as these logs should be considered highly sensitive, and forwarding them as plaintext can generally be considered a bad idea (it could lead to data leaks if intercepted in plaintext). In the **Custom Log Format** tab, you can change how outgoing syslog messages are formatted for each log type. This may be handy if your syslog server configuration has been tweaked to accept different log formatting.

For email, we can create a profile in **Device** | **Server Profiles** | **Email**. Here, you need to provide your email relay address, your *from* and *to* email addresses, and a friendly display name. You also have the option to customize the log format. Do use the email option sparingly and with appropriate filters as it could cause a flood of emails if attached to a log forwarding profile that sees many events. Refer to the *Configuring log forwarding* section for more details on adding filters.

For HTTP-based systems, we can create a profile in **Device** | **Server Profiles** | **HTTP**. These profiles can be used to interact with a ticketing system like Salesforce to automatically open a ticket with IT if a specific event is detected. Simply create a new profile and add a name. The default settings are HTTPS over port 443 using TLS 1.2. A certificate profile can be added if needed. The HTTP method is POST, but it can be changed to DELETE, GET, or PUT. Finally, a username and password are required. This forwarding mechanism should also be set with an appropriate filter so there's no flood of support tickets for common events. If, for some reason, the system uses unencrypted HTTP, this can also be set but this is obviously not recommended.

You have now learned which options are available to forward logs from your device and how to set them up. In the next section, we will learn how to configure log forwarding and review how we can select which logs are forwarded.

Configuring log forwarding profiles

The firewall will not automatically forward all logs to Panorama or **Logging Service**. Log forwarding needs to be configured and assigned to specific logs or log types before anything is sent out. Two main types of logs can be forwarded:

- **System event logs:** Any logs related to how the system is operating or events happening on the system itself
- **Traffic flow-related logs:** All logs related to packets flowing through the data plane of the firewall

Device daemon-related logs are only stored locally.

> **Important note**
>
> Only logs that are being stored locally can be forwarded. Any rule, policy, or profile that is set to not log cannot generate logs to be forwarded. Forwarded logs will also remain available locally (for as long as storage allows for the log to be retained); they are not purged after being forwarded.

In the firewall, you can check whether log forwarding is available and working with the following commands:

```
> reaper@PA-440> request logging-service-forwarding status

Logging Service Licensed: Yes
Logging Service forwarding enabled: Yes
Duplicate logging enabled: No
Enhanced application logging enabled: Yes

Logging Service License Status:
Status:
        Status: success
        Expiration date: August 30, 2027
```

```
        Msg: License is valid
        Last Fetched: 2024/11/19 11:36:47

Fetch:
        Status: Success
        Msg: Successfully fetched license
        Last Fetched: 2024/08/26 16:04:59

Install:
        Status: Success
        Msg: Successfully install fetched license
        Last Fetched: 2024/09/01 00:25:30

Upgrade:



Logging Service Certificate information:
         No certificate found


Logging Service Customer file information:
        Customer ID: 1979111159
        EAL Ingest FQDN: 955cebdc-2d12-423d-b287-65de3f3c1e13.fei-lc-prod-de.
gpcloudservice.com
        Ingest FQDN: 955cebdc-2d12-423d-b287-65de3f3c1e13.in2-lc-prod-de.
gpcloudservice.com
        Info: Successfully fetched Logging Service customer info
        Query FQDN: 955cebdc-2d12-423d-b287-65de3f3c1e13.api2-lc-prod-de.
gpcloudservice.com:444
        Region: Germany
        Status: success
        Last Fetched: 2024/12/14 23:04:58


Logging Service Preference List
Forward to all: Yes
Serial Number: PANW_LOG_RECEPTOR_SRV FQDN: 955cebdc-2d12-423d-b287-
65de3f3c1e13.in2-lc-prod-de.gpcloudservice.com
```

```
------------------------------------------------------------------------------
---------------------------------------------
      Type      Last Log Created        Last Log Fwded       Last Seq Num Fwded
Last Seq Num Acked         Total Logs Fwded
------------------------------------------------------------------------------
---------------------------------------------


Log Collector           : RECEPTR04USSTG
Conn ID                 : lr-35.198.100.110
Connection IP           : 35.198.100.110
Conn Source IP          : lr - def
High speed mode         :    Disabled
Connection Status       : lr - Active
DNS                    :
                    msg : Successfully resolved FQDN for connid (lr-
35.198.100.110-def), IP (35.198.100.110)
                 status :     success
              timestamp : 2024/12/13 16:59:50

Registration           :
                    msg : Successful registration with lr-35.198.100.110-def
                 status :     success
              timestamp : 2024/12/13 16:59:53

SSL                    :
                    msg : ssl channel established
                 status :     success
              timestamp : 2024/12/13 16:59:52

TCP                    :
                    msg : tcp connection established
                 status :     success
              timestamp : 2024/12/13 16:59:50

Conn Uptime             :           0
Re-conn Count           :           0

Rate                    :  8 logs/sec
```

In the output, you'll notice a couple of different sections that will tell you if the license is valid and installed, the FQDNs that will be used for log forwarding, and whether the firewall was able to resolve and connect to them.

Let's first take a look at the system logs.

System logs

In **Device | Log Settings**, you can set forwarding profiles for **System**, **Configuration**, **User-ID**, **HIP match**, **GlobalProtect**, and more. Simply add a new profile for the logs that need to be forwarded to be centrally available.

If you create a log forwarding profile for, say, a system log, you can check the box next to **Panorama/ Logging Service** to forward logs to Panorama or the cloud, and/or you can set any of the other log forwarding preferences.

> **Note**
>
> The column name may vary depending on whether your firewall logs to a collector or to SLS, so your view might differ slightly from the screenshot below. If logging to a collector, the column will be labeled **Panorama**, if logging to SLS, the column will be labeled **Cloud Logging**.

You can also create multiple forwarding profiles. Each profile has a **Filter** field at the top that has severity filters prepopulated, as well as a filter builder. You can create your own filter incorporating the `AND` or `OR` operators so that only specific events trigger this forwarding action.

For example, as depicted in the following screenshot, an email could be sent out to the security team, and a syslog event sent if a failed authentication event is detected for an administrator trying to log into the firewall, while all logs with a severity of **medium** or higher (the geq operator) are forwarded to Panorama or **Logging Service**:

System

☐	NAME	DESCRIPTION	FILTER	PANORAMA	SNMP TRAP	EMAIL	SYSLOG	HTTP
☐	logs-to-panorama		(severity geq medium)	☑				
☐	alert-OpSecTeam	failed login	(eventid eq auth-fail)	☐		SecTeam-email	splunk	

⊕ Add ⊖ Delete Clone PDF/CSV

Figure 10.9: The Log Forwarding filter for failed authentication and Panorama logs

When creating filters, you can use the built-in **Filter Builder** feature, as shown in the following screenshot. All of the available attributes are there, and in many cases, the values are also prepopulated. Simply select the attributes, operators, and values, then click **Add**, and the filter is created:

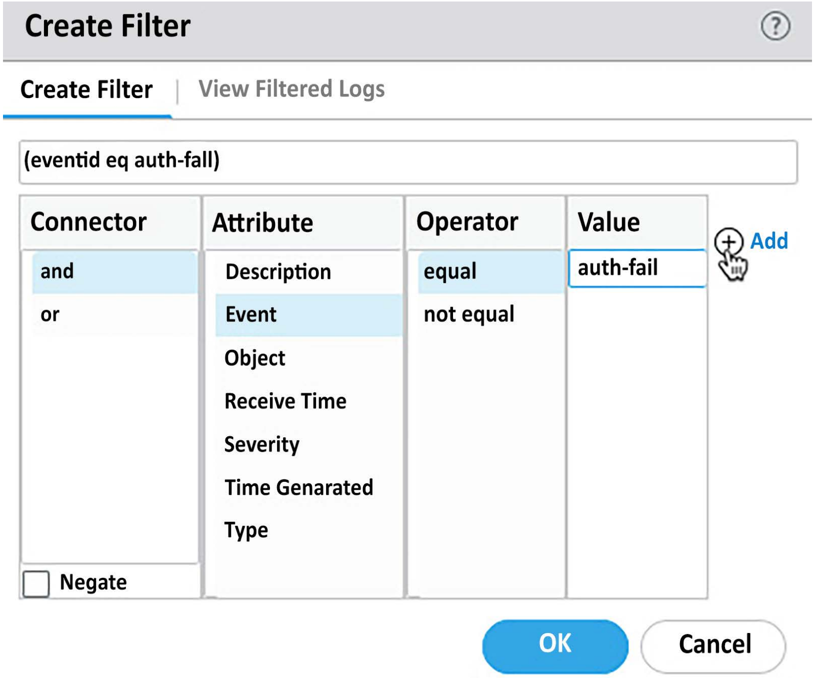


Figure 10.10: Filter Builder

Note

While firewalls can send system logs to Panorama, log collectors or SLS, Panorama itself cannot forward its system logs to SLS.

Next, let's take a look at all the logs that relate to packets flowing through the system.

Firewall logs

For logs related to sessions handled by the firewall, a log forwarding profile needs to be created in **Objects** | **Log Forwarding**.

Important note

If you create a log forwarding profile and name it `default`, it will be automatically added to any new security rule that is created, thus ensuring log forwarding to Panorama or **Logging Service** is not forgotten.

For each log type (**traffic**, **threat**, **url**, **wildfire**, **auth**, **data**, and **tunnel**) that you want to forward, you can create a rule with instructions.

> **Note**
>
> A quick recap of the log types:
>
> - Traffic logs relate to how all sessions are handled by the firewall
> - Threat logs contain any virus, malware, or threats that were detected
> - URL logs relate to URL filtering actions (where a traffic log has an allow action, the URL log may say the action is block)
> - WildFire logs relate to files that were intercepted and their verdict
> - Auth logs relate to sessions hitting authentication rules
> - Data logs relate to file-blocking actions taken
> - Tunnel logs relate to GRE, VXLAN, or not-encrypted IPSec sessions that were inspected

You can also add more specific rules that perform a specific action if a certain event is encountered. In the following screenshot, you can see an example of this where a syslog and email message will be sent only if a brute-force attack of high or critical severity is detected:

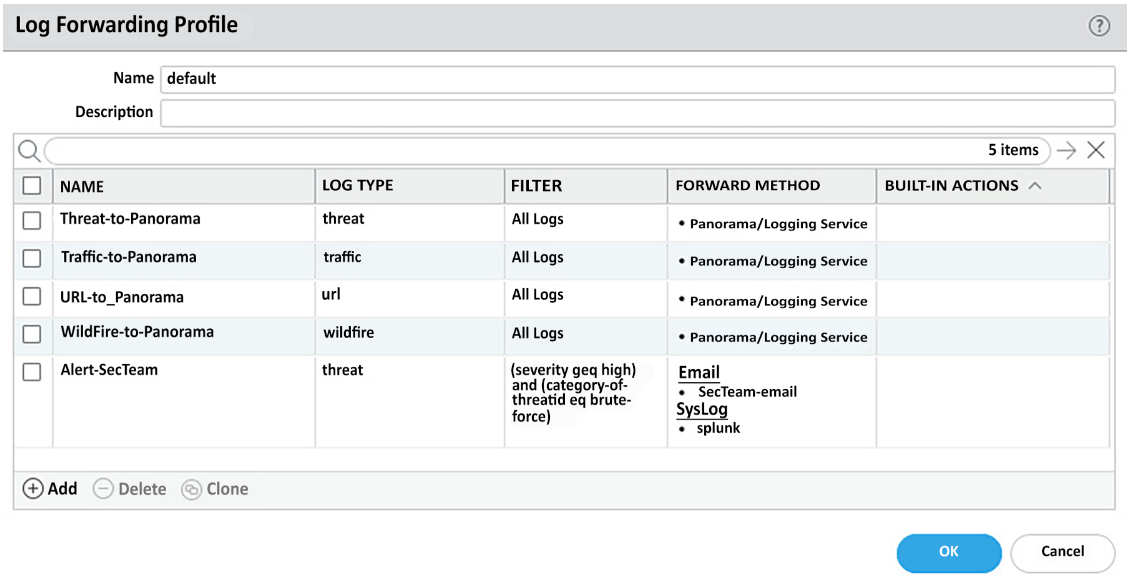


Figure 10.11: Log Forwarding Profile

If you want to verify which kinds of logs are captured with the filter you created, you can review them in **View Filtered Logs**. As you can see in the following screenshot, the filter will be transported into a log view so that you can review the type of logs that will be forwarded:

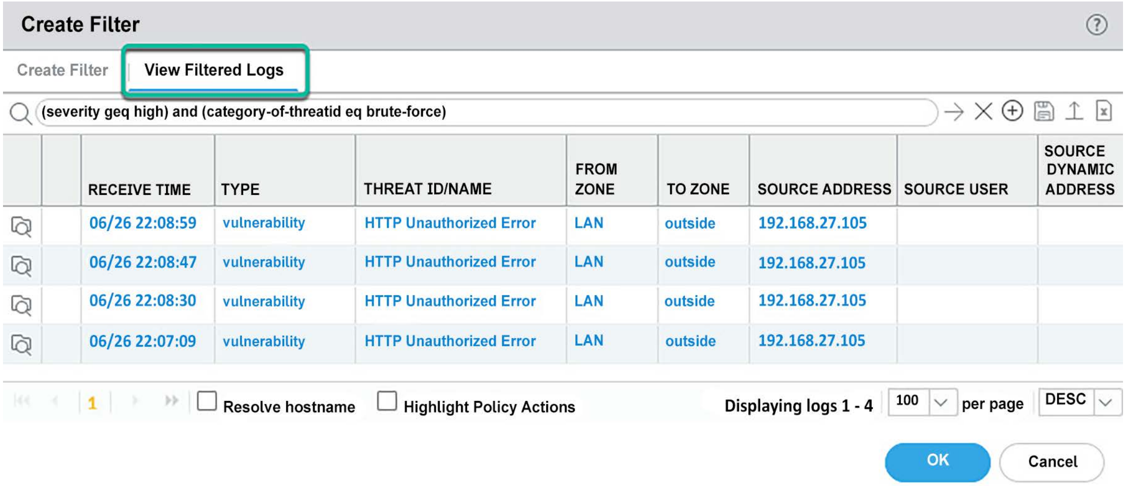


Figure 10.12: The View Filtered Logs preview tab

If you want to build more specific log forwarding rules, you can either add more rules to the log forwarding profile or you can create more log forwarding profiles.

As an example, you may need to forward all critical severity events to an incident response team, but there may be a different team for different servers. You can create two log forwarding profiles, each with a rule that filters for all critical events and forwards them to a different email profile for each log forwarding profile. Then, attach these log forwarding profiles to two different rules.

Important note

While you can have many log forwarding profiles, only one profile can be attached to a security rule at a time.

Let's take a look at this with a practical example. In **Objects** | **Log Forwarding**, we create two different log forwarding profiles. As you can see in the following screenshot, they both forward all logs to Panorama/Logging Service, but also have a filtered rule for threat logs of critical severity with a different email action set:

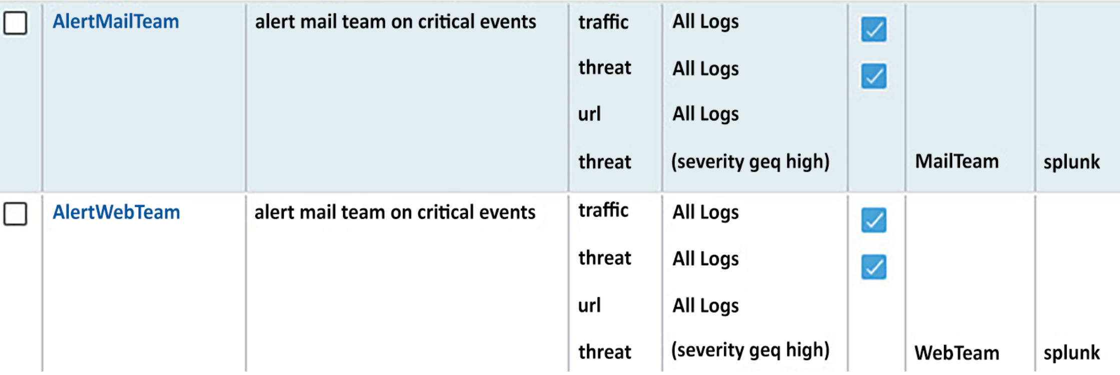


Figure 10.13: Log forwarding profiles with a filter

In **Policies** | **Security**, we create two new rules – one for inbound connections to the web server farm and one for inbound connections to the mail server farm – and attach the log forwarding profiles, as in the following screenshot:

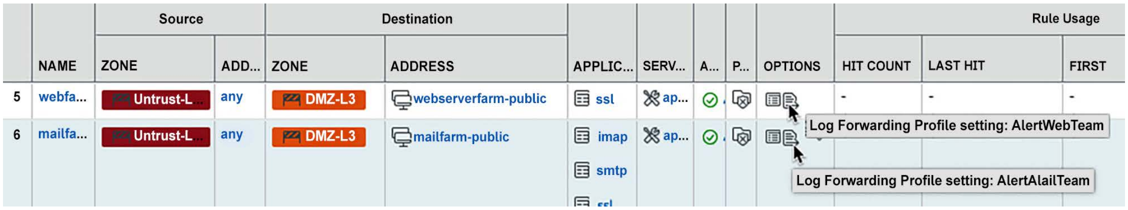


Figure 10.14: Log forwarding profiles attached to security rules

While it is good practice to forward all logs to Panorama, some applications, such as DNS, may generate so many logs that it may be better not to log these sessions at all, not even on the local firewall. The result is that no traffic log is written for these sessions, but any *threat* actions triggered by a security profile will still be logged in the threat log.

For these cases, you would disable the **Log at Session End** and **Log at Session Start** options, but still set a log forwarding profile so that threat logs are forwarded to Panorama, as in the following screenshot:

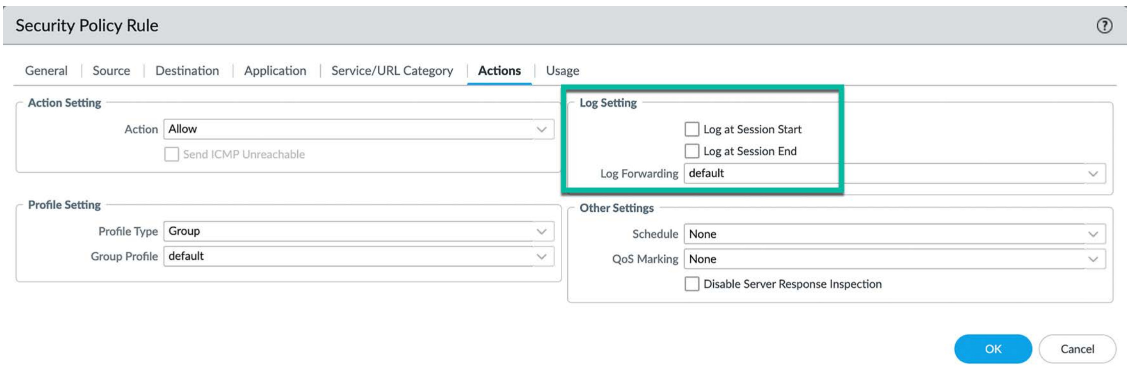


Figure 10.15: Log settings for a chatty security rule

Log at Session End will write the log once the session is completed, including all the session details, in a single log file. **Log at Session Start** will create a log entry at the start of each session and each time the application of the session changes. This means there will be several logs for a single session, so this will have an impact on the log volume. It can be useful as a troubleshooting tool to identify the stages a session goes through.

In this section, you have learned how to set up log collection and how to forward logs from the firewall to a remote server. In the next section, we'll look at how to filter logs.

Filtering logs

When you access any of the logs in **Monitor** | **Logs**, the sheer volume of information can be overwhelming and difficult to navigate at first. Once you learn how to master log filters, you'll be able to access the information you need quickly. Log filters are built by combining several statements via logical operators. Most fields in the log view are clickable and will automatically create a filter for you. You can then edit the filter and add more conditions to return the information you need.

For example, if you want to look at a 5-minute timeframe, you can click on any date in the log view twice and then edit both entries to look something like this:

```
( receive_time geq '2025/04/05 14:45:00' ) and ( receive_time leq '2025/04/05
14:50:00' )
```

`receive_time` is the parameter for when a log was received. `geq` stands for **Greater than or Equal**, while `leq` means **Less than or Equal**. So, this filter restricts the log view to anything received after `2025/04/05 14:45`, but before `14:50` of that same date.

> **Important note**
>
> `receive_time` is the time the log is received ("written") by the `logreceiver` process. This entry will usually be written at the **session's end**, so the session could have started much earlier. There is an additional column that you can activate that is called `generate_time`, which is when the log collection for a particular session is started at the **start of the session**.

You can add additional filters by clicking on and editing desired information, such as adding port `443` and sessions that have been allowed:

```
and ( port.dst eq 443 ) and ( action eq allow )
```

If you need to add a source, destination, or any IP address or subnet, you can add any of the following variants:

```
and ( addr.src in 192.168.27.253 )
and ( addr.dst in 192.168.27.253 )
and ( addr in 192.168.27.253 )
and ( addr.src in 192.168.27.0/24 )
and ( addr.src notin 192.168.27.253 )
```

For addresses, you can use `.src` or `.dst` to denote a source or destination or leave the extension to `addr` blank to indicate *anywhere*. For addresses, you can also set subnets of any size or add `not` to the operator to negate the statement.

For the `eq` operator, you can use `neq` to **negate**, and as a negative connector, you can use `AND NOT`, which allows plenty of flexibility as both the following statements have the same outcome:

```
( port.dst eq 443 ) and not (app eq ssl)
( port.dst eq 443 ) and  ( app neq ssl )
```

You can also add round brackets to combine statements in an `AND` or `OR` statement, as follows:

```
( port.dst eq 443 ) and  (( app eq facebook-base ) or ( app eq facebook-video
))
```

The preceding filters require port `443` to be used in the session, but their application can either be `facebook-base` or `facebook-video`.

Most filters use the `eq`, `neq`, `leq`, `geq`, `in`, and `notin` operators, but there are two exceptions:

- Some filters can have an `is present/is not present` statement by using `(x neq '')` or `(y eq '')` (double single quote marks).

  For example, `user.src neq ''` means a user must be present, and logs that don't contain a username will be filtered out.

- The `Flags` attribute uses `has` as it indicates whether the log entry has a flag set for a special condition – for example, PCAP, NAT, or SSL proxy – which is added to the log entry to indicate that a packet capture was stored for this session or threat and that the session was NATed or SSL-decrypted.

As you can see in the following screenshot, you can also use Filter Builder by clicking on the green + sign to the right of the filter bar:

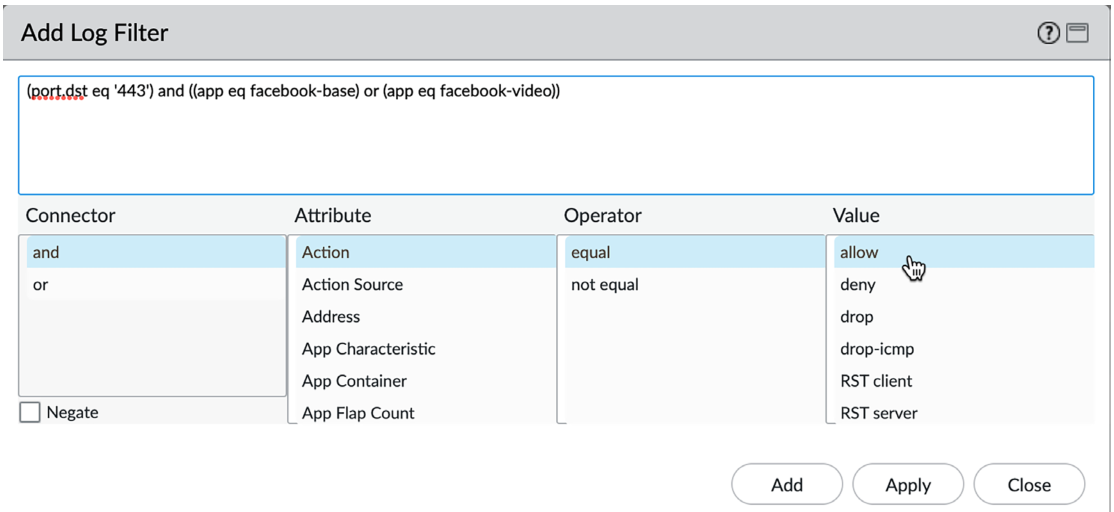


Figure 10.16: Using the log Filter Builder

To add a filter, do the following:

1. Select the **Connector**.
2. Choose the **Attribute** that you want to filter by.
3. Set the appropriate **Operator**.
4. Select or fill out the **Value**.
5. Click **Add**.
6. Click **Apply**.
7. Click **Close** or add another filter condition.

Once you have set up all the appropriate filters and found the log you are looking for, you can click on the little magnifying glass icon on the left-hand side of the log entry to drill down into the session details of the log. In the following screenshot, you can see that there is additional information about the session:

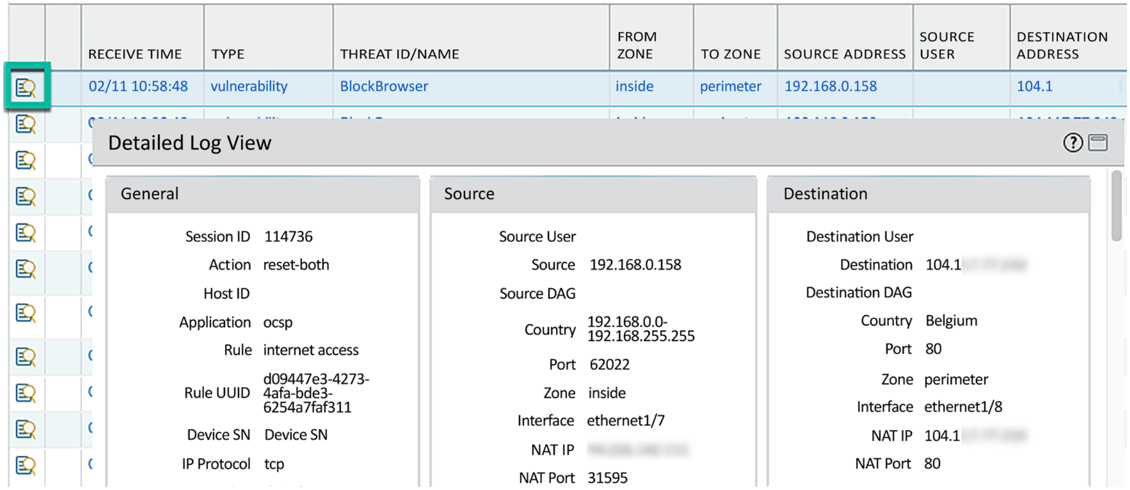


Figure 10.17: Detailed log view

At the bottom of the detailed view, there are related log files. Clicking on these will bring up those logs' details, as you can see in the following screenshot. This allows you to review any related log files to learn more about what is happening with the session.

In many cases, there will be a **traffic** log, a **url** log, and a **threat** log listed, so you can review all the details for each log from one window:

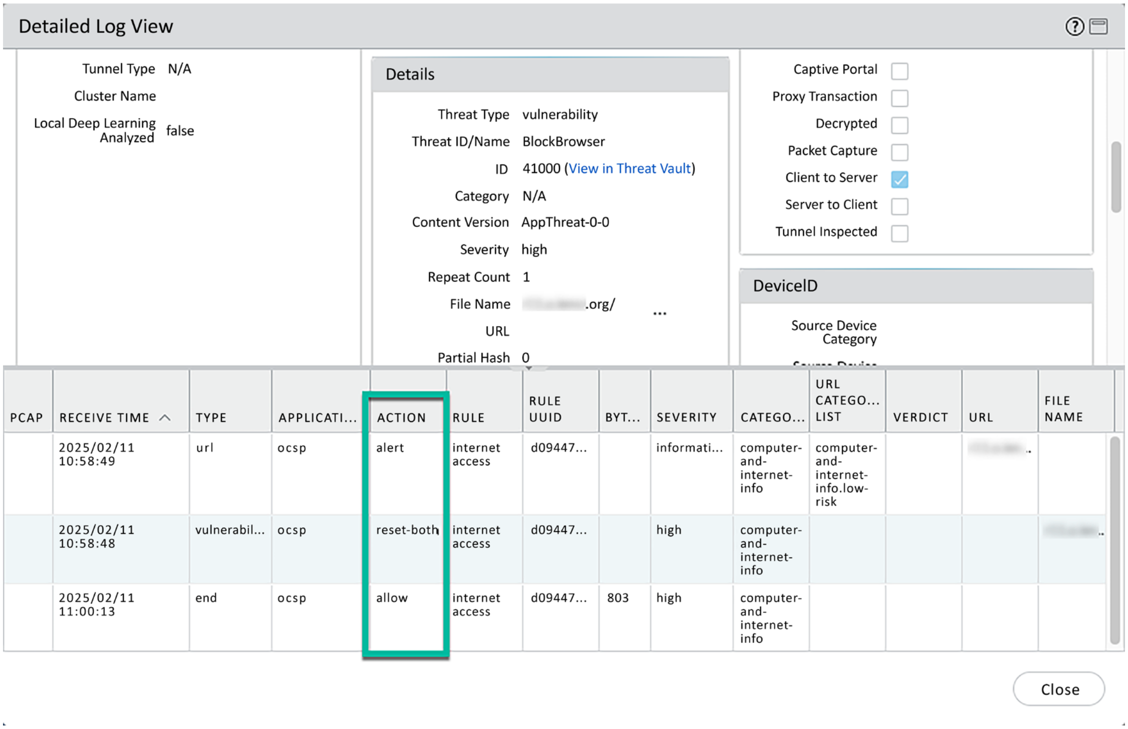


Figure 10.18: Related log file details in the detailed log view

You may have noticed that the action is different on different logs:

- The **traffic** log records what happened to the session at the network layer (in this case, the TCP session was ended naturally)
- The **threat** log records what happened at the application layer, which may be that a file was discarded, the user was redirected to a block page, or other actions
- The **url** log indicates that the website was allowed and logged

In some cases, threats may be expected for certain situations as they could simply be badly implemented services or intentionally changed protocols. For these situations, you can add exceptions by hovering over the threat name in the threat log and clicking on the arrow and then the **Exception** dialog.

As you can see in the following screenshot, you can then select the security profile to which you want to add an exception, and the IP (source or destination) to which you want to set the exception:

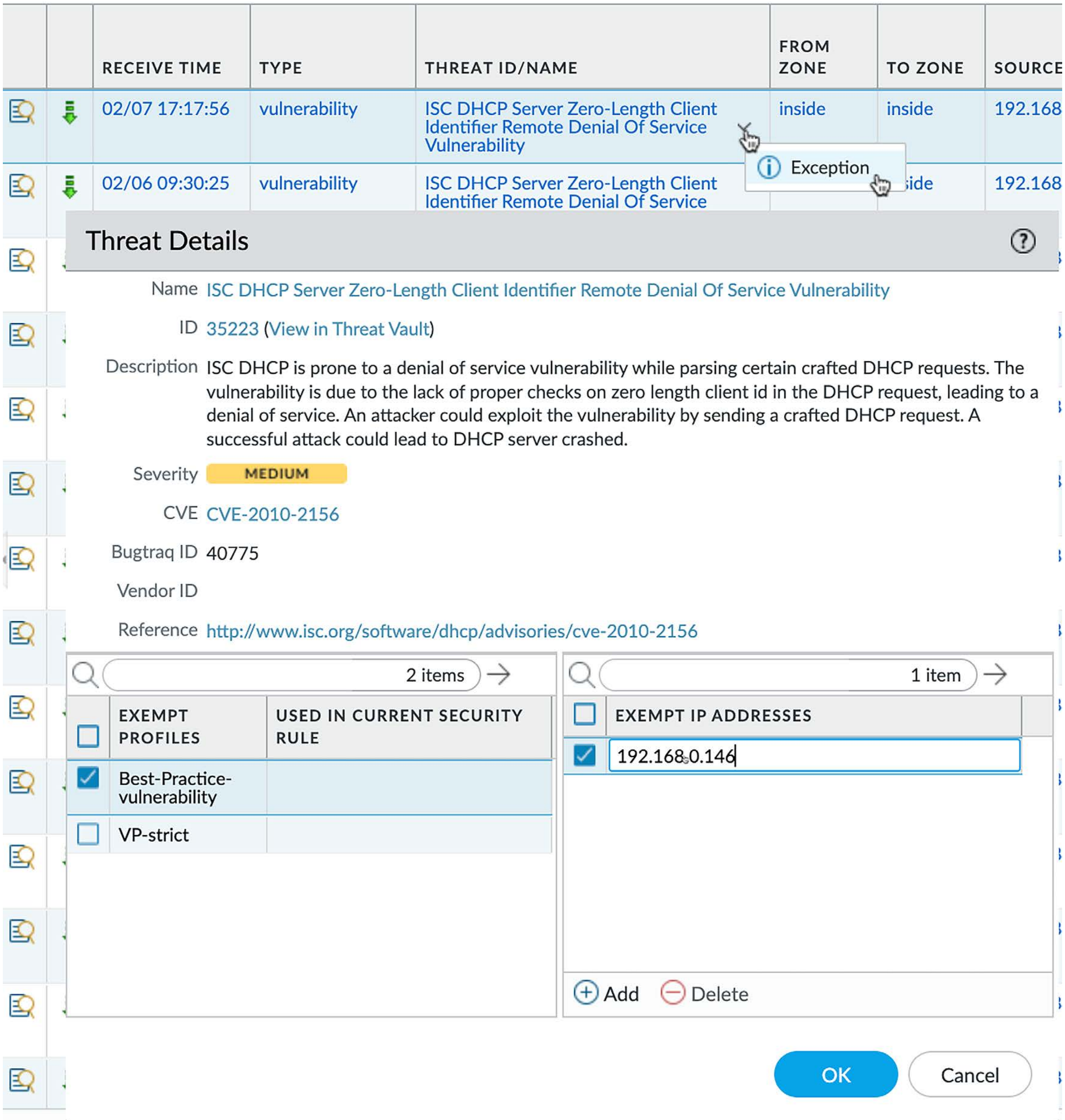


Figure 10.19: Adding exceptions for threats

For the last step, you may want to go to the security profile in **Objects** | **Security Profiles** and change the exception action associated with the vulnerability to something else (you can use the exception to change the behavior to allow or block exempted IP addresses, depending on your needs). In the following screenshot, you can see how you can change the action of the exception. By default, an exception is set to the **allow** action, which stops the logging of these events as well. Depending on your needs, you may opt to set the exception to **Alert** so that logs are still created.

You may notice, in the following screenshot, that the default action of this threat is **default (allow)**, but it was denied in the logs earlier. This means the security profile associated with the security rule that this session was hitting is configured to bypass default actions and apply different actions.

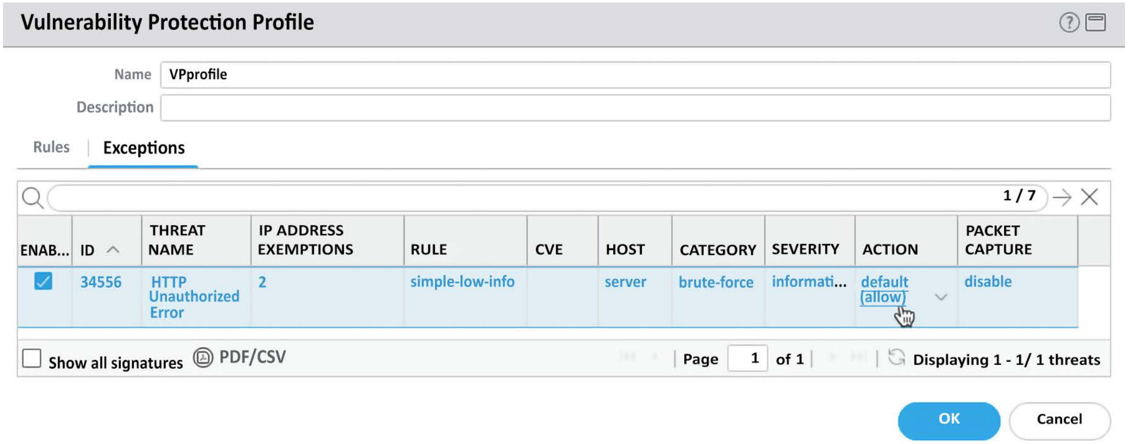


Figure 10.20: Changing the action of an exception

With the information you just learned, you should now be able to find logs that are relevant to your needs quickly and drill down into the finer details of each session, as well as find associated logs and add exceptions to threats where needed. In the next section, we will look at reporting and how to create custom reports.

Predefined reports and creating custom reports

Reports can be generated on the firewall to provide an overview at a glance about which applications are most popular or how many threats were detected for a certain timeframe.

The firewall has a set of **predefined reports** that run overnight and provide the most common insights. You can also build your own **custom reports** that contain the data that's most relevant to you. Let's look at predefined reports first.

Predefined reports

These reports provide a wide variety of information about the types of applications, threats, traffic, and URL filtering activity. They are set to run at 2 A.M., but if this is not a convenient time, you can change the start time in **Device** | **Setup** | **Management** | **Logging and Reporting Settings** | **Log Export and Reporting** and change **Report Runtime**. For Panorama Managed firewalls, go to **Templates** | **<Template Stack of the intended device> or <Template shared by all intended firewalls>** | **Device** | **Setup** | **Management** | **Logging and Reporting Settings** | **Log Export and Reporting**.

As you can see in the following screenshot, you can also disable some reports in the **Pre-Defined Reports** tab by unchecking them and committing the change:

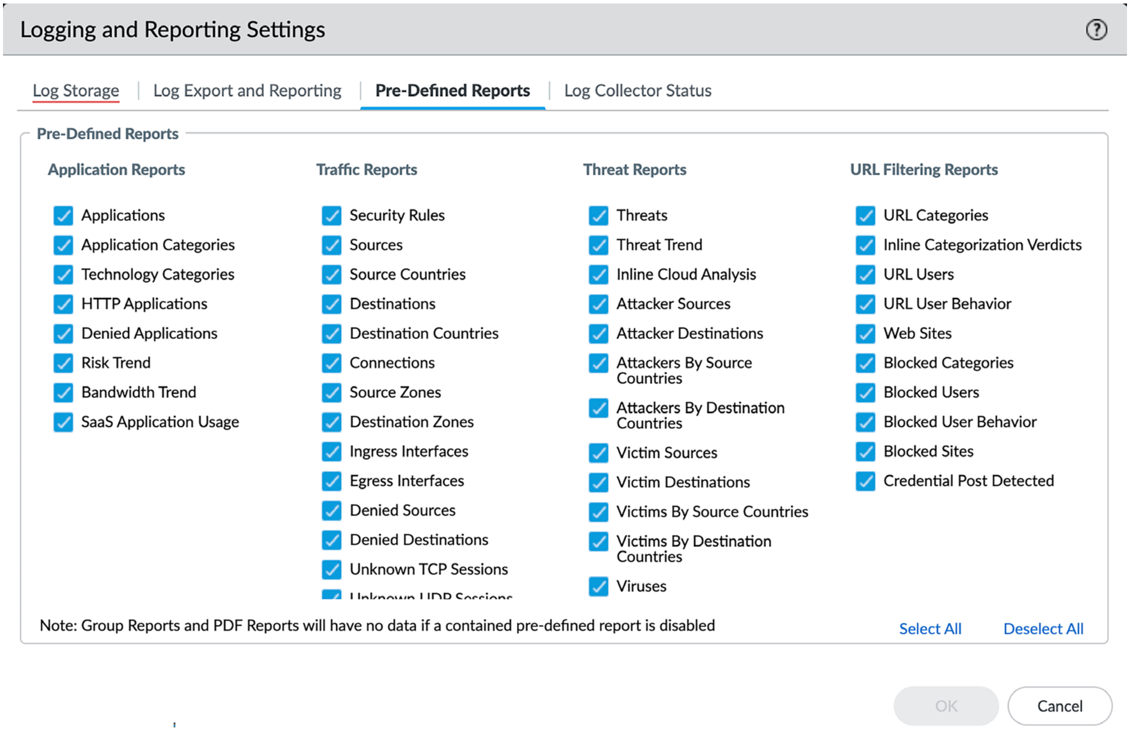


Figure 10.21: Enabling or disabling predefined reports

You can find the reports in **Monitor** | **Reports** (on Panorama, make sure the **Device Group** is set to **All**). On the right-hand side, you can select which report category you want to see, and then select one of the reports. As you can see in *Figure 10.22*, once a report type is selected, you can use the calendar at the bottom to select which day you want to review, which will then load the corresponding report on the left-hand side.

Entries seen in these reports can be clicked on to drill down into more detailed information, which will take you to the **Application Command Center** (ACC).

To find a report, follow these steps:

1. On the right-hand side, open one of the section titles you're most interested in like **Custom Reports**, **Application Reports**, **Traffic Reports**, etc. When the title is expanded, find the report you want to see.
2. Below the reports, select the date for which you want to see the report. Dates with available reports will be in bold, and dates with no reports will be in regular text.
3. Once the report is loaded, you can click on the blue text where you want to see additional details.

These steps are shown in the following screenshot:

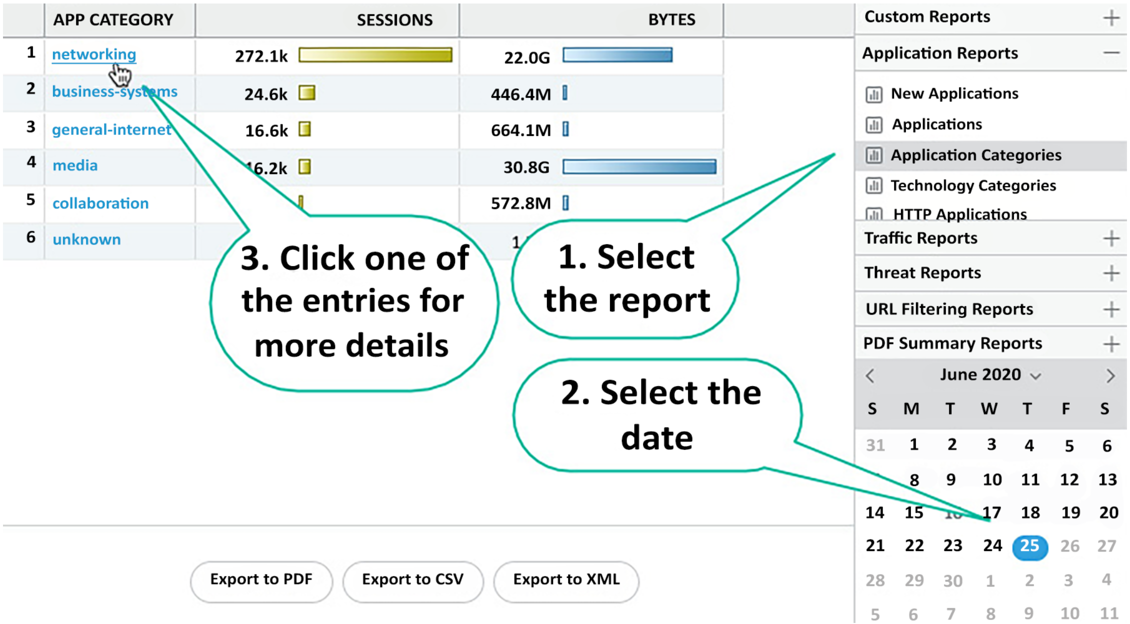


Figure 10.22: Predefined reports

Custom reports

In **Monitor** | **Manage Custom Reports**, you can build custom reports. There are two major sources of information that can be used to generate reports – **Summary Databases** and **Detailed Logs**:

- **Summary databases** consist of pre-summarized statistics on applications, traffic, threats, and tunnels, which the data plane collects and stores. Reports created from these databases are faster to generate but may not have all the columns that a log has.
- **Detailed log databases** are the actual log files that are parsed and data extracted to generate the report. These reports take longer to generate and may see increased management plane CPU usage during generation, but can contain more information than the summary databases.

The predefined reports can be loaded as a template so that you can fine-tune a type of report if you like the original but want more granularity or, for example, an additional column.

If you want a custom report to run periodically, you must enable **Scheduled**.

Let's create an example report:

1. Add a new report and select **Load Template** to load `Top-Destinations`.
2. The template automatically loads the traffic summary database and loads the columns with **Destination Address**, **Destination User**, and **Bytes and Sessions**, as well as sets **Sort By** to **Sessions**.
3. Set **Scheduled**.
4. Set **Time Frame** to **Last Calendar Week**.
5. You can click **Run Now** to see what the report will look like. You'll see that the report simply shows which IP addresses are the most popular destinations based on the session count.
6. Now, add the **Application** column by selecting it from the available columns and clicking on the little + sign.
7. Now, click on **Run Now** again and compare the two reports. The new report still has the most popular destinations sorted by session count, but now, the destinations are split up based on **Application**.
8. Now, set **Group By** to **Application** for **10 Groups**.

   Your custom report will now look as in the following screenshot:

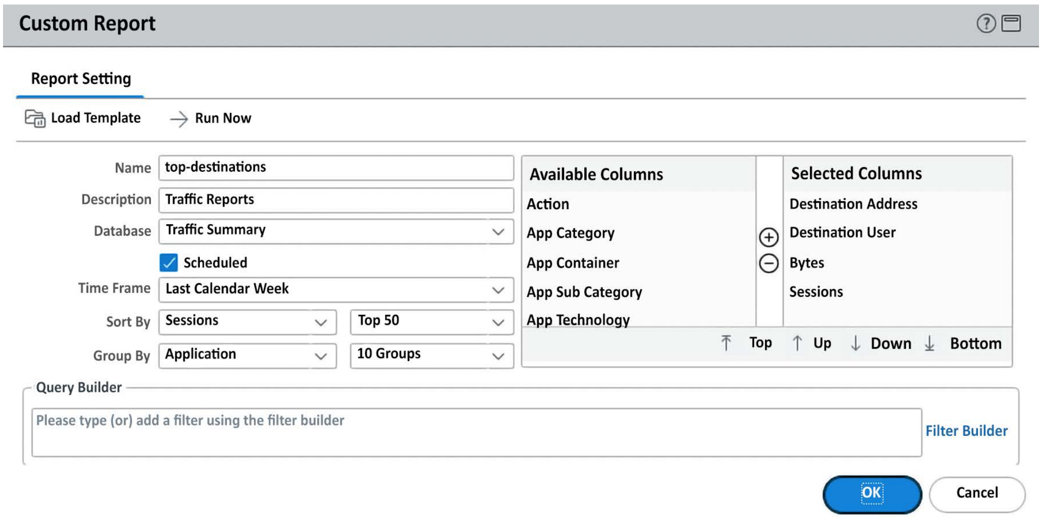


Figure 10.23: Custom report from a template

9. If you hit **Run Now** again and compare the report, you will notice that the destinations are now sorted in groups per application, with the top destination for each application sorted by session count.
10. Click **OK**.

You can also add filters to get more granular reports:

1. Add a new report and call it `Threats per Week`.

2. Select the **Threat Summary** database.
3. Set **Scheduled.**
4. Change **Time Frame** to **Last Calendar Week.**
5. From the columns, select **Count**, **Action**, **Severity**, **Threat/Content Name**, **Application**, **Source Address**, and **Source User**.
6. Sort by **Count** and **Top 10**.
7. Group by **Application** and **10 Groups**.
8. To prevent this report from getting filled to the brim with informational severity threats, click on the **Filter Builder** option in the bottom-right corner.
9. Set a **And Severity Greater than or Equal high** filter, then click on **Add**, and then **Apply**.

   Your custom report will look as in the following screenshot:

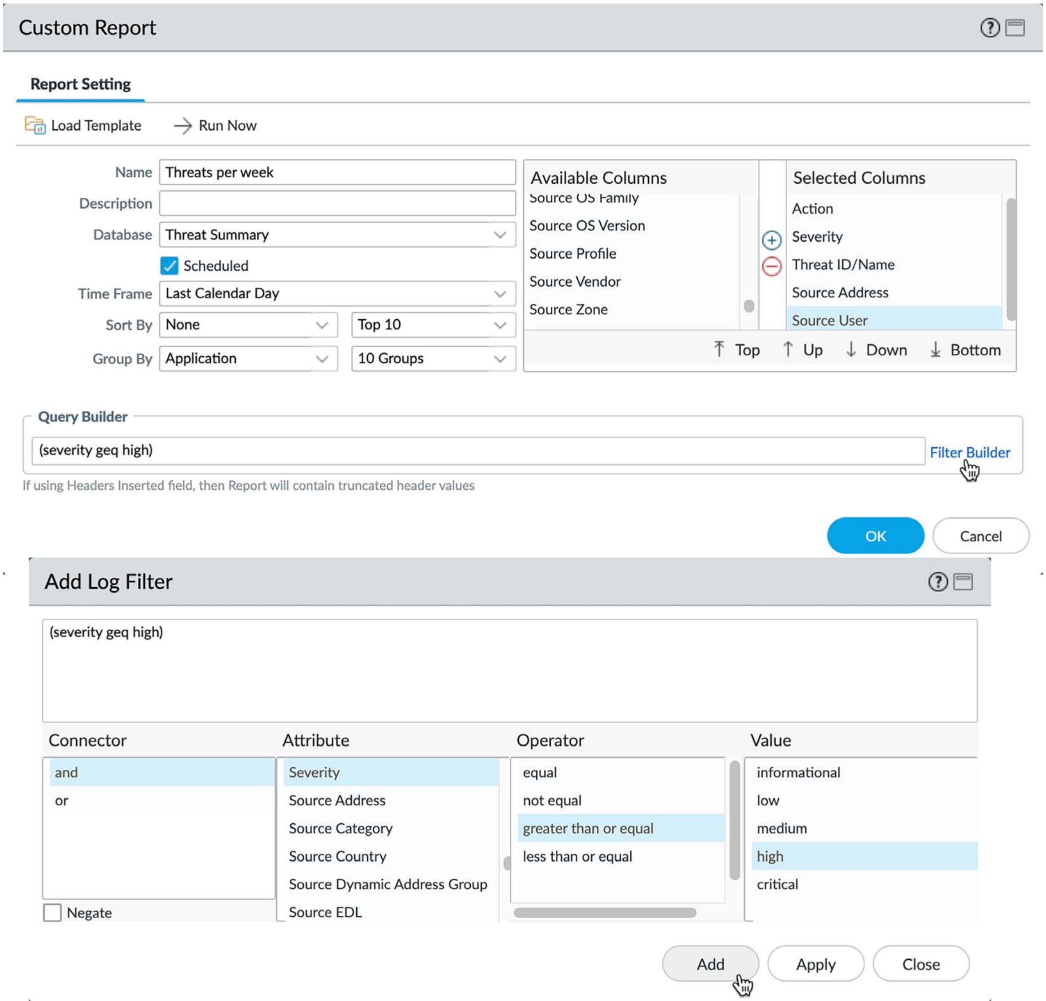


Figure 10.24: Custom report with additional filters

10. Click **Run Now** to get a preview of the report. Only high or critical vulnerabilities will show up in the report.
11. Click **OK**.

To be able to send out emailed reports, we still need to create a report group or a PDF summary.

In **Monitor** | **PDF Reports** | **Manage PDF Summary**, you can create a new PDF summary. A new PDF summary will have all the threat reports selected and part of the **Application Reports**. You can disable and add any predefined or custom reports, as well as **Trend Reports**, which are only available in PDF summaries. So, go ahead and create a new PDF summary:

1. Remove all the predefined reports.
2. Add all the trend reports.
3. Name the report `Trends`.
4. Click **OK**.

The PDF summary will look as in the following screenshot:

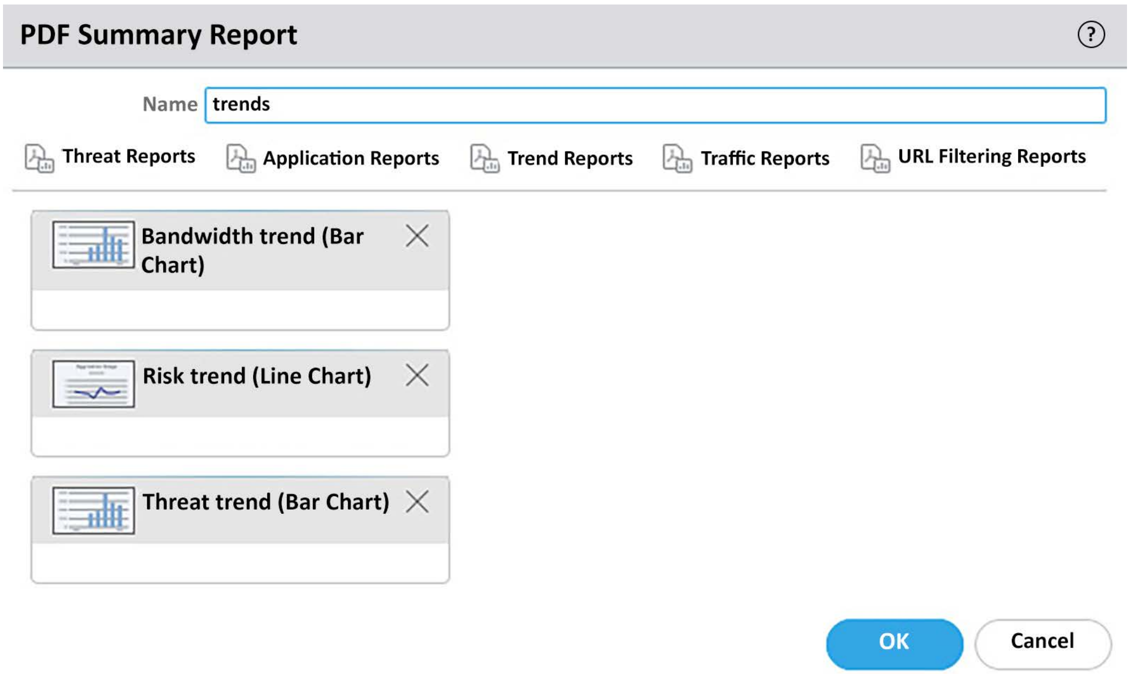


Figure 10.25: PDF summary report creation

In the report groups, you can group predefined and custom reports, and you can also add summary PDFs:

1. Add a new report group and name it `Weekly Report`.
2. Select **Title Page** and set **Title** to `Weekly Report`.
3. Add the PDF summary report.

4. Add the two custom reports (make sure to select the reports listed under **Custom Report** and not the ones under **CSV**).
5. Add any reports you'd like to get a weekly report on.

   The **Report Group** page will look similar to the following screenshot. Add additional reports as you wish:

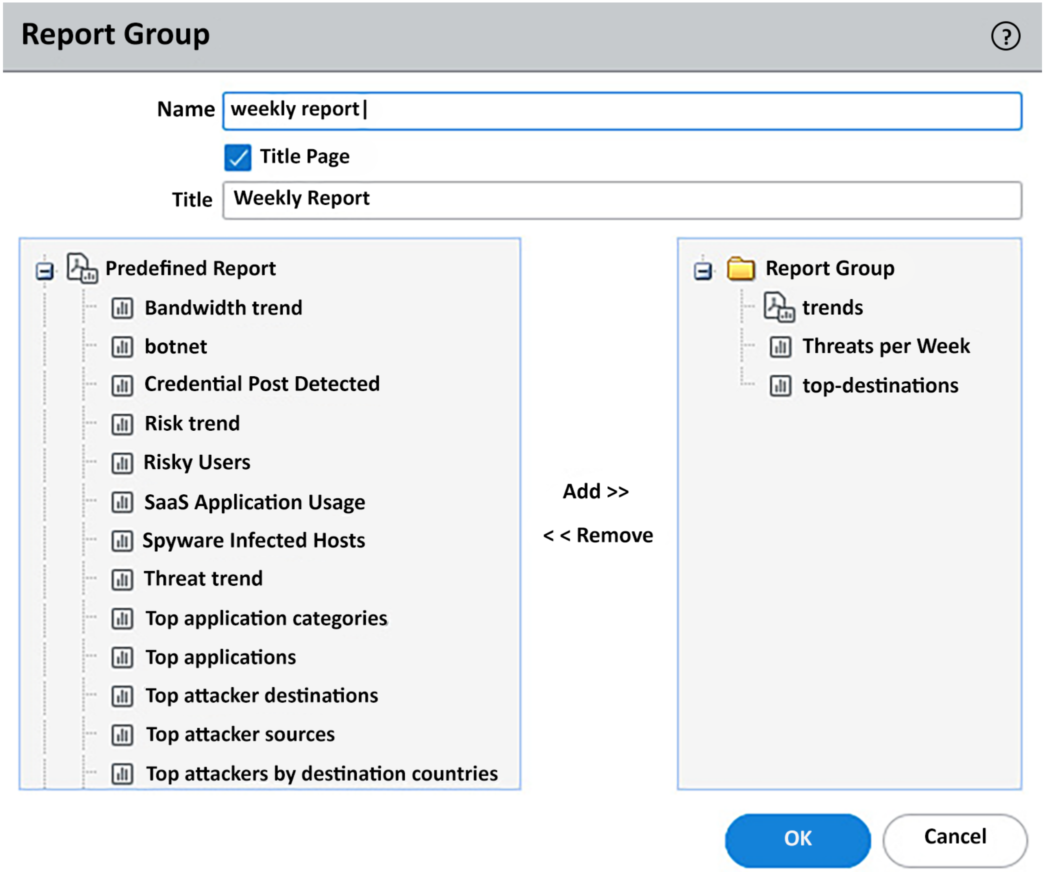


Figure 10.26: Report Group

6. Click **OK.**

The last step is to create an email scheduler in **Monitor | PDF Reports | Email Scheduler:**

1. Set **Name** to `Weekly Report`.
2. Select the `Weekly report` report group.
3. Select one of the email profiles you created earlier, or create a new one for these reports.
4. Set **Recurrence** to `Every Monday`.

The **Email Scheduler** page will look similar to the following screenshot:

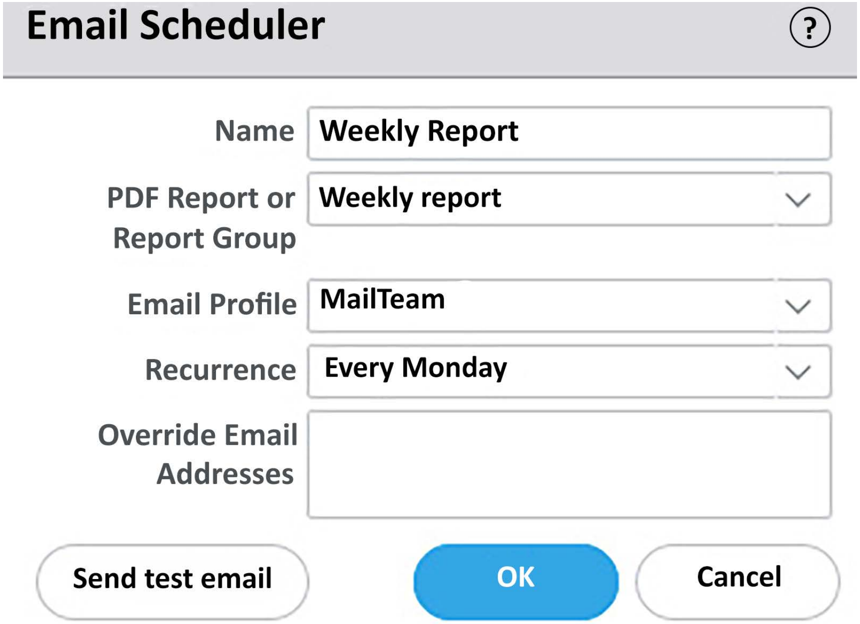


Figure 10.27: Email Scheduler for reports

5. Click **OK**.
6. Commit the changes.

The system will now start collecting statistics to create custom reports and a summary PDF. The resulting output will be emailed every Monday. The first time that this report will be completed and emailed could take more than a week as custom reports take a full Monday to Sunday week to create a full report (so, if today is Friday, then the first report containing statistics from the custom reports will arrive in 10 days). This applies to all recurrences, like `Last Calendar Week` or `Last Calendar Month`, where the report can only run once a full week or full month has passed.

There are also two *on-the-fly* reports intended to supply information about a user or SaaS applications:

- **User Activity Report** creates a report regarding user or group activities. You only need to supply the username or group name and a timeframe for the report to be generated (and select whether you want to see detailed browsing information, which could be a privacy concern).
- **SaaS Application Usage** lets you run a report on the past several days for source users and zones, or only source zones on the usage of SaaS applications.

The **SaaS Application Usage** report will mention sanctioned and unsanctioned SaaS applications, as shown in the following screenshot. To mark applications as **Sanctioned**, open **Objects** | **Applications** and look for the applications you want to mark. In the **Applications** dialog, hit **Edit** in the tags, then select **Sanctioned**, and then click **OK**:

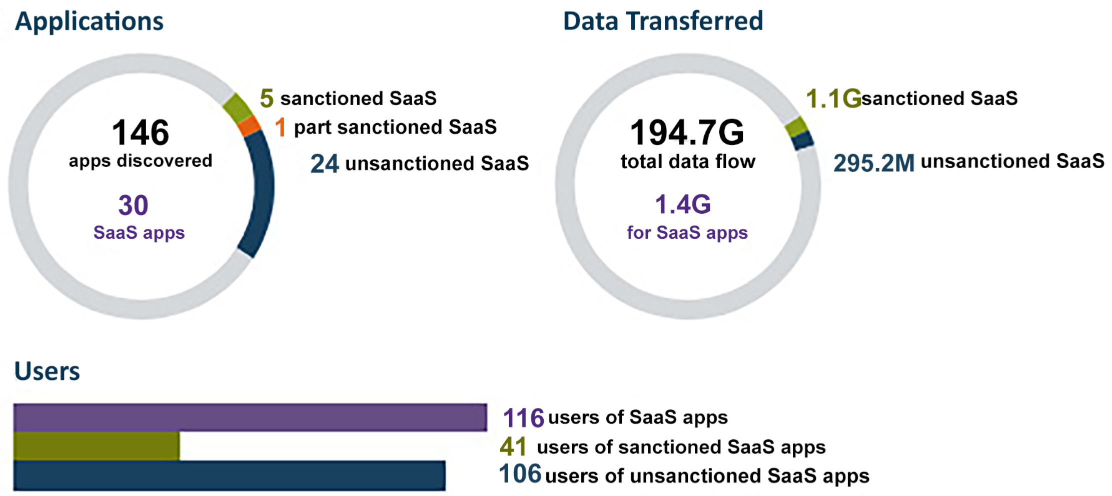


Figure 10.28: Sanctioned and unsanctioned applications in a SaaS report

You can now build and schedule your own reports. You may have noticed that you can drill down, or zoom into, the reports by clicking on addresses, applications, threats, or other details, which then redirects you to the **Application Command Center** (ACC).

Using the Application Command Center

As opposed to reports that run on a daily basis, the ACC is a live correlation tool that lets you get a quick look into what is happening in your network by using simple graphs that you can drill down into for more information. There are four default tabs:

- **Network Activity** gives you an overview of all the applications seen in the specified timeframe, their byte count, the session count, the threat count, and the number of users. If you scroll down, you will see more detailed source and destination graphs and which rules have been hit most.
- **Threat Activity** gives you a breakdown of all the types of threats and how many times they were seen.
- **Blocked Activity** shows which applications have been blocked due to threats, content, or URL actions.
- **Tunnel Activity** is used to report on tunnel inspection for GRE, GPRS, and non-encrypted IPSec.

You can also add a tab and create a page with all the widgets you like in one single pane, which may be useful if you want to be able to keep tabs on something more specific. As shown in the following screenshot, when you are investigating an entry, you can either create global filters from the left-hand side filter creator, or you can click on the little arrow that appears when you hover over any item that can be filtered:

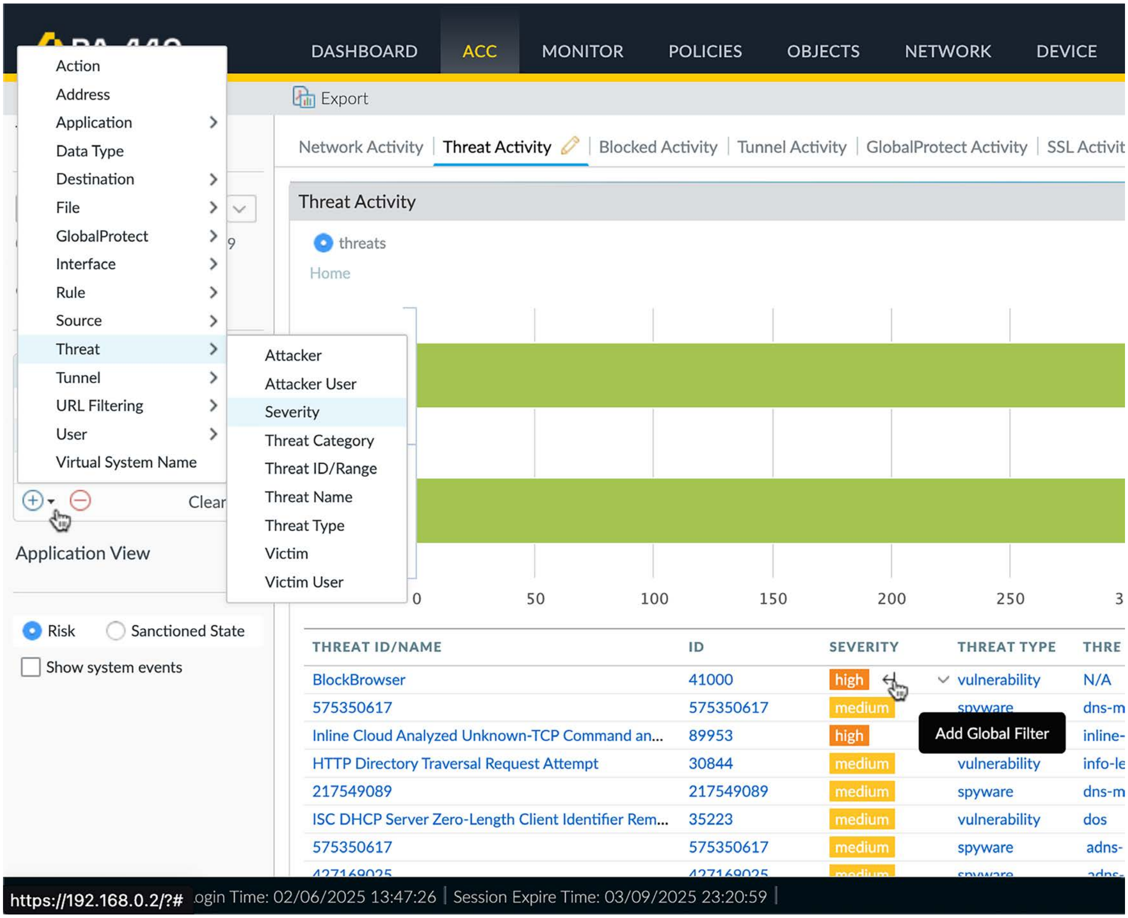


Figure 10.29: Adding filters in the ACC

Once you've drilled down to the information you want to investigate and you want to access the associated logs, you can use the **Jump to Logs** quick link, which will take you to the log viewer with the appropriate filters already filled in, as you can see in the following screenshot:

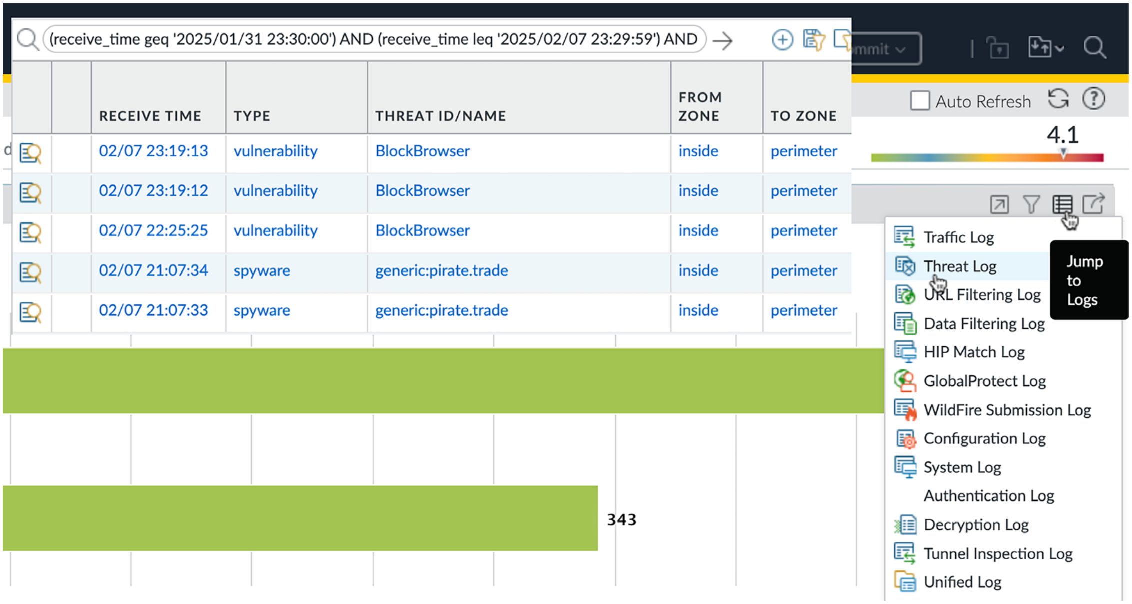


Figure 10.30: The Jump to Logs link in the ACC

As another example, as you can see in *Figure 10.31*, there was a peak in my network traffic in the outbound direction at this particular date and time (note that although the following screenshots are a little bit older than the previous ones, the output still looks the same):

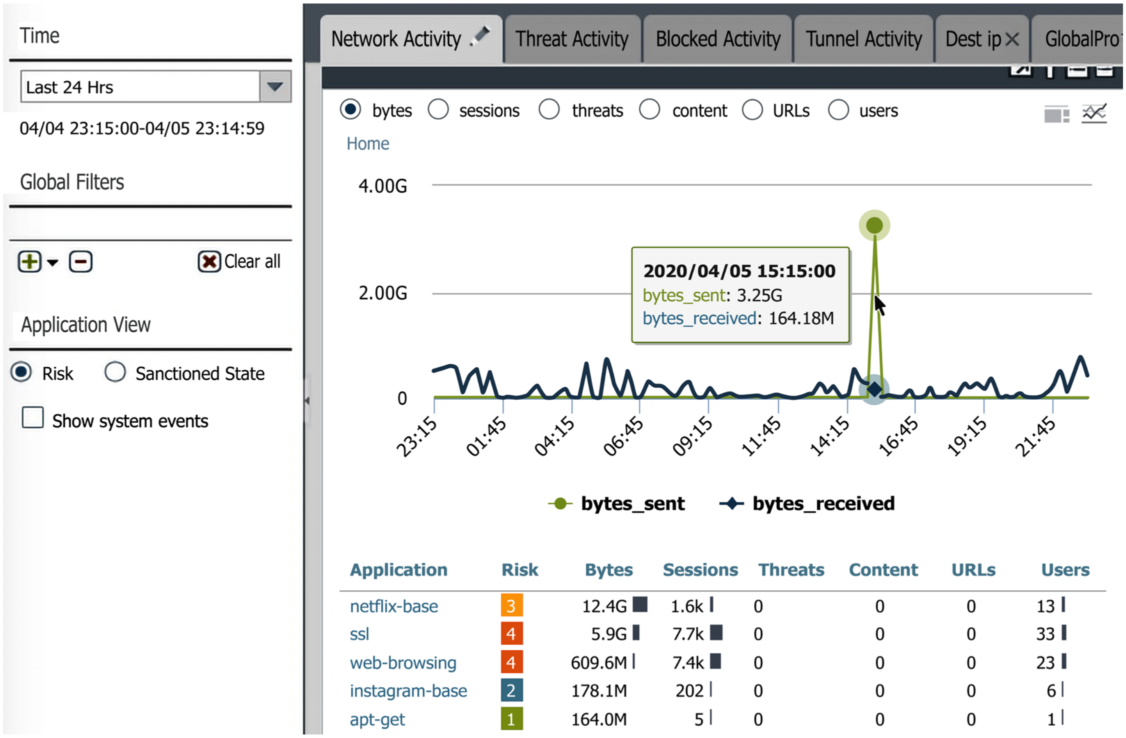


Figure 10.31: Network Activity in the ACC

If I scroll down to the source and destination IP address widgets, I can see that there is a lot of traffic flowing to `192.168.27.5`, which is a VM server in my **DMZ** (**demilitarized zone**).

So, as you can see in the following screenshot, I can click on the arrow to the right of the IP to add it as a filter:

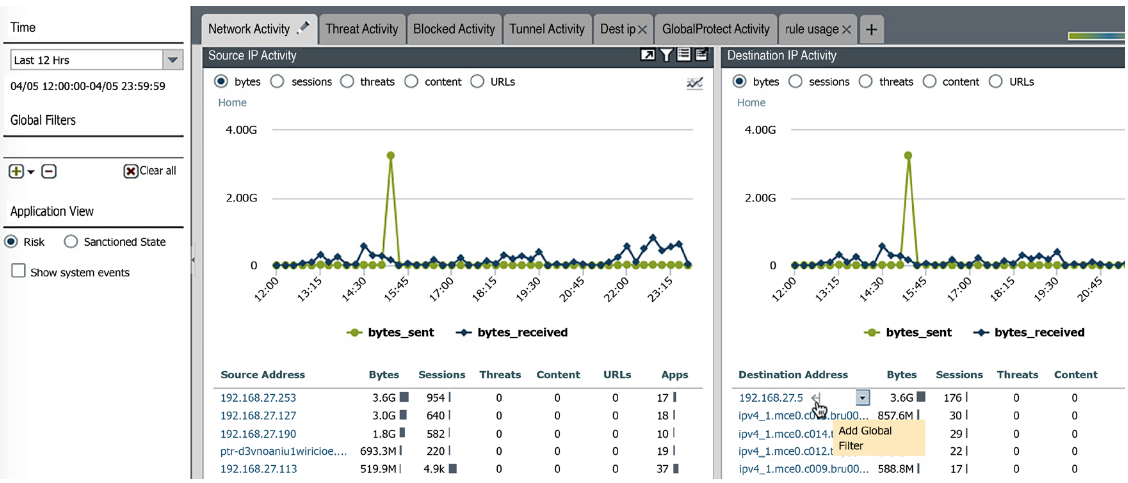


Figure 10.32: Reviewing the source and destination IP in the ACC

After applying the filter, as you can see in the following screenshot, I can see that the application used to transmit this volume was `ssl`, and the sender was `192.168.27.253`, which is my laptop:

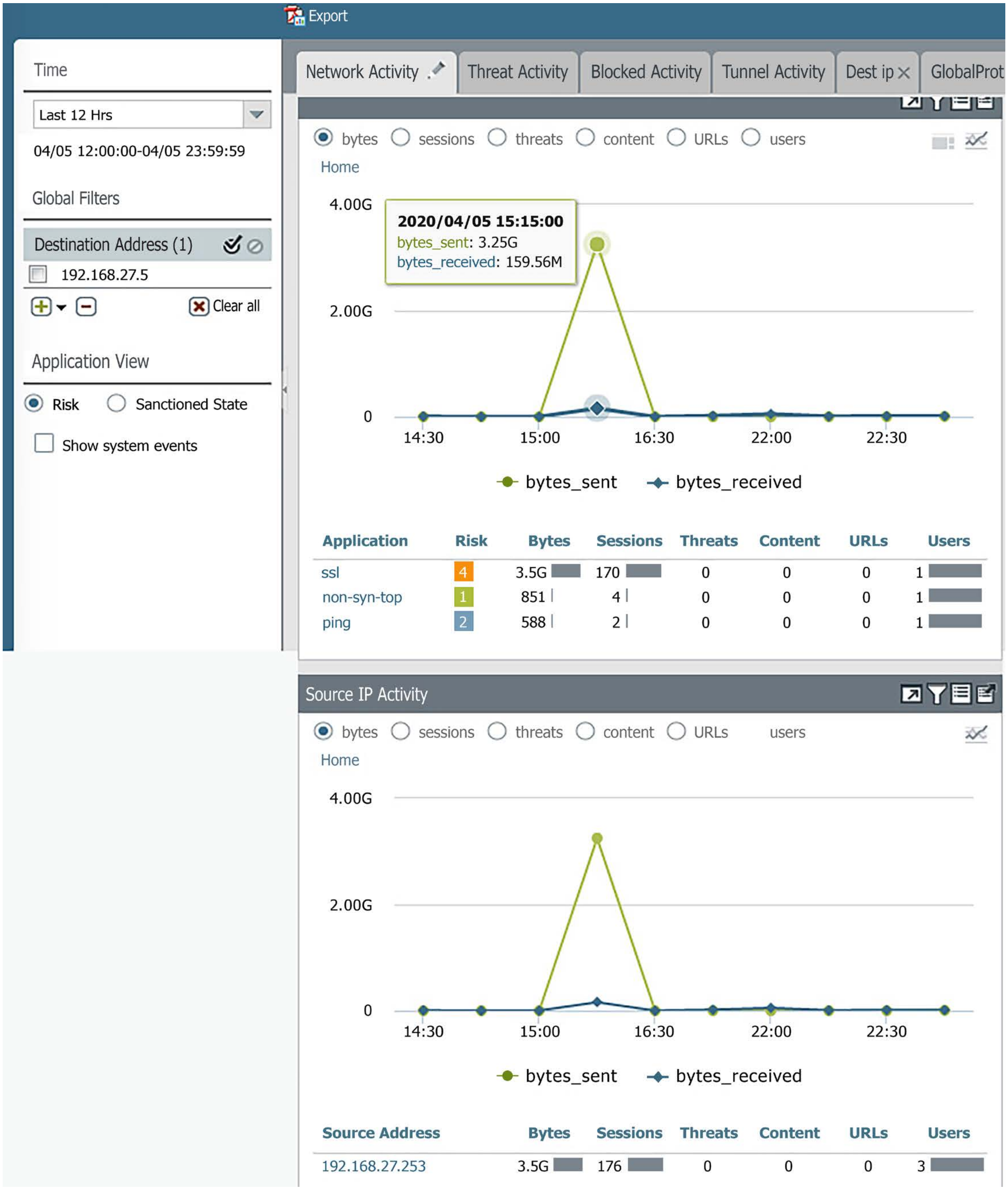


Figure 10.33: Filtered view in the ACC

You can now use the ACC to gain an eagle-eye view of the things happening on your network. I do encourage you to create a custom tab and add widgets. One of my favorite combinations of widgets is using **Rule Usage**, **Rules Allowing Apps On Non Standard Ports**, and **Security Policies Blocking Activity** to keep track of my security policy and help me make tweaks where needed.

Summary

In this chapter, you learned all about how logging works and how to scale and set up the infrastructure to capture logs. You also learned some methods to send logs out to Palo Alto Networks logging appliances or cloud instances. You learned how to set up forwarding to syslog servers and send out emails on certain events. Finally, you learned how to leverage filters to drill down into detailed information so that you can quickly find what you need in the ACC and how to manage the built-in reports and create custom ones.

If you're studying for the PCNSE, take note of the different ways in which logs can be forwarded to external devices (Panorama, SLS, syslog, and so on) and be able to identify the different types of logs (traffic, threat, system logs, and so on).

In the next chapter, we will be learning how to set up site-to-site and GlobalProtect VPN tunnels and how to create custom applications and threats.

11

Virtual Private Networks (VPNs)

There are several ways of connecting devices in a secure way. In this chapter, we will learn about site-to-site VPNs and the challenges you may encounter when connecting to different vendors. Palo Alto Networks firewalls currently support the following protocols:

- **Generic Routing Encapsulation** (**GRE**) is a fairly old protocol that is not very secure but can be useful if legacy devices need to be connected to the firewall to provide rudimentary security to the encapsulated packets.
- **Internet Protocol Security** (**IPSec**) is the de facto tunneling protocol between remote sites and can be used for very strong encryption.
- **Secure Socket Layer** (**SSL**), which is really **Transport Layer Security** (**TLS**), is used to connect endpoints over a *network-friendly* protocol if IPSec is unavailable.

We will also learn how to set up a GlobalProtect user VPN and verify whether hosts connecting remotely are in a permissible state to enter the network or need to be quarantined.

More specifically, we're going to cover the following main topics:

- Configuring GRE
- Configuring an IPSec site-to-site VPN
- Configuring GlobalProtect

By the end of this chapter, you'll be able to connect remote locations and remote users to a data center or central office in a secure way.

Technical requirements

In this chapter, we will be covering remote connections and protection from inbound connections. If you have a lab environment where you can simulate setting up VPN connections to other devices or produce incoming connections from a client, this will help greatly in visualizing what is being explained.

Configuring GRE

To set up GRE tunnels, you can set up a connection in **Networks** | **GRE Tunnels**. All you need to configure is the following:

- **Name** can be any description.
- **Interface** is the interface the GRE tunnel is bound to as the source interface.
- **Local Address** is the source IP associated with the source interface.
- **Peer Address** is the IP for the remote peer.
- **Tunnel Interface** is required as a routing destination for the remote network.
- **TTL** has a default value of 64.
- **Copy ToS Header** can be used to copy the **Type Of Service** (**ToS**) header from the internal IP header to the external IP header.
- **ERSPAN** can be enabled to have the firewall decapsulate the **Encapsulated Remote Switched Port ANalyser** (**ERSPAN**) sent through the GRE tunnel. This allows a switch to be configured to send mirrored traffic through a GRE tunnel to the firewall for inspection and advanced logging, similar to a TAP port.
- **Keep Alive** settings are needed because, by default, the firewall will send a keepalive every 10 seconds. **Retry** and **Hold Timer** are used to retry and re-establish a connection if the keepalive fails.

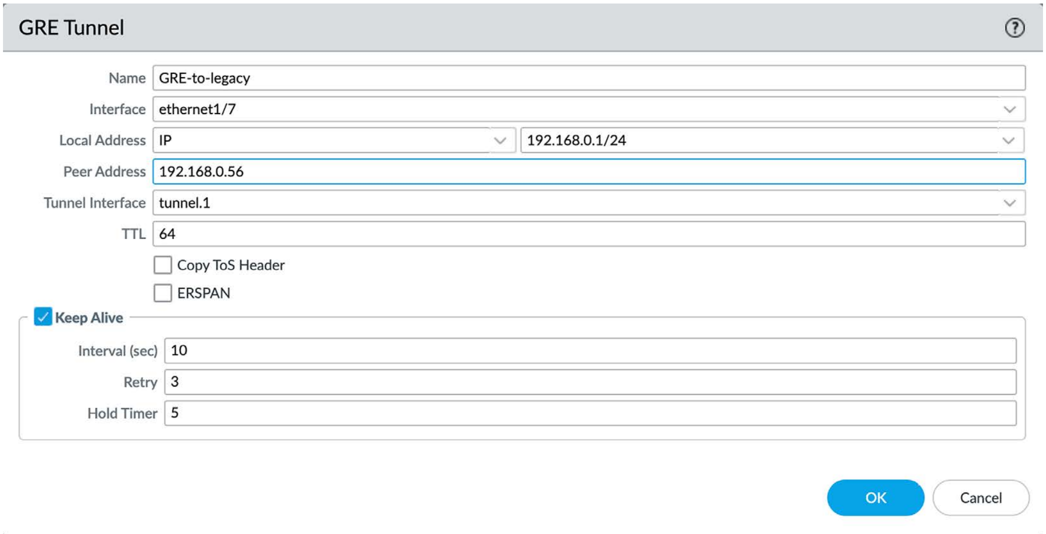


Figure 11.1: GRE tunnel configuration

Set up the same configurations on the remote end to get it going. In the virtual router, add a route for the remote subnet to the above tunnel interface and add a security rule that allows traffic to or from the tunnel interface zone.

In the next section, we will set up IPSec connections and learn about the different ways to implement the configuration.

Configuring the IPSec site-to-site VPN

Before you can set up a VPN tunnel between two peers, you first need to agree on the cryptography settings that will need to be applied on both sides so that the tunnel can be negotiated. If the remote end is not under your control, you will need to reach out to your peer to agree on which configuration to use.

In the first phase (phase 1) of the negotiation, both peers authenticate one another through the **Internet Key Exchange** (**IKE**) process.

Once the authentication has been established, an IPSec **Security Association** (**SA**) is created on both sides that contains all the parameters needed to set up the phase 2 IPSec VPN tunnel.

The phase 1 crypto profile can be created in **Network** | **Network Profiles** | **IKE Crypto.** As you can see in the following screenshot, there are three default profiles already present with the following settings:

	NAME	ENCRYPTION	AUTHENTICATION	DH GROUP	KEY LIFETIME
☐	default	aes-128-cbc, 3des	sha1	group2	8 hours
☐	Suite-B-GCM-128	aes-128-cbc	sha256	group19	8 hours
☐	Suite-B-GCM-256	aes-256-cbc	sha384	group20	8 hours

Figure 11.2: IKE crypto profiles

The default profile represents the most common cryptographic scheme and should not be used unless the remote peer does not know which cryptographic profile is configured, or if the remote end is a legacy appliance with limited cryptographic capabilities.

The **Suite-B** profiles (already superseded by the **Commercial National Security Algorithm** (CNSA) Suite) are NSA-recommended cryptographic settings. These profiles contain good options and are recommended for most situations, but use your judgment and confer with the remote peer about which cryptographic options are best suited for phase 1.

Important note

It is recommended to use **Suite-B-GCM-128** for small remote devices and **Suite-B-GCM-256** for larger peers to optimize workloads caused by the crypto ciphers.

The latest recommendations can be found at `https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm`. Ideally, review which settings are supported on both devices and pick a set that meets the highest possible security standards.

Setting up a (phase 1) IKE Crypto profile

To add a new IKE Crypto profile, review the options in the following steps:

1. Click on **Add** and, in the **General** tab, name the profile so that you can easily identify it.

2. Set **DH group**:
    - **DH Group 1:** `768-bit group`
    - **DH Group 2:** `1024-bit group`
    - **DH Group 5:** `1536-bit group`
    - **DH Group 14:** `2048-bit group`
    - **DH Group 15:** `3072-bit group`
    - **DH Group 16:** `4096-bit group`
    - **DH Group 19:** `256-bit elliptic curve group`
    - **DH Group 20:** `384-bit elliptic curve group`
    - **DH Group 21:** `521-bit elliptic curve group`

    Alternatively, select one of the PQ KEMS, as seen in *step 7*, for true Post Quantum Key Exchange Mechanism IKEv2 deployment.

    For maximum compatibility, it is best practice to use a traditional DH group here and only use the PQ KEM in the additional rounds in *step 7*.
3. Set **Authentication**:
    - `md5` (available for legacy compatibility; do not use unless necessary)
    - `sha1` (available for legacy compatibility; do not use unless necessary)
    - `sha256`
    - `sha384`
    - `sha512`
4. Set **Encryption**:
    - `Des` (available for legacy compatibility; do not use unless necessary)
    - `3des` (available for legacy compatibility; do not use unless necessary)
    - `aes-128-cbc`
    - `aes-192-cbc`
    - `aes-256-cbc`
5. Set **Key Lifetime** in hours (`8` is the industry default).
6. **IKEv2 Authentication Multiple** lets you set the number of IKEv2 rekeys that are allowed before the gateway is forced to start a fresh authentication. This will hinder snooping efforts.
7. In the **Advanced Options** tab (available in PAN-OS 11 and higher) we can set IKEv2 **Post Quantum Key Exchange Mechanisms** (**PQ KEMs**) in up to 7 additional rounds. These additional rounds allow even more resilience against decryption in case one of the PQ KEMs does become vulnerable, by adding more mechanisms. Select one of the following mechanisms for one or more of the additional rounds (ensure the peer supports additional rounds):
    - **Kyber-512**
    - **Kyber-768**

   - **Kyber-1024**
   - **Bike-L1**
   - **Bike-L2**
   - **Bike-L3**
   - **Frodokem-640-aes**
   - **Frodokem-640-shake**
   - **Frodokem-976-aes**
   - **Frodokem-976-shake**
   - **Frodokem-1344-aes**
   - **Frodokem-1344-shake**
   - **Hqc-128**
   - **Hqc-192**
   - **Hqc-256**
   - **Classic-mcelliece-348863**
   - **Classic-mcelliece-348863f**
   - **Ntruprime-sntrup761**

8. Click **OK**.

> Do not use `md5`, `sha1`, `des`, or `3des` unless you are required to connect to a legacy device that does not support more modern algorithms, as all of these options are easily defeated by modern cracking and decryption tools.

> **Important note**
>
> PQ KEMs covered in RFC 9242 and RFC 9370 allow for large post quantum keys and multiple rounds of key exchange that protect encrypted traffic from being decrypted by future quantum computers (i.e., not relying on basic mathematics that can be brute forced but, rather, on logic that needs to be known by both sides).
>
> More information about these key exchange mechanisms can be found on these websites:
>
> - `https://pq-crystals.org/kyber/`
> - `https://csrc.nist.gov/csrc/media/Events/2024/fifth-pqc-standardization-conference/documents/papers/a-lean-bike-kem.pdf`
> - `https://csrc.nist.gov/CSRC/media/Presentations/FrodoKEM/images-media/FrodoKEM-April2018.pdf`
> - `https://pqc-hqc.org/doc/hqc-specification_2024-02-23.pdf`
> - `https://classic.mceliece.org/`
> - `https://ntruprime.cr.yp.to/`

The phase 2 cryptographic profiles can be found in **Network** | **Network Profiles** | **IPsec Crypto**. As you can see, there are three preconfigured profiles that you can opt to use if they suit your needs:

☐	NAME	ESP/AH	ENCRYPTION	AUTHENTICATION	DH GROUP	LIFETIME	LIFESIZE
☐	default	ESP	aes-128-cbc, 3des	sha1	group2	1 hours	
☐	Suite-B-GCM-128	ESP	aes-128-gcm	none	group19	1 hours	
☐	Suite-B-GCM-256	ESP	aes-256-gcm	none	group20	1 hours	

Figure 11.3: IPSec Crypto profiles

Setting up a (phase 2) IPSec Crypto profile

In IPSec, the **Encapsulating Security Payload** (**ESP**) protocol provides full encryption of the payload, while the **Authentication Header** (**AH**) adds authentication headers that guarantee the integrity of the payload but does not encrypt or otherwise obfuscate the payload by itself.

To add a new IPSec Crypto profile, *pick the strongest options available on both peers*. If possible, use **Suite-B-GCM-128** for small remote devices and **Suite-B-GCM-256** for larger peers, or create a new profile with the following steps:

1. Click on **Add** and name the profile so that you can easily identify it.
2. Set the IPSec protocol to **ESP** or **AH**.
3. **Set Encryption:**
    - `3des` (available for legacy compatibility; do not use unless necessary)
    - `aes-128-cbc`
    - `aes-192-cbc`
    - `aes-256-cbc`
    - `aes-128-ccm`
    - `aes-128-gcm`
    - `aes-256-gcm`
    - `Null`
4. Set **Authentication**:
    - `md5` (available for legacy compatibility; do not use unless necessary)
    - `sha1` (available for legacy compatibility; do not use unless necessary)
    - `sha256`
    - `sha384`
    - `sha512`
    - `none`
5. Set **DH group**:
    - **DH Group 1:** `768-bit group`
    - **DH Group 2:** `1024-bit group`
    - **DH Group 5:** `1536-bit group`

- **DH Group 14:** `2048-bit group`
- **DH Group 15:** `3072-bit group`
- **DH Group 16:** `4096-bit group`
- **DH Group 19:** `256-bit elliptic curve group`
- **DH Group 20:** `384-bit elliptic curve group`
- **DH Group 21:** `521-bit elliptic curve group`

Alternatively, select one of the PQ KEMs, as seen in *step 8*, for true PQ KEMs IKEv2 deployment.

> For maximum compatibility, it is best practice to use a traditional DH group here and only use the PQ KEM in the additional rounds in *step 8*.

Another alternative is to select **no-pfs** (no **Perfect Forward Secrecy**). PFS ensures keys are not reused and if a key is compromised somehow, past and future keys are not exposed. While it is recommended to always use a DH group, PFS may be disabled for performance reasons (the same base key is reused for all rekeys, decreasing calculations). Selecting **no-pfs** disables the use of Diffie-Hellman groups.

6. Set **Lifetime** in hours (1 is the industry default).
7. Optionally, enable **Lifesize**, which triggers a rekey if a certain amount of data has been transmitted.
8. In the **Advanced Options** tab (available in PAN-OS 11 and higher) we can set IKEv2 PQ KEMs in up to 7 additional rounds. These additional rounds allow even more resilience against decryption in case one of the PQ KEMs does become vulnerable, by adding more mechanisms. Select one of the following mechanisms for one or more of the additional rounds (ensure the peer supports additional rounds):

   - **Kyber-512**
   - **Kyber-768**
   - **Kyber-1024**
   - **Bike-L1**
   - **Bike-L2**
   - **Bike-L3**
   - **Frodokem-640-aes**
   - **Frodokem-640-shake**
   - **Frodokem-976-aes**
   - **Frodokem-976-shake**
   - **Frodokem-1344-aes**
   - **Frodokem-1344-shake**
   - **Hqc-128**
   - **Hqc-192**

- **Hqc-256**
- **Classic-mcelliece-348863**
- **Classic-mcelliece-348863f**
- **Ntruprime-sntrup761**

9. Click **OK**.

> Do not use `des`, `3des`, `md5`, or `sha1` unless you need to connect to a legacy system that does not support stronger algorithms.

Setting up the IKE Gateway

The next thing we need to set up is the IKE Gateway, which can be found in **Network** | **Network Profiles** | **IKE Gateways**. The IKE Gateway represents the settings needed during phase 1. IKE phase 1 is the authentication phase where the peers verify each other's authenticity before moving on to creating a secure tunnel in phase 2.

Follow these steps to create the IKE Gateway:

1. Click on **Add** and set a descriptive name for the peer you will be connecting to.
2. Set **Version** to **IKEv2 only mode**, or **IKEv2 preferred mode** if you're not sure whether the remote end supports IKEv2.

   If the remote end only supports IKEv1, leave the default of **IKEv1 only mode**, which will skip attempting to negotiate IKEv2.
3. Choose whether you'll set up a tunnel between **IPv4** or **IPv6 nodes.**
4. Select the physical interface that will be maintaining the connection to the remote end (this could be a loopback interface as well).
5. Set **Local IP Address**. (If the physical interface is set as DHCP client, leave **Local IP Address** as **None**.)
6. Select whether the peer has a static IP or a resolvable FQDN.

   If the remote peer has a dynamic IP address assigned by the ISP or uses a private IP on the public interface, select **Dynamic** as the **Peer IP Address Type**.
7. Set **Peer Address** by adding an IP or FQDN (if the peer is dynamic, this field disappears).
8. Select **Pre-Shared Key** for **Authentication** (see *steps 12–18* for the process to set a **certificate** for authentication).
9. Type in and confirm the **Pre-Shared Key** (**PSK**).

10. Optionally, you can use a local and peer identification parameter. If unused, both peers will identify themselves by their physical IP address during the negotiation. Some use cases where identification is required are:

    - When one or both peers are behind a NAT device, the default identifier (physical IP address of the host) will not match the source IP address as that has been changed by the NAT device.
    - Multiple tunnels are being set up between the same IP addresses. The identifiers will help identify each tunnel from the other.

    Identifier options include FQDN, an IP address (this option can be used to match the upstream NAT IP), a key ID (binary format ID string in HEX), or a user FQDN (an email address).

    The IKE Gateway should look similar to the following screenshot:

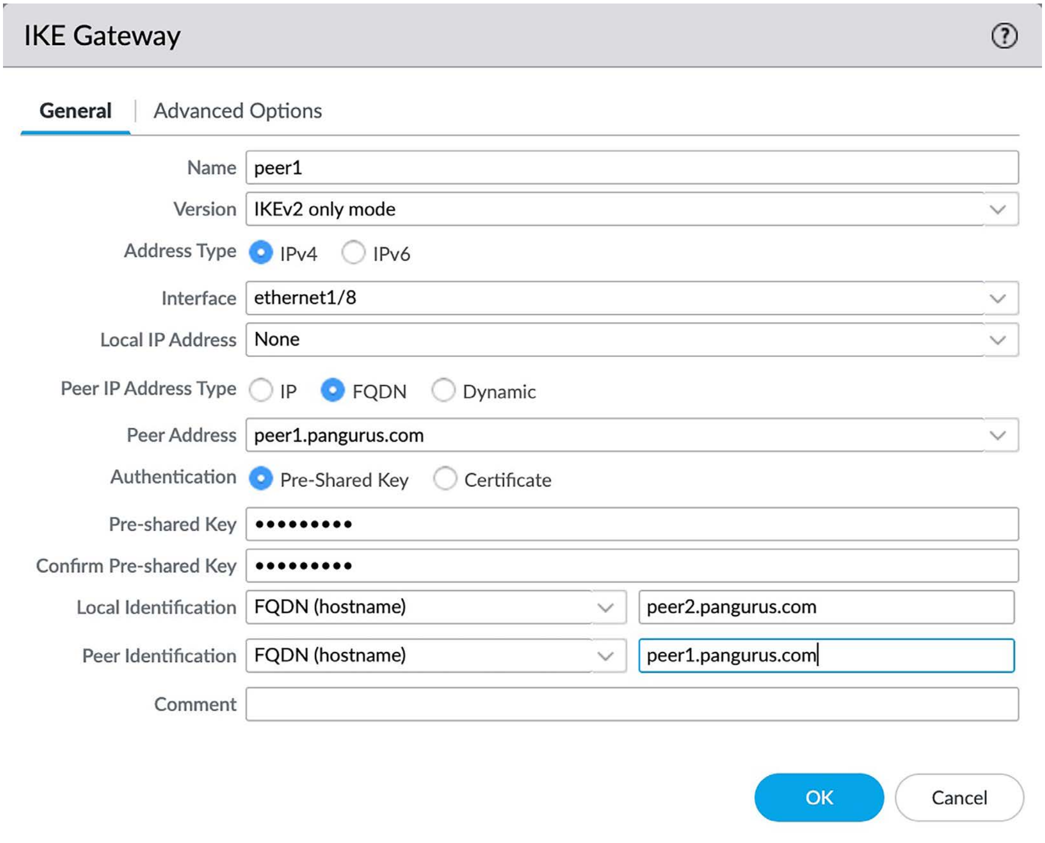


Figure 11.4: IKE Gateway

11. For Certificate-based authentication (instead of Pre-shared Key), continue to *step 12*. Otherwise, go to **Advanced Options** and skip to *step 19*.

    If you selected **Certificate** as the **Authentication** method, the last few steps are a little different.

12. Select **Local Certificate**. If it hasn't been uploaded or generated yet, you can do so from the dropdown.
13. You can optionally set **HTTP Certificate Exchange** to use the hash-and-URL exchange method to let the peer know where to fetch the certificate from.
14. Select **Distinguished Name**, **FQDN**, **IP**, or **User FQDN** for **Local and Peer Identification** and set a matching value for the **Local and remote peer**.
15. For **Peer ID Check**, set **Exact** if **Peer Identification** must exactly match the peer certificate and **Wildcard** if the identification is a subdomain or the certificate is a wildcard certificate.
16. Optionally, if the data used in the identification does not match that of the certificate, select **Permit peer identification and certificate payload identification mismatch**.
17. Add or create the certificate profile that supports the local certificate.
18. Go to **Advanced Options.**

The **Local** and **Peer** identification can be used as customized identification (rather than the IP address) or can be used to match the physical IP address if either peer is behind a NAT device. In the case where certificates are used, both values are matched against the certificate **Common Names** (**CNs**) and could cause issues if there are mismatches.

In the **Advanced Options** tab, follow these steps:

1. Set **Enable Passive Mode** if the local device should only receive inbound connections and not attempt to connect to the remote peer. This can help preserve the bandwidth or prevent unsuccessful connection attempts if the remote peer goes offline regularly or has a dynamic IP that is prone to change.
2. Set **Enable NAT Traversal** if either side is behind a NAT device.
3. If you selected **IKEv1 only mode** or **IKEv2 preferred mode**, there will still be an **IKEv1** tab available.

   In **IKEv1**, if this tab is available, set the following:

   - In **Exchange Mode**, select **main** if both sides use a static IP or **aggressive** if at least one side has a dynamic IP. You can use auto if you are not sure.
   - Set the **IKE Crypto Profile** to the desired IKE Crypto profile.
   - If fragmentation is expected, tick the **Enable Fragmentation** box.
   - Review the parameters for **Dead Peer Detection.** These are used to verify if the remote peer is still responsive before disconnecting the tunnel. Both peers will have the same DPD settings.

4. If you selected **IKEv2 only mode** or **IKEv2 preferred mode**, there will be an **IKEv2** tab. In the **IKEv2 | General** tab, set the following:
    - Set the **IKE Crypto Profile** to the desired IKE Crypto profile.
    - Enable **Strict Cookie Validation** to enforce IKEv2 cookie validation. The initiator must always send an **IKE_SA_INIT** containing a cookie.
    - If fragmentation is expected, enable **IKEv2 Fragmentation**, which will alter the MTU of IKEv2 packets. Review and update as needed.
    - **Liveness Check** will send an empty informational packet if no IKEv2 packet has been received (idle) for the amount of time specified and will function as a keepalive. After 10 liveness packets have been sent with no reply, the tunnel is broken down and needs to be reinitiated.
5. In the **IKEv2 | PG PPK** tab, you can enable **Post-Quantum Pre-shared Key**, which resists attacks by quantum computers. This option is only available in IKEv2. There are two negotiation modes available:
    - **Preferred**: If the peer supports RFC8784 PPKs, PPKs are used. If the peer does not support RFC8784, the IKEv2 handshake falls back to the classic key exchange, Diffie-Hellman.
    - **Mandatory**: The responding peer must support RFC8784 PPKs. If the peer does not support PPKs, the connection is aborted.

    A PPK Key ID needs to be created if PG PPK is enabled:

    - **PPK KeyID** is a name that identifies the associated PPK.
    - **PPK Secret** and **Confirm PPK Secret** are the shared secret. One can be autogenerated by clicking the **Generate Strong PPK** link.
    - **PPK length** sets the required length of the PPK secret.
6. In the **IKEv2 | PQ KEM** tab, you can enable Post Quantum Key Exchange for this IKE gateway (as configured in the **IKE Crypto profile** | **Advanced Options** | **Additional Rounds**).

    Block IKEv2 if vulnerable cipher is used can be enabled to automatically start blocking IKEv2 if a cipher is compromised in the future.

7. Click **OK**. The **Advanced Options** settings should now look similar to the following screenshot:

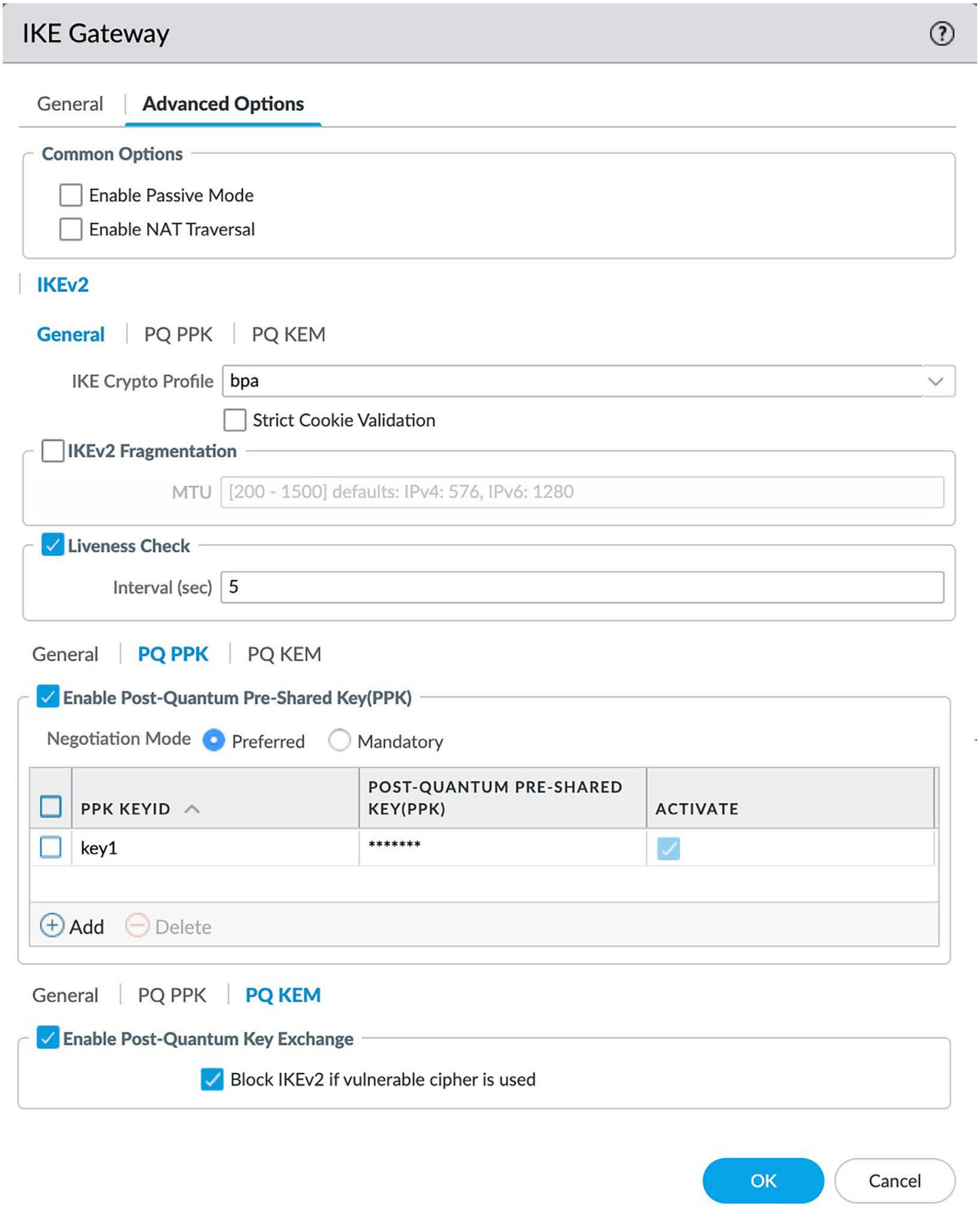


Figure 11.5: The IKE Gateway advanced settings

Setting up the tunnel interface

Before we set up the actual tunnel, make sure you have a tunnel interface available in **Network | Interfaces | Tunnel**. If no free one is available, create a new one and make sure to set it to a unique zone, such as **VPN**, and add it to your virtual router.

If you need to set up tunnel monitoring, if dynamic routing (OSPF or BGP) is negotiated across the tunnel, or if the remote end requires **numbered tunnel interfaces**, you can add an IP address, but this is not required if the tunnel is set up between two Palo Alto Networks devices. If you are going to use tunnel monitoring, also enable a management profile that allows **ping**, as follows:

INTERFACE	MANAGEMENT PROFILE	IP ADDRESS	VURTUAL ROUTER	SECURITY ZONE	FEATURES
tunnel		none	default	vpn	
tunnel.3	ping	172.31.0.1/30	default	vpn	

Figure 11.6: Tunnel interface

As shown in the following diagram, phase 1 is established between the physical (or loopback) interfaces of both peers and serves to carry IPSec phase 2. IPSec is established between the tunnel interfaces on both ends. Tunnel interfaces are virtual interfaces and should be treated as if there is a physical interface with a network connected to it, as well as being configured with its own zone.

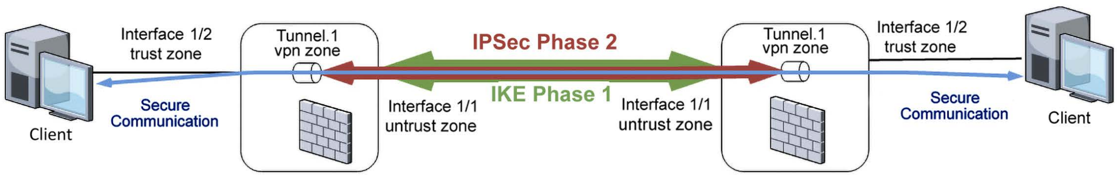


Figure 11.7: A VPN tunnel from the firewall's perspective

In the security policy, note the following points:

- Connections from the client will be established between the **trust zone** and the **VPN zone.**
- The IPSec tunnel will require a security rule that is established from the **Untrust zone** to the **Untrust zone** (from the external interface out to the internet).

Creating the IPSec tunnel

To create the IPSec tunnel, go to **Network | IPSec Tunnels** and follow these steps:

1. Click on **Add** and provide a descriptive name.
2. Set the appropriate tunnel interface.
3. For **Type**, you can have the firewall create the SPI automatically with **Auto Key**, set it manually with **Manual Key**, or set the tunnel for **GlobalProtect Satellite**.

4. If you select **Auto Key**, set the following:

    - Set **IPv4** or **IPv6** for **Address Type**.
    - Select the appropriate IKE Gateway (which we created in the previous section).
    - Select the IPSec Crypto profile.
    - Select **Show Advanced Options**.
    - Select **Enable Replay Protection**.
    - ToS headers can be carried from the inner IP header to the outer IP header by enabling **Copy ToS Header**. You can transport a GRE tunnel inside the IPSec tunnel by selecting **Add GRE Encapsulation**, which will add a GRE header after the IPSec header.
    - In **IPSec Mode,** select **Tunnel** to use a full IPSec tunnel or **Transport** to enable authentication headers, which does not encrypt the payload, but adds header information to confirm if the payload has been tampered with.
    - Enable **Add GRE Encapsulation** to add additional GRE encapsulation to the payload.
    - Enable **Tunnel Monitor** and set the remote tunnel interface IP for **Destination IP**. Add a monitoring profile to set an action if the tunnel fails:

        i. **wait-recover** will keep the tunnel interface up and will keep routing packets to it until the tunnel is restored.

        ii. **fail-over** will bring the interface down and have routing take care of packets via an alternative route.

    Use **fail-over** if you set up a second tunnel; otherwise, use **wait-recover**. Your tunnel configuration will look as in the following screenshot:

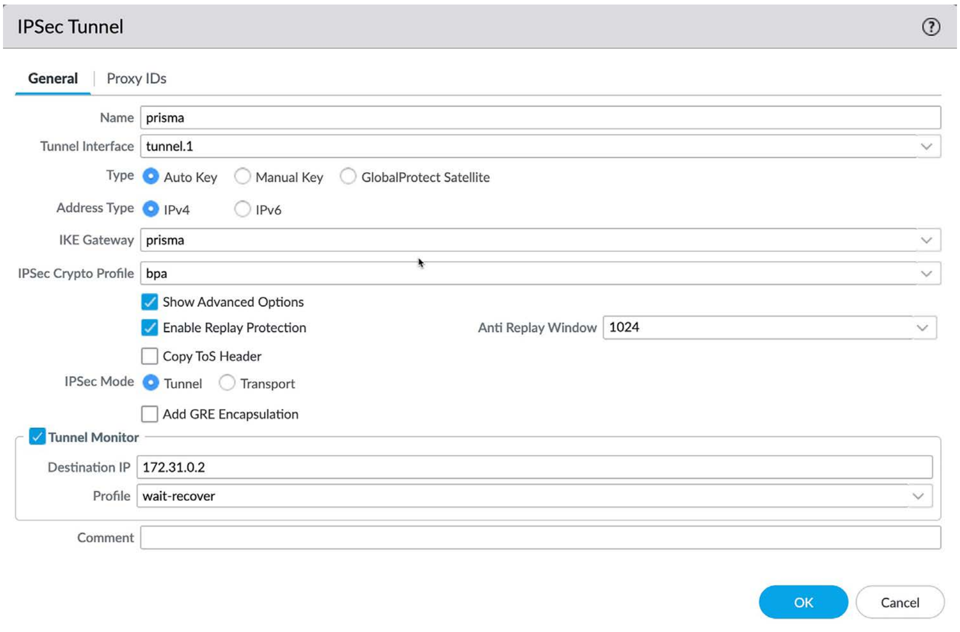


Figure 11.8: IPSec tunnel configuration

- In the **Proxy IDs** tab, local to remote IP subnet pairs can be added to restrict the tunnel to just allow communication between these subnets. It influences the security associations that are created and is required if the remote peer is a policy-based device. The default setting (no Proxy IDs) will generate a single security association pair for 0.0.0.0/0 to 0.0.0.0/0.
- **Proxy-ID Strict Matching Mode** can be enabled to force both sides to use exactly the same subnets for Proxy-ID negotiation.

5. In **Manual Key**, you get to set all the phase 1 and phase 2 parameters for a single IPSec tunnel. This works well with a route-based peer but could become troublesome with a policy-based peer, as multiple manual IPSec tunnels will need to be created.
6. In **GlobalProtect Satellite**, set the following:
    - Set (IP) **Portal Address**.
    - Select the external interface.
    - Set the local IPv4 or IPv6 address.
    - Open the **Advanced Options** tab.
    - Either select **Publish all static and connected routes to Gateway** to share the entire routing table to the GlobalProtect gateway or manually configure the subnets to publish to the gateway.
    - If you have an external device certificate for the firewall, select **External Certificate Authority** and set the certificate and matching certificate profile to authenticate against the gateway.
7. Click **OK**.

It is worth noting that a policy-based firewall will create an IPSec tunnel based on subnet pairs as defined in a policy (**subnet-A-local** gets access to **subnet-X-remote**), whereas a routing-based firewall will simply create a tunnel and then route packets into it. The Palo Alto Networks firewall is route-based, so it will default to using a single tunnel for all communications. Proxy IDs force splitting the single configuration into multiple IPSec tunnels.

Pro tip

While having a single tunnel simplifies configuration, it may suffer from performance degradation due to how sessions are handled on the data plane and a single tunnel will be processed by a single CPU. Creating multiple tunnels through Proxy IDs will spread the load over more cores.

The drawback is that each Proxy ID pair tunnel counts toward the system's maximum number of IPSec tunnels. For example, the PA-410 supports up to 1,000 tunnels. If each IPSec tunnel has 10 Proxy ID pairs, only 100 tunnels are supported.

The last step is to add routes that forward any packets destined for the remote subnet into the tunnel. In **Network** | **Virtual Routers**, open the virtual router that holds the tunnel interface. In **Static Routes**, add a new route:

1. Give it a descriptive name.
2. Set the **Destination** subnet.
3. Select the tunnel interface for **Interface**.
4. **Next Hop** can either be **None** to simply route packets into the tunnel, or the remote tunnel IP, which some systems may require.
5. Change **Admin Distance** and **Metric** if needed.
6. Click **OK** and commit your changes. The route should look similar to the following screenshot:

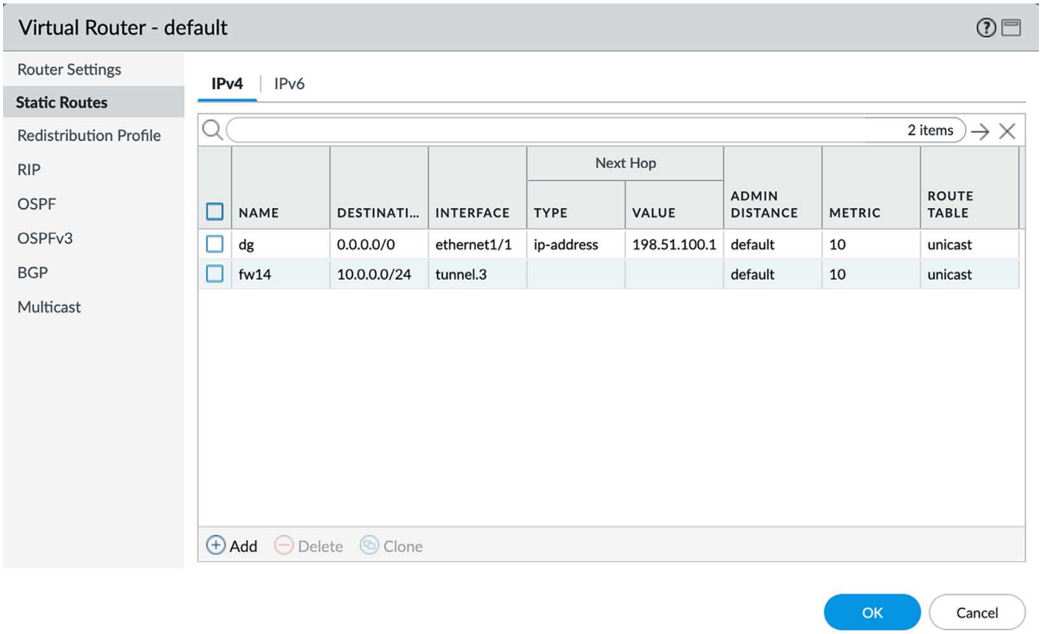


Figure 11.9: A static route into a tunnel

To test connectivity and manually initiate the connection, you can use the following commands to initiate phase 1 and phase 2, respectively:

```
> test vpn ike-sa gateway <IKEgateway>
> test vpn ipsec-sa tunnel <tunnel name>
```

The IKE SA first needs to be negotiated successfully before the IPSec SA can be tested. You can follow the connection attempts through the system log in **Monitor** | **Logs** | **System** while using the **(subtype eq vpn)** filter. If Proxy IDs were configured, multiple tunnels will exist, representing each subnet pair. Use the *tab* key in the above `ipsec-sa` command to get a list of the available tunnels.

You can follow the actual process logs via the CLI to see how the tunnel is being negotiated and set up any errors or interesting information if the tunnel doesn't come up:

```
> tail follow yes mp-log ikemgr.log
> tail follow yes mp-log cryptod.log
```

Here is a checklist of the things you need to agree on with the remote peer:

- For phase 1, which encryption authentication, Diffie-Hellman group and key lifetime will be used. Will PQ KEM be available?
- For phase 2, whether you will set up ESP or AH. If you choose ESP, which encryption algorithm will be used, and if you choose AH, which authentication will be used, which Diffie-Hellman group, and how long should the lifetime be?
- Does the remote peer support IKEv2? Can PQ KEM be used?
- What is the remote peer IP or FQDN, or is the host on a dynamic IP?
- Will you use a PSK or a certificate to establish phase 1 authentication?
- Is either host behind a NAT device?
- Does the remote end support replay protection?

Now that you have a firm understanding of how to set up a site-to-site VPN, we will move on to configuring GlobalProtect for a client VPN.

Configuring GlobalProtect

Using a site-to-site VPN is a very robust and secure method of connecting two systems; however, it is less appropriate and much harder to configure for endpoints such as laptops or mobile phones. To accommodate many different OSes and easier configuration options, GlobalProtect is available to provide connectivity to employees, contractors, and guests.

GlobalProtect is an SSL-capable VPN client that also supports IPSec, which means that the VPN connection can tunnel over HTTPS, so the client will likely be able to connect from most locations where traditional IPSec may be blocked by a firewall or other filtering device. IPSec can be enabled and set as the preferred connection method with a fallback to SSL if IPSec is blocked.

Most of the GlobalProtect functionality does not require an additional license, but there are a few features that do require a license:

- Performing **Host Information Profile** (**HIP**) checks and applying security rules based on the HIP status
- Supporting GlobalProtect on mobile endpoints (such as Android, iOS, Chrome OS and Linux), or IoT devices
- IPv6 support for external gateways
- Splitting tunnels based on destination domains, a client process, or a streaming application
- Splitting DNS

- Providing a clientless VPN alternative
- Identifying the managed devices by their serial number
- Quarantining compromised devices
- GlobalProtect App log collection for troubleshooting: this is an on-premises extension of **Autonomous Digital Experience Management** (**ADEM**) used in Prisma Access to send GP data up to SLS when a user is experiencing issues with their connection.

There are two main components that need to be configured when setting up GlobalProtect:

- A **portal** that serves configuration updates to all connected clients, provides clients with a download page to get the client package installer files for Windows and Mac, and provides a clientless VPN portal
- A **gateway**, which is where the agent connects to establish a secure connection

A typical deployment has one portal and as many gateways as needed. Gateways can be spread over strategic locations, so users always have an optimal connection to the corporate network. Gateways can be deployed on physical or virtual appliances on-premises or in the cloud (such as with Azure, AWS, GCS, and so on) or as part of Prisma Access. An internal gateway can also be set up to function as a User ID and HIP enforcement point for internal users to be able to access sensitive resources on the network. A single portal is needed to distribute the agent configuration and provide the available gateways.

Setting up the portal

To create a new **Portal** object, go to **Network** | **GlobalProtect** | **Portals** and follow these steps:

1. Click on **Add**.
2. In the **General** tab, set a name for the portal.
3. Select the interface that the portal will be listening on:
    - If you have an IP to spare, I would recommend creating a loopback interface in the external/untrust zone.
    - Use the **Untrust** interface to make the portal available on the internet.
    - Use an **internal** interface to only provide portal services to internal or connected hosts (the latter means you will not be able to change critical information easily as users need to be logged in first before being able to get config updates when they are connected remotely).

> Setting the portal on a loopback interface makes any packets carrying an exploit targeting the portal IP go through full threat prevention before actually hitting the interface. This should be considered best practice.

4. Select **IPv4**, **IPv6**, or **IPv4 and IPv6**.

5. In the **Appearance** dropdowns, you can choose to use the default page, upload a custom page, or disable the landing page entirely (when disabled, the agent will be able to fetch the configuration, but no page is displayed if someone connects using a browser).
6. In **Log settings**, keep **Log Unsuccessful SSL Handshake** enabled.

   Then move on to the **Authentication** tab.

7. You need to provide an SSL/TLS service profile that serves the certificate that will be used for the portal. Ensure the certificate matches the FQDN that is used for the portal (for example, `portal.example.com`) in the Common Name and SAN, and has been imported in **Device | Certificate Management | Certificates**, then create a new service profile in **Device | Certificate Management | SSL/TLS Service Profile.** Set **Min. version** to **TLSv1.2** and disable weak ciphers.
8. Create a new client authentication:
   - Set a descriptive name.
   - You can select which client OS this authentication method will apply to. Set **Any** for everyone or set a specific OS if different OSes should log in using different profiles (for example, LDAP authentication for Windows machines and RADIUS for Linux clients).
   - Choose the authentication profile that will be used to authenticate users. You can create a new one from the dropdown.
   - By default, users need to provide a username and password. Optionally, if client certificates are enabled (see the next step), provide a *client certificate*. You can set **Allow Authentication with User Credentials OR Client Certificate** to **Yes** so that users can log in with either their username/password *or* a client certificate. Setting **Allow Authentication with User Credentials OR Client Certificate** to **No** will require both a client certificate and username/password for a user to gain access, if a certificate is configured in the next step.
   - Click **OK.**
9. If you want clients to use a client certificate when connecting, create a certificate profile in **Device | Certificate Management | Certificate Profile**:
   - Set a profile name.
   - Set the **Username** field to **Subject** (Common Name) or **Subject Alt** (and select **Email** or **Principal Name**). Leave this as **None** if a generic (machine) certificate will be used, rather than a personalized one.
   - Set **NetBIOS Domain** in **Domain.**
   - Add the CA certificate that will sign the user certificates and add appropriate OCSP (**Online Certificate Status Protocol**) URLs.
   - Click **OK.**

In the **Portal Data Collection** tab, you can have GlobalProtect collect the Windows registry key or Mac Plist entries. These values can be used to select which configuration is sent to the client. Collection can be configured as follows:

1. Set the certificate profile that will be used to match the machine certificate used by the GlobalProtect agent.
2. Add the registry/Plist keys that need to be registered.
3. In the **Agent** tab, you can control the configuration sent to the agent so that it can establish a connection.

   In the **Trusted Root CA** box, you can add CA and intermediary certificates if the portal and gateway certificates are self-signed so that the client trusts the certificates. An **SSL Decryption** certificate can also be installed in the client's trusted root certificates by checking **Install** in **Local Root Certificate Store**.

   Agent User Override Key is the **master key** used in the ticketing process to allow users to disable an always-on GlobalProtect agent on their system. If left unchanged, the system will use the system's default key to sign tickets. You can choose to change this key for security reasons (administrators will not need to know the key).

Multiple agent configurations can be created for different user types or client machines. The agent configs are processed **top to bottom** when a user connects, so make sure the more specific profiles are placed at the top. Create a new profile as follows:

1. Click on **Add**.
2. In the **Authentication** tab:

   - Set a descriptive name.
   - The client certificate can be used to push a certificate and its private key to the client. This certificate can be used to authenticate against the gateways.
   - The user credentials are saved by default in the GlobalProtect agent. You can set the **Set Save User Credentials** option to **Yes** or **No**, but also to **Save Username Only** so that only the username can be saved, or **Only With User Fingerprint** so that the credentials can be saved if the user uses biometric authentication.
   - Select **Generate cookie for authentication override** so that a (unique) cookie is generated and sent to the client by the portal after the user first logs in to the portal. This cookie can be used in lieu of "regular" authentication to improve user experience: a cookie remains valid for a certain amount of time and during this time the user does not need to reauthenticate using the normal authentication if they get disconnected or move to a different network.
   - Select **Accept cookie for authentication override** if the cookie will be used to authenticate, rather than the user credentials. Set an amount of time for which the cookie will be valid (maximum **365** days). Once the cookie expires, the user needs to provide credentials when logging in to the portal and will receive a fresh cookie.
   - If cookies are used, set the certificate that will be used to encrypt them.

- You can select which components (**Portal**, **internal-gateway**, the external gateways manual, or **Autodiscover**) will require **two-factor authentication** (**2FA**). This will help force MFA on certain components while allowing credentials to be saved for others.

3. In the **Config Selection Criteria** tab, you configure which user/user group or type of endpoint device this configuration will apply to:
   - In the **User/User Group** tab, a user or LDAP group and a client OS can be selected.
   - In the **Device Checks** tab, you can set an action to check whether a machine account with a device serial exists or a machine certificate has been installed.
   - In the **Custom Checks** tab, you can look for the registry key or Plist entries we set in the **Portal Data Collection** tab.

   If this section is left empty, the configuration will apply to everyone that is able to authenticate. If a component (for example, a user group) is configured and a user is not a member of the configured group, the next agent configuration will be checked until a match is found.

4. In the **Internal** tab, we can set **Internal Host Detection** for IPv4 and IPv6 and **Internal Gateways** for HIP checking:
   - Enable **Internal Host Detection IPv4** and set the IPv4 address and FQDN hostname of an internal resource to prevent internal hosts from setting up a VPN tunnel to the external gateways while inside the network (this can be any internal server or host or an internal gateway configured on the firewall). The client will perform a reverse lookup of the IP address against the internal DNS server for the IP address. If the returned record matches the FQDN in internal host detection, the agent will remain disconnected. This requires an entry on the internal DNS server's **in-addr-arpa** record.
   - Set **IPv6** and **IPv6 enabled FQDN** if IPv6 is used in the network.
   - Add Internal Gateways by their IP or FQDN (this value will need to match the certificate used on the internal gateway). **Source addresses** can be added to control which subnets will connect to a specific internal gateway. The internal gateways will be used for GlobalProtect to report User-ID information and provide HIP reports.

5. In the **External** tab, IP addresses or FQDN names for all available external gateways can be added, as well as third-party VPN clients. Adding these clients instructs the GlobalProtect agent to ignore routing added by these other VPN clients to prevent conflicts.

   When multiple gateways exist, the GlobalProtect agent will poll (through a TLS handshake) all of them to see which ones provide the optimal connection speed. Highest priority will always be preferred unless the average of all measured gateways' TLS responsiveness is lower than the highest priority gateway.

   The **cut-off time** is the time allowed for a gateway to reply. Create a gateway using the following steps. Repeat these for each gateway that needs to be made available to the agent:
   - Add a gateway and give it a descriptive name. This name will be visible to the user, so it should help them understand where their connection is being made to.
   - Add the FQDN or IP the connection will be made to. This should match the certificate that will be used on the **Gateway** object.

- Add a source region. This can be **Any** or any country, subnet, or global region. Setting a more specific value will limit connectivity to this gateway to the users that are in the country or region.
- Set a priority: from lowest to highest or manual user selection only. The priority has an inferior value compared to the result of the responsiveness poll – the **highest**, **high**, and **medium** priority items will be polled and connected to the gateway providing the fastest TLS response. If none of these priorities are available, the agent will move to the lower-priority gateways. The **Manual Only** priority excludes the gateway from the TLS connectivity test and will only be used if the user manually chooses to connect to it.
- Check the **Manual** box if the user is permitted to select this gateway as a preferred connection, but the gateway should still be included in the automated selection process. If the user selects the gateway as preferred, GlobalProtect will always connect to this gateway unless it is not available.
- Click **OK**.

Gateway connection priority is determined by TLS responsiveness; there are two slightly different scenarios in determining which gateway a GlobalProtect agent connects to:

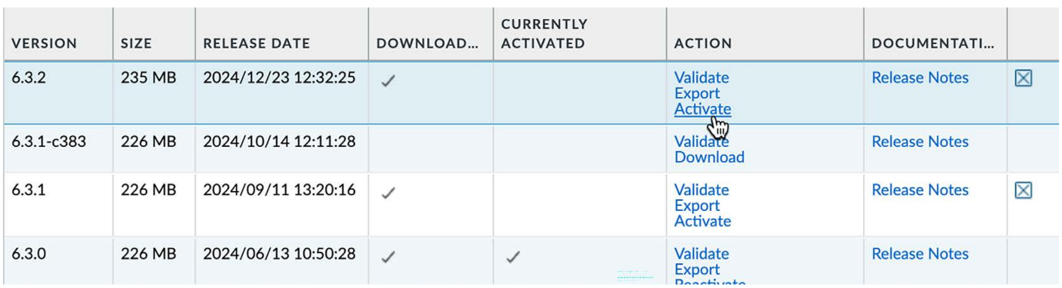
- In Prisma Access deployments, the client is served a smaller list of gateways, restricted to the in-country gateway and the regional backup. The in-country gateway has a priority of 1, and the regional backup a priority of 5. Both gateways will be tested for responsiveness by measuring TLS responsiveness (in milliseconds). The priority will be used as a multiplier so the lowest priority gateway has a higher calculated cost, meaning the in-country gateway is usually preferred unless it is responding much slower than the penalized cost for the regional backup.
- In a deployment with multiple gateways, connection priority will be calculated based on the gateway's TLS responsiveness (in milliseconds) and its priority (highest, high, medium… let's call them 1, 2, and 3, respectively). All gateways will be measured and highest (1) priority gateways will always be preferred over lower priority (2 and 3) gateways unless the highest (1) gateway's responsiveness is higher than the average responsiveness across all gateways. For example, if the highest (1) priority gateway measures 80 ms and the average of all other gateways is 79 ms, only then will a lower priority (high, 2) be considered.

In the **App** tab, we can configure how the GlobalProtect agent will behave. On the right-hand side, you can do the following:

- Enable a Welcome Page that pops up every time a user connects.
- If users are normally *not* allowed to disable GlobalProtect but an exceptional event could require some users to disable their agent, a passcode can be set here to share with users or IT staff.
- A password can be set for users to be able to uninstall GlobalProtect.
- A **Mobile Device Manager** (**MDM**) can be set to enroll mobile devices connecting through GlobalProtect.

On the left-hand side, you can configure how the agent behaves. I'll highlight the options that may need to be changed from the default or that are of interest:

1. **Connect Mode** is set to **user-logon** (**always on**) by default and ensures the GP agent establishes a VPN connection as soon as the user logs on to their machine. This can be changed to:
   - **On-demand**, which lets the user decide when to connect.
   - **Pre-logon** (**always on**), which establishes a connection using the machine certificate before the user logs on to their desktop environment. This facilitates logon scripts and allows IT staff to connect to a logged-out machine, for example.
   - **Pre-logon then On-demand**, which sets up a VPN connection when the laptop is booted up but not logged on and lets the user choose when to connect once they are logged in to the desktop.
2. **GlobalProtect App Config Refresh Interval** is the frequency at which the agent should refresh its configuration. By default, the agent will check for a fresh configuration every 24 hours. If you plan to introduce some changes, it may be beneficial to temporarily decrease the interval so changes can be rolled back more quickly.
3. **Allow User to Disconnect GlobalProtect** is set to **Allow**. Change this to **Allow with Comment**, **Allow with Passcode**, or **Allow with Ticket** (or **Disallow** altogether). **Allow with Ticket** requires users to call in and get a challenge response from an admin that can run **Generate Ticket** in **Network | GlobalProtect | Portals.**
4. **Display the following reasons to disconnect GlobalProtect** provides the user with an option prompt each time they want to disconnect GlobalProtect. This allows you to track why users are disconnecting so you can adjust some configuration accordingly (or train users on VPN hygiene).
5. With **Allow User to Uninstall GlobalProtect App**, Windows users can be prevented from uninstalling GlobalProtect or required to enter a password before being able to uninstall.
6. **Allow User to Upgrade GlobalProtect App** will prompt the user by default if an upgrade is made available on the portal. This is achieved by downloading and activating a new agent via **Device** | **GlobalProtect Client**, as illustrated in the screenshot below. This can be set to **Disallow**, **Allow Manually**, **Allow Transparently**, or **Internal**. Both **Allow Transparently** and **Internal** will update the agent automatically, but **Internal** will only perform the upgrade when the user is on the corporate network.



VERSION	SIZE	RELEASE DATE	DOWNLOAD...	CURRENTLY ACTIVATED	ACTION	DOCUMENTATI...	
6.3.2	235 MB	2024/12/23 12:32:25	✓		Validate Export Activate	Release Notes	☒
6.3.1-c383	226 MB	2024/10/14 12:11:28			Validate Download	Release Notes	
6.3.1	226 MB	2024/09/11 13:20:16	✓		Validate Export Activate	Release Notes	☒
6.3.0	226 MB	2024/06/13 10:50:28	✓	✓	Validate Export	Release Notes	

Figure 11.10: Activating a new GlobalProtect package

7. Set **Allow users to Sign Out from GlobalProtect** to **No** if users are not permitted to sign out and thereby disable GlobalProtect.
8. **Allow user to extend GlobalProtect User Session** allows a user to extend the current session (default is **No**). By default, a user session can last up to 30 days before being forcibly disconnected, requiring reauthentication. This setting allows a user to refresh this timeout without reauthenticating
9. **Use Single Sign-on** (separate option for Windows or macOS) is set on by default and will reuse the user login credentials to establish a connection.
10. **Use Single Sign-on for smart card PIN** (Windows only) allows the reuse of smartcard credentials used to log in to Windows without asking the user for a PIN.
11. **Clear Single Sign-on Credentials on Logout** (Windows only) deletes the stored credentials from GlobalProtect if the user logs off from their desktop.
12. **Use Default Authentication on Kerberos Authentication failure** can be set to fall back to default authentication if Kerberos authentication fails.
13. **Use Default Browser for SAML authentication** if set to **Yes,** will invoke the system's default browser (Edge, Chrome, Firefox, etc.) to present the user with the SAML authentication screen instead of the native GlobalProtect browser. This may be required for certain types of authentication (like WebAuthn for YubiKey) or to meet authentication conditions.
14. **Automatic restoration of VPN Connection Timeout** specifies what GlobalProtect should do and how long it should wait in the event the tunnel gets disconnected. A value of 0 means GlobalProtect should not try to re-establish the tunnel. A value between 1 and 180 indicates how long GlobalProtect should keep trying to re-establish the tunnel.

> With **Always-On** enabled, when a user switches from an external to an internal network, GlobalProtect will try to re-establish the VPN tunnel. The user must select **Rediscover Network** in GlobalProtect to switch to the right network setting.

15. **Wait Time Between VPN Connection Restore Attempts** specifies the number of seconds between each attempt to reconnect a disconnected tunnel.
16. **Endpoint Traffic Policy Enforcement** can be set to disable all network access on the physical network adapter if GlobalProtect is not connected to an internal or external gateway (no exceptions need to be set for the gateway IPs or FQDNs):
    - **No** disables enforcement
    - **TCP/UDP Traffic Based on Tunnel Address Type** enables enforcement for TCP and UDP only for the IP (IPv4 or IPv6) type used inside the tunnel
    - **All TCP/UDP Traffic** enables enforcement for TCP and UDP for both IPv4 and IPv6
    - **All Traffic** enables enforcement for all protocols, including ICMP

    If Endpoint Traffic Policy Enforcement is set:

    - Set **Captive Portal Exception Timeout** to allow users to authenticate to a captive portal if they are at a hotel or airport before network access is blocked.

- Edit **Traffic Blocking Notification Message** so that users are made aware when traffic is being blocked due to GlobalProtect not being connected.
- Specific IP addresses or FQDN hosts can be set to **bypass Enforce GlobalProtect for Network Access** so these hosts are still reachable even when traffic is being blocked (this can be useful to maintain connectivity to help pages or authentication portals).

17. **Enforce GlobalProtect Connection for Network Access** can be enabled to force all traffic to traverse the GlobalProtect tunnel. Enabling this setting will block all traffic to and from the host until the agent is connected to an internal or external gateway.

    If this option is set, also set **Captive Portal Exception Timeout** to allow users to authenticate to a captive portal if they are at a hotel or airport before network access is blocked.

18. **Allow traffic to specified hosts/networks when Enforce GlobalProtect Connection for Network Access is enabled and GlobalProtect Connection is not established** – up to 10 IP addresses or network segments can be set here to exclude enforcement when GlobalProtect is disconnected. Separate entries with a comma, and do not leave spaces. This setting can allow users to access local resources when the GlobalProtect tunnel is disconnected.
19. **Allow traffic to specified FQDN when Enforce GlobalProtect Connection for Network Access is enabled and GlobalProtect Connection is not established** – up to 40 FQDNs can be set here to exclude enforcement when GlobalProtect is disconnected. Separate entries with a comma, and do not leave spaces. This setting can be used to ensure access to the cloud-hosted **identity provider** (**IdP**) is available so the agent can authenticate.
20. **Captive Portal Exception Timeout** provides a grace period if enforcement is enabled for the user to establish a connection to a captive portal and authenticate before local access is blocked. A value of **0** means GlobalProtect will not allow users to connect to a captive portal.
21. **Automatically Launch Web Page in Default Browser Upon Captive Portal Detection** will automatically launch a browser and open the configured FQDN or IP address if a captive portal is detected on the network.
22. **Traffic Blocking Notification Delay** is the amount of time GlobalProtect waits after network connectivity is established to display a message that traffic blocking is enabled (due to the VPN tunnel not being established).
23. For **Display Traffic Blocking Notification Message**, select whether to display a message if traffic enforcement is enabled.
24. For **Allow User to Dismiss Traffic Blocking Notifications**, selecting **No** will force always displaying the traffic enforcement message.
25. For **Display Captive Portal Detection Message**, select whether to display a message when a captive portal is detected on the network.
26. For **Captive Portal Detection Message**, enter the message to display if the above option is set to **Yes**.
27. **Captive Portal Notification Delay** is the delay in seconds for GlobalProtect to wait before displaying the captive portal message.

28. **Client Certificate Store Lookup** determines where GlobalProtect can look for valid client certificates when prompted to present one:
    - **User:** Look only in the user local certificate store
    - **Machine:** Look only in the local machine certificate store
    - **User and Machine:** Look both in local user and machine certificate store
29. **SCEP Certificate Renewal Period** sets the number of days before a SCEP-generated certificate expires that the portal will request a new certificate from the internal PKI system.
30. **Extended Key Usage OID for Client Certificate** allows you to set the **object identifier** (**OID**) you want GlobalProtect to use to find the client certificate on a system that has multiple certificates installed. Some common OIDs include:
    - `1.3.6.1.5.5.7.3.1`: Server authentication
    - `1.3.6.1.5.5.7.3.3`: Code signing
    - `1.3.6.1.5.5.7.3.4`: Email protection
    - `1.3.6.1.5.5.7.3.5`: IPSec end system
    - `1.3.6.1.5.5.7.3.6`: IPSec tunnel
    - `1.3.6.1.5.5.7.3.7`: IPSec user
    - `1.3.6.1.5.5.7.3.8`: Time stamping
    - `1.3.6.1.5.5.7.3.9`: OCSP signing
31. **Retain Connection on Smart Card Removal** allows GlobalProtect to remain connected when a smartcard that was used to authenticate is removed.
32. **Enable Advanced View** can be set to **No** if users should have a simplified experience. They will not be able to access advanced features in GlobalProtect.
33. **Have User Accept Terms of Use before Creating Tunnel** presents the user with a terms of use message if set to **Yes**. You must configure the GlobalProtect **Welcome** page in the **General** tab of the portal configuration.
34. **Enable Rediscover Network** option can be set to **No** to prevent the user from manually rediscovering the network connection.
35. **Enable Resubmit Host Profile** option can be set to **No** so the user is unable to manually resubmit the agent's latest HIP.
36. **Allow user to change Portal Address** can be set to **No** if the user is not allowed the option to change the portal address, which could be used to enter portal information to a different organization. The drawback is that the portal address will need to be pushed (via **Global Policy Objects** (**GPOs**), for example) to the clients:

```
Windows: HKEY_LOCAL Global Policy Objects (GPO _MACHINE\SOFTWARE\PaloAlto
Networks\GlobalProtect\PanSetup with the Portal key.
Mac plist: /Library/Preferences/com.paloaltonetworks.GlobalProtect.
pansetup.plist with the Portal key
```

37. **Allow User to Continue with Invalid Portal Server Certificate** is set to **No** by default. It can be set to **Yes** to accommodate experimental portals if a public certificate with a matching Common Name is not available.
38. **Display GlobalProtect Icon** lets you disable the GlobalProtect icon in the system tray.
39. **User Switch Tunnel Rename Timeout** can be useful when IT persons need to connect to a client machine using **Windows Remote Desktop** (**RDP**). By default, the tunnel is renamed immediately, potentially causing different security rules based on group membership to be applied immediately. In some cases, it may be useful for the tunnel not to be renamed and the user's original access to remain active for a while after the new user has logged on.
40. **Pre-Logon Tunnel Rename Timeout** has three settings:
    - `-1` maintains the active pre-logon tunnel and applies the user's UserID to the connection.
    - `0` terminates the pre-logon connection and tries to re-establish a new connection with the user's credentials.
    - `1-600` indicates the number of seconds a pre-logon tunnel can remain active while the user logon to the desktop is processed. If the user logs on within the allowed timeframe, the tunnel is renamed, else the tunnel is terminated.
41. You can keep the tunnel connected for a specified amount of time by setting a timeout in **Preserve Tunnel on User Logoff.**
42. **Connect with SSL Only** forces the use of SSL for this agent configuration profile, even if IPSec is enabled on the gateway (leave this as **No**).
43. **Custom Password Expiration Message** displays a custom message (maximum 200 characters) when the user's password is about to expire.
44. **Automatically Use SSL When IPSec Is Unreliable** specifies the amount of time GlobalProtect will switch to SSL if IPSec connectivity fails. During this time, GlobalProtect will not attempt to re-establish IPSec.
45. **Display IPSec to SSL Fallback Notification** determines whether or not the user is informed that their tunnel has fallen back to SSL.
46. **Connect with SSL Only** will force the use of SSL even if IPSec is available on the gateway.
47. **GlobalProtect Connection MTU** sets the **MTU** (**Maximum Transmission Unit**) inside the GlobalProtect tunnel (the default is **1400**).
48. **Maximum Internal Gateway Connection Attempts** sets the maximum number of times GlobalProtect tries to connect to an internal gateway if the first attempt fails. A value of **0** will prevent GlobalProtect from trying after the first failure.
49. **Enable Advanced Internal Host Detection** enables an extra layer of security where GlobalProtect will validate the server certificate of the internal gateway and perform a reverse DNS lookup to validate the internal gateway FQDN.
50. **Portal Connection Timeout** sets how long GlobalProtect will wait before timing out a connection request to the portal (default is **30 seconds**).
51. **TCP Connection Timeout** and **TCP Receive Timeout** are the amount of time GlobalProtect will wait to timeout TCP sessions.

52. **Split-Tunnel Option** specifies if split tunnel only splits off network connections, or also splits off DNS queries for FQDN configured in split tunneling:
    - **Network traffic only** resolves all DNS on the GlobalProtect DNS server and only splits off the actual connections
    - **Both network traffic and DNS** performs local DNS lookup for split tunnel FQDN records and then splits the connections off locally
53. **Resolve All FQDNs Using DNS Servers Assigned by the Tunnel**, if set to **Yes**, will force all DNS queries to the GlobalProtect tunnel DNS server. **No** will allow local DNS lookups.
54. **Proxy Auto-Configuration (PAC) File URL** can be enabled to fetch a `proxy.pac` or `wpad.dat` file to autoconfigure the local proxy configuration.
55. **Detect Proxy for Each Connection**, if set to **No**, will use the proxy for all connections. **Yes** will auto detect the proxy for every connection.
56. **Set Up Tunnel Over Proxy** determines whether GlobalProtect must use or bypass proxy configuration.
57. **Send HIP Report Immediately if Windows Security Center (WSC) State Changes** will immediately send a HIP update if the state of WSC changes. Setting this to **No** will prevent the update.
58. Enable **Inbound Authentication Prompts** from MFA Gateways can be set to **Yes** if additional authentication policies are going to be used to protect vulnerable internal resources that require an additional MFA authentication. In this case, the **GlobalProtect** agent will present the MFA landing page and pop up a warning message so the user is made aware of why an additional MFA window appears.
59. **Network Port for Inbound Authentication Prompts (UDP)** specifies on which port inbound MFA prompts are received (default is **4501**).
60. **Trusted MFA Gateways** specifies a list of trusted firewalls or authentication gateways, which will limit which inbound MFA prompts are allowed.
61. **Inbound Authentication Message** customizes the inbound MFA message.
62. **IPv6 Preferred** sets which IP version to prefer. **No** prefers IPv4, and **Yes** prefers IPv6 in a multi stack environment.
63. **Device Added to Quarantine Message** can display a custom message if the device was put into quarantine.
64. **Device Removed from Quarantine Message** displays a custom message when the device was removed from quarantine.

In the **HIP Data Collection** tab, you can select whether to collect HIP data or not. If a GlobalProtect license has been purchased, each GlobalProtect agent on Windows or macOS will send a HIP report about running processes, patch levels, and so on to the gateway when they connect, and periodically afterward. You can exclude several categories and vendor products from being collected, or add custom checks to Windows and macOS hosts for specific registry entries to be present or processes to be running.

If you want to collect HIP data, set the **certificate profile** to verify the machine certificate sent by the GlobalProtect agent. Once you've reviewed HIP checks, click **OK** to complete the agent configuration.

You can also add a clientless VPN, which provides users with a portal page with clickable links to internal applications, without needing to install VPN software.

Clientless VPN

In the **Clientless VPN** tab of the portal configuration, you can create a portal interface that allows users to connect to a web page and have access to internal applications without needing to set up a full tunnel. It works by populating the **Portal** page with tiles that lead to application interfaces:

1. In **General**, enable **Clientless VPN**:

    - Set the **FQDN** or **IP** of the portal.
    - Set the security zone. This will be the source zone for outgoing proxied connections from the firewall to the application.
    - Select a **DNS Proxy** object. Create one if you don't have one yet; it does not need to be attached to an interface for it to work with **Clientless VPN**.
    - Change **Login Lifetime** and **Inactivity Timeout** if the defaults (`3 hours` and `30 minutes`, respectively) are not suitable.
    - In **Max User**, select the maximum number of concurrent users. The default is `10`.

    The configuration should look similar to the following screenshot:

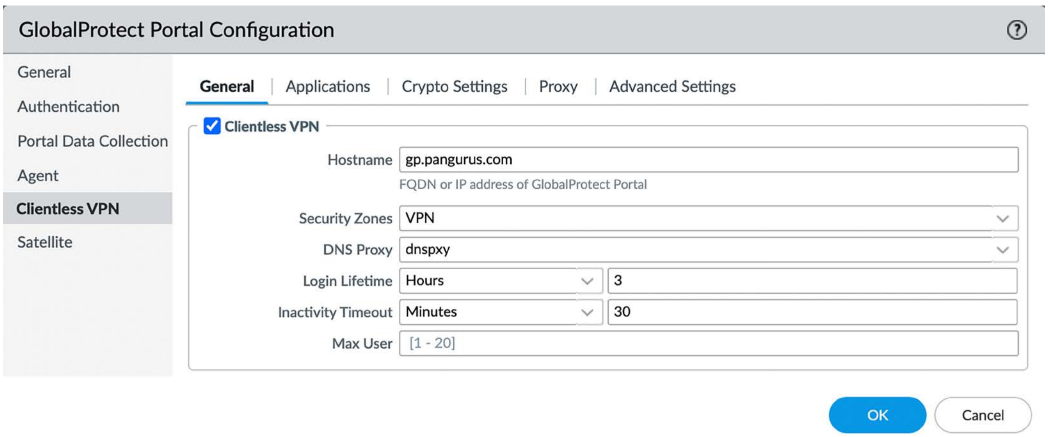


Figure 11.11: Clientless VPN

2. In the **Applications** tab, you can select the clientless applications that are available to users. You can either create these individually in **Network** | **GlobalProtect** | **Clientless Apps** and then create **Clientless Apps** groups in **Network** | **GlobalProtect** | **Clientless App Groups** or you can create apps directly with the following steps:

    - Click **Add** to add new application-to-user mappings.
    - **Display application URL address bar** lets a user input custom URLs, which the clientless VPN will proxy for them. Disable this option unless your users are allowed to browse the internet via a clientless VPN.
    - If an application should only be visible to a specific user or user group, click on **Add** and select the users(s) or group(s) you want it to be visible to.
    - In **Applications**, click on **Add** and select an application, or create a new clientless app:

        a. When creating a new clientless app, set a name so that the user will be able to identify the application.

        b. Set an appropriate URL for the application.

        c. Add a description with additional details.

        d. Optionally, upload an icon for the **Clientless App** tile.

        e. Click **OK**.

    Create or add additional applications as needed. The **Applications** tab should look similar to the following screenshot:

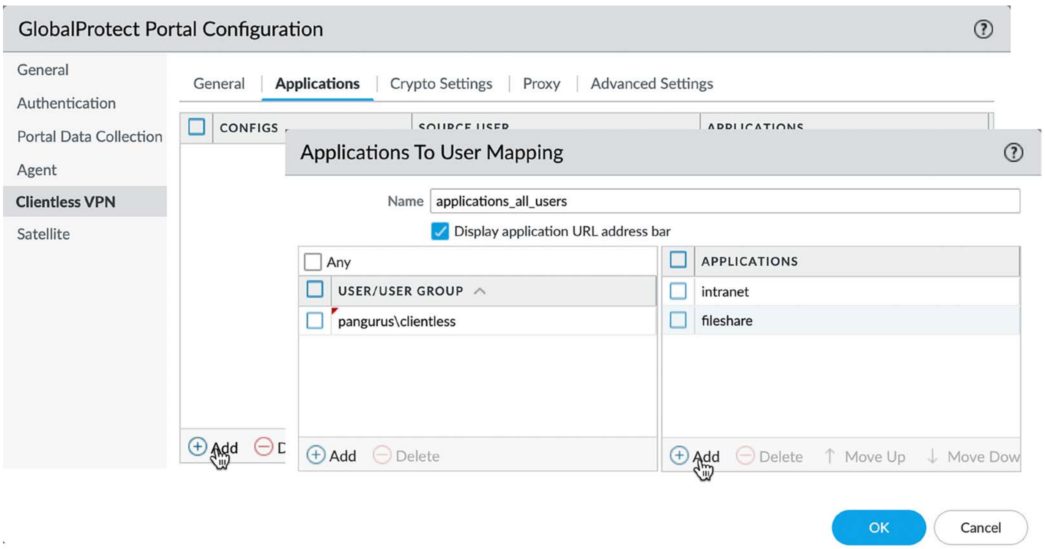


Figure 11.12: Clientless applications

3. In the **Crypto Settings** tab, you can control the security of the outbound connections from the firewall to the applications:

    - Set **Min Version** to **TLSv1.2**.
    - Disable **SHA1** as this is no longer considered a secure algorithm. It should only be used if there is no alternative available to communicate with the remote peer.
    - Enable all the server certificate verifications unless some internal certificates are known to be problematic.

> If you're also going to allow applications outside of your network, it is a good practice to enable all the block options under **Server Certificate Verification** to prevent users from accessing sites with "bad" certificates.

The **Crypto Settings** tab will look similar to the following screenshot:

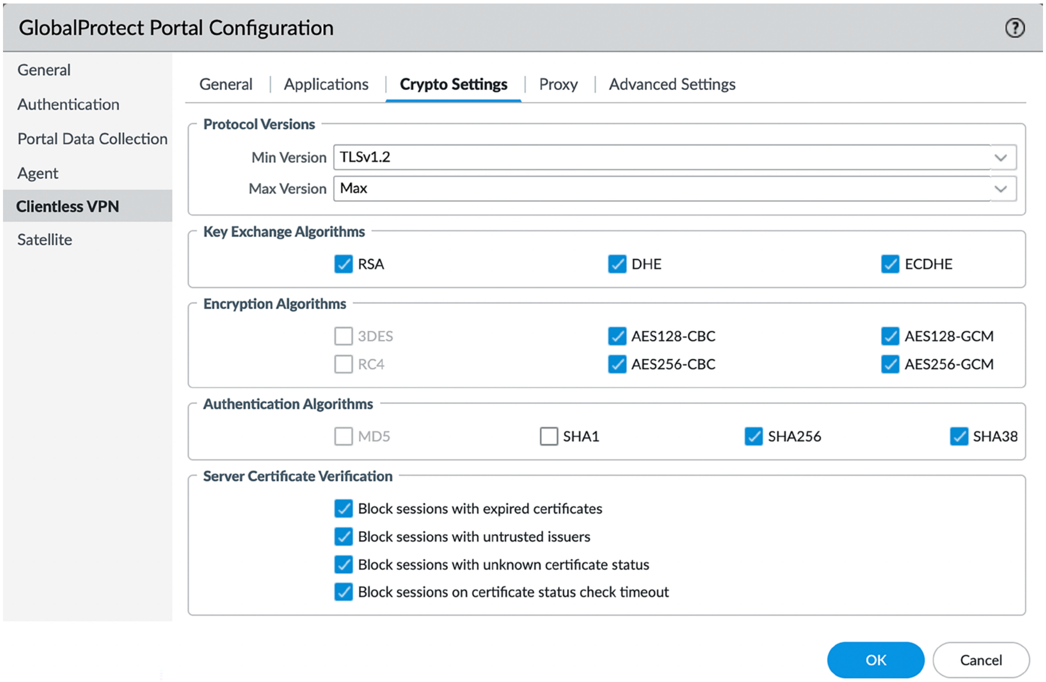


Figure 11.13: Clientless Crypto Settings

4. In the **Proxy** tab, additional proxy servers can be configured if the outbound connections need to pass through a proxy server. Proxy rules can be configured for specific domains and processed from top to bottom, so put the most specific ones at the top:

    - Click on **Add** and set a descriptive name
    - Add the domains that need/don't need to be proxied, one per line

- Check or uncheck the **Use Proxy** box
- Fill out the proxy IP or FQDN, port, and credentials details
- Click **OK** and add additional proxy server settings as needed

Here is an example of a proxy setting for an intranet page:

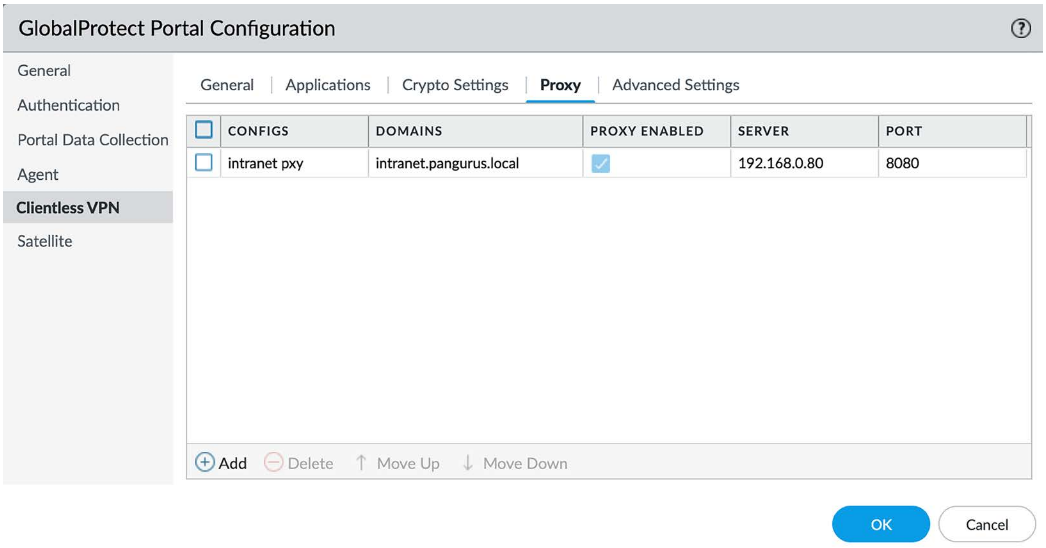


Figure 11.14: Clientless Proxy

5. In **Advanced Settings**, you can add exclusions for any applications that have a sub-page or reference links that should not be accessed through the portal. Adding a URL to the list will prevent clientless VPN users from accessing such links. Paths are not supported, however.

In the following screenshot, you can see what the **GlobalProtect** portal looks like with some clientless applications and the **Application URL** enabled. The **GlobalProtect Agent** can be downloaded from here as well:

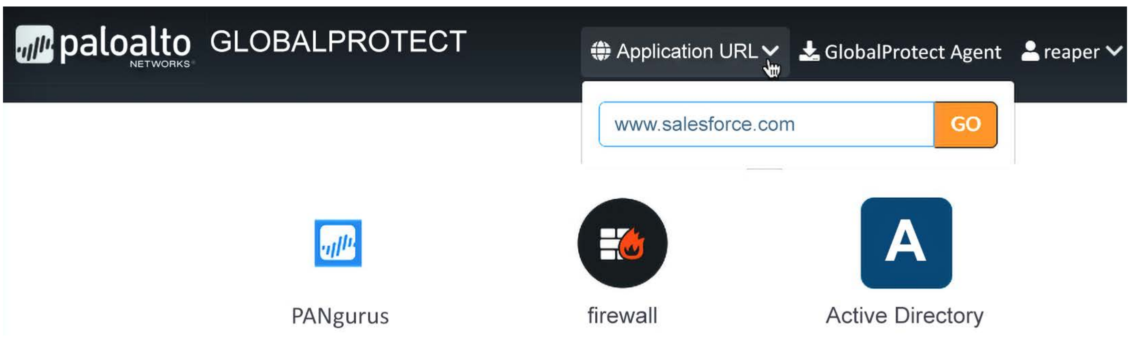


Figure 11.15: Clientless VPN-enabled GlobalProtect portal

In the **Satellite** tab, you can configure firewall appliances that will use a simplified VPN to connect to the organization. This is an ideal solution if, for example, several smaller firewalls are being used to set up pop-up locations or operate a booth at conventions, to quickly set up (and break down shortly after) the VPN tunnels, so the remote team has access through an actual firewall for additional security over the GlobalProtect agent. Larger or static sites will benefit most from a traditional VPN connection.

Follow these steps to create a satellite group:

1. Click on **Add**.
2. In **General**, set a descriptive name and review **Config Refresh Interval**. This sets the cadence for how frequently satellites check whether there is a new connection configuration available.
3. In the **Devices** tab, add devices by their serial number and set a descriptive name for each device.
4. In the **Enrollment User/User Group** tab, you can add users or groups of users that are allowed to manually enroll devices. If a new device is set up in the field and the serial number has not been communicated for it to auto-enroll, the admin will be prompted to manually enroll, at which time their username or group membership must match the one you set here.
5. In the **Gateways** tab, configure which gateways the satellite will connect to and what their routing priority will be. As opposed to GlobalProtect agents, which connect to the fastest or highest priority gateway, satellites connect to all the gateways you configure and use routing priority to direct traffic.
6. Click **OK**.
7. Click **OK** to complete the portal configuration.

Now that the portal is configured, you can start adding the internal and external gateways.

Setting up the gateway

Gateways are what the agents connect to. Each firewall can have multiple gateways but they can't share an IP address, so if multiple gateways are needed, they will each require a unique IP. A portal and a gateway can share the same IP.

> **Important note**
>
> Gateways cannot coexist with an HTTPS-enabled management profile (*never* set a management profile that hosts services on an external or untrusted interface). In these cases, you should enable the management profile on a loopback interface and use port address translation; so, for example, port 4443 is translated to 443 on the loopback interface. Or (preferably) rely on GlobalProtect to provide a secured connection before connecting to an internal loopback with management enabled.

To create a new gateway, go to **Network** | **GlobalProtect** | **Gateways** and follow these steps:

1. Click on **Add** and provide a descriptive name.
2. Select the appropriate interface and select **IPv4**, **IPv6**, or **IPv4 and IPv6** and set the IP. Just like the portal part, it is good practice to set the gateway on a loopback interface.
3. In the **Authentication** tab, set the following:
    - Use the same SSL/TLS service profile as the portal if you reuse the same FQDN or have the certificate set to an IP. If you want to use a different FQDN, import or generate the appropriate certificate and create a new SSL/TLS service profile.
    - To use client certificates, set **Certificate Profile**.
    - Create a new client authentication profile, set a descriptive name, and select the appropriate authentication profile. You can use the same profile as the portal or create a new one that leverages MFA for added security. Review whether you need credentials and a client certificate, or credentials or a client certificate.

In the **Agent** tab, several considerations can be made that will alter the user experience:

1. In the **Tunnel Settings** tab, set the following:
    - **Tunnel mode** must be enabled for external gateways. On an internal gateway, you can leave **Tunnel mode** disabled if you intend to use the gateway for HIP and authentication and identification only. If **Tunnel mode** is enabled, the agent will set up a tunnel even when inside the network for added security.
    - Select a tunnel interface. This interface can be created in **Network** | **Interfaces** | **Tunnel** and does not require an IP address, but it does need to be set in the appropriate virtual router and in a different security zone than the local network. Check the box next to **Enable User-ID** in the zone configuration.
    - Leave **Max Users** empty to allow the maximum number that your platform supports.
    - Unchecking **Enable IPSec** will force the use of SSL/TLS. If IPSec is checked but is unavailable from the agent's location, the fallback protocol is SSL/TLS.
    - Create a new **GlobalProtect IPSec Crypto** profile from the dropdown that uses a GCM cipher (the default uses aes-128-cbc).
    - The **Enable X-Auth support** option enables third-party VPN clients that support X-Auth to connect to a gateway. Enable this if you want to connect OpenVPN, for example. Set **Group Name** and **Group Password**. You can force the user to need to reauthenticate when the IPSec key expires by unchecking **Skip Auth on IKE Rekey**.
2. In the **Client Settings** tab, you can control how the gateway interacts with a subset of users, the host OS, or the region. Skip this step if split tunneling configuration is not required:
    - Click on **Add** and set a descriptive name.
    - Select the source user, host OS, region, or source addresses that should apply to the intended users. Otherwise, leave them all as **Any** if all users should fall into this profile.

- In the **Authentication Override** tab, select whether an override cookie should be generated, whether it should be accepted, and the amount of time the cookie should be valid for, and then select the encrypt/decrypt certificate for the cookie. It should be signed by the trusted root certificate associated with the portal and gateway certificates.
- In the **IP Pools** tab, you can set regular IP pools, which will be assigned from top to bottom, or you can enable **Retrieve Framed-IP-Address attribute from authentication server** if your authentication server supports the **Framed-IP-Address** attribute so that clients are assigned a static IP address when they log in. **Authentication Server IP Pool** needs to be large enough to support all users as these are static assignments that are not cleared after a user logs off.
- In the **Split Tunnels** tab, you can configure which route sessions should take. If left blank, all sessions will be sent over the tunnel. Direct access to the user's home network (or hotel, airport, coffee shop, and so on) can be disabled by checking **No direct access to local network:**
    - Subnets added to the **Include** field will cause the GlobalProtect agent to *exclusively* route sessions destined for these subnets over the VPN tunnel.
    - Subnets added to the **Exclude** field will not be sent over the tunnel and will use the client's local routing table.
    - **Include** and **Exclude** domains and their associated *ports* can be used to control which FQDNs are or are not sent over the tunnel. This feature requires a license.
    - **Include and exclude Client Application Process Name** lets you control which processes (`C:\Users\user\AppData\Local\myapp\myapp.exe`, for example) will send all of their traffic over the tunnel or are disallowed from using the tunnel and "break out" to the internet over a local link. This feature also requires the GlobalProtect license.

3. The **Client IP Pool** tab holds the global IP pool for all users connecting to the gateway matching the current profile. Multiple pools can be added, both IPv4 and IPv6. IP addresses are assigned from the top pool first; once this is depleted, the next pool will be used, and so on. The IP pools will automatically be added to the virtual router that the tunnel interface belongs to.
4. In the **Network Services** tab, you can control which DNS and WINS servers the agent receives when a connection is established, and the DNS suffixes relevant to your organization. If a dynamic or DHCP client interface exists, this can be set as an inheritable source for DNS and WINS information to be passed along to GlobalProtect agents.
5. In the **Connection Settings** tab, set the following:
    - **Login Lifetime** sets the maximum amount of time a user is allowed to be connected continuously (default is **30** days).
    - **Notify before lifetime expires** notifies the user their session is about to time out and they should expect to reauthenticate (default is **30 minutes**).
    - **Login Lifetime Expiration Message** allows you to customize the notification message.

- **Inactivity Logout** disconnects the user after no traffic has been sent through the tunnel for the specified amount of time.
- **Notify before Inactivity Logout** pops up a message letting the user know their session is about to idle out (default is **30 minutes**).
- **Inactivity Logout Message** allows you to customize the inactivity logout message.
- **Notify user on administrator initiated logout** can be enabled to notify the user an admin has logged them out.
- The **Disable Automatic Restoration of SSL VPN** option will prevent automatic reconnection after the connection is interrupted for any reason, requiring the user to manually reconnect. This option will prevent always-on mode from working.
- **Restrict Authentication Cookies** lets you set limitations to the authentication override by restricting the cookie to only work on the original source IP or the subnet that the cookie was created for (if the user shifts to a different IP or subnet, the cookie will no longer work for authentication override and the user will need to reauthenticate).

6. In the **Video Traffic** tab, you can force video applications to use the local internet breakout instead of the tunnel to conserve bandwidth. Any video streaming service that is not allowed should not be excluded and is instead blocked on the firewall by the security policy. This feature requires a license.
7. In the **HIP Notifications** tab, you can create profiles containing HIP objects or HIP profiles and their **User Notification** settings:

- Click on **Add** and select the HIP profile or the HIP profile to match (see the following bullet points).
- If a match needs to be reported to the user, set **Enable Match Message**, set a system tray balloon or pop-up message, and type the text that needs to be displayed to the user.
- If a required check was not detected (**not-match**) and this event needs to be reported to the user, set **Enable Not Match Message**, set a system tray balloon or pop-up message, and type the text that needs to be displayed to the user.

8. In the **Satellite** tab, you can configure the tunnel settings for Satellite firewalls.
9. In the **Tunnel Settings** tab, set **Enable Tunnel Configuration** and set a tunnel interface. Since these will be branch offices, you should use a different tunnel interface, with an IP assigned, and a different security zone from the one that the regular gateway is using. The tunnel monitoring settings are the IP addresses that the remote gateways will use to monitor connectivity and fail over to a different gateway if monitoring fails. Set this to the tunnel IP.
10. In the **Network settings** tab, DNS settings and DNS suffixes can be set, or an inheritance source can be set. The IP pool will be used to assign an IP to the remote tunnel interface. Access routes let the remote peer set routes into the tunnel to reach the main site's network. Leave this blank to send everything into the tunnel.

11. In the **Route Filter** tab, you can enable **Accept published routes** to install routes advertised by the satellites into the virtual router. To prevent overlaps with local subnets, you can add subnets that will be accepted this way into the **Permitted Subnets** field.
12. Click **OK**.

You now have a fully functional gateway that your users can start connecting to. If you want to perform HIP checks, here's how to set those up.

HIP objects and profiles

HIP checks verify whether the agent's host OS lives up to the standards set forth by your organization. Remember that a license is required to perform these checks on your hosts.

Before we begin, verify that the GlobalProtect data file is being downloaded periodically in **Device | Dynamic Updates**. This will ensure that the firewall has current information on vendor patch levels and software versions.

A HIP object would typically cover one type of device for manageability, as there may be managed Windows and macOS laptops, company-owned mobile devices, and BYOD devices. All of these will have different characteristics. You can create HIP objects in **Objects** | **GlobalProtect** | **HIP Objects**. Follow these steps to build a basic HIP object:

1. Click on **Add** and set a descriptive name.
2. In the **General** tab, provide all the relevant host information, such as the OS version, the GlobalProtect client version, the domain, and, for mobile devices, which Wi-Fi network or carrier they are connected to.
3. In the **Mobile Device** tab, you can enable this profile for mobile devices and set parameters for the types and models of the device, the phone number, and the IMEI number. You can have HIP verify whether the passcode is enabled, the mobile device is jailbroken, disk encryption is enabled, and whether certain applications are installed.
4. In the **Patch Management** tab, you can set detection for missing patches by severity level and different vendors. These patch signatures are included in the Dynamic Updates package.
5. In the **Firewall** tab, you can enable detection if the firewall software is installed and enabled.
6. In the **Anti-Malware** tab, you can enable detection for installed antivirus or anti-malware software and see whether real-time scanning is enabled, check the minimum virus definitions and the product version, and see when the last scan took place.
7. In the **Disk Backup** tab, you can enable detection for backup software and see when the last backup was run.
8. In the **Disk Encryption** tab, you can enable detection for encryption software and see whether certain locations have been encrypted.
9. In the **Data Loss Prevention** tab, you can enable detection for data loss software and see whether it is enabled.

10. In the **Certificate** tab, you can verify whether the certificates used by GlobalProtect have specific attributes set.
11. In the **Custom Checks** tab, you can add checks for running processes and registry or Plist keys.

In **HIP Profiles**, you can combine HIP objects through **AND**, **OR**, and **NOT** conditions, which allows you to build a set of conditions that apply to many devices. Once you add these conditions to GlobalProtect or the security policy, security controls can be applied to users meeting or failing said checks.

A HIP profile could, for example, be set as follows:

```
("corp-laptop" or "corp-mobile") and not "byod"
```

This can be done to include all the corporate devices, but not the private ones.

To create security rules that leverage HIP profiles, do the following:

1. Create a new security rule and set a descriptive name.
2. In the **Source** tab, set the GlobalProtect security zone and create and set a user IP pool object.
3. In the **Source User** column, set the user group, and in the **Source Device** column, set the HIP profiles to apply this rule to. Only devices matching the HIP objects in the profile will match this rule.
4. You can set the **HIP** dropdown to **no-hip** if this rule does not require HIP information to be available from the client, which allows third-party VPN clients to access resources, while **any** will allow any device. In the **Destination** tab, set an appropriate destination, such as the DMZ servers or other internal resources.
5. Add appropriate applications in the **Applications** tab.
6. Set services or the destination URL categories in the **Service/URL Category** tab.
7. Set the action, threat profiles, and logging settings in the **Actions** tab.
8. Click **OK**.

You can also set a HIP match for **Quarantine**, which will include any devices that the administrator has manually added to quarantine by adding the device through **Device** | **Device Quarantine** or by manually selecting it in a traffic or threat log, or any devices that were added to quarantine automatically by matching a security rule with a log forwarding profile that has a quarantine action set, as shown in the following screenshot:

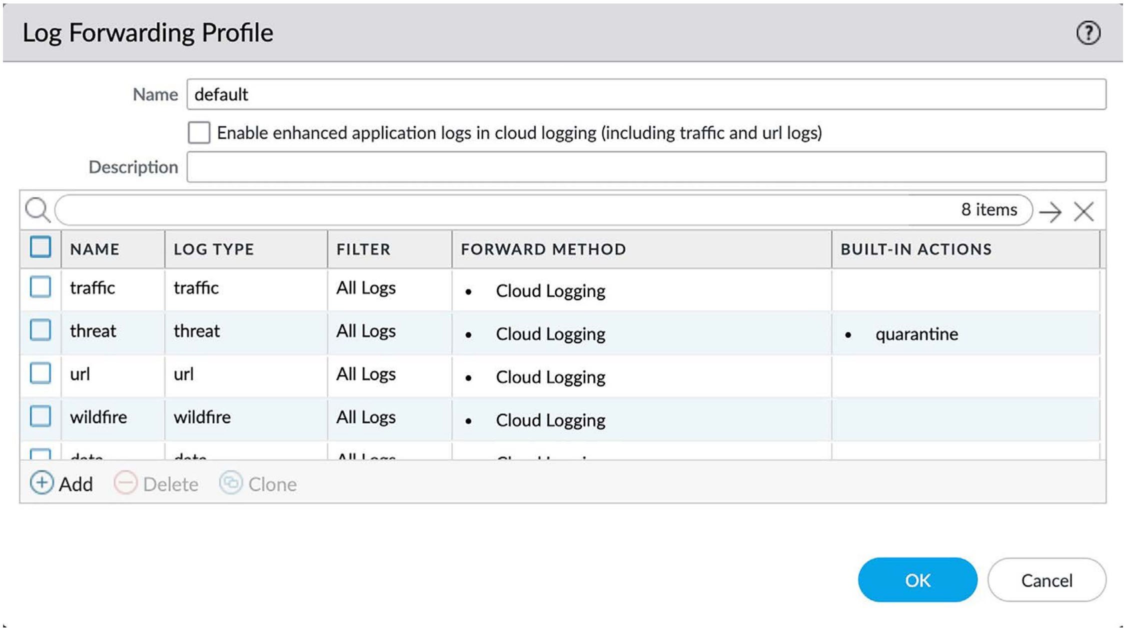


Figure 11.16: Automated quarantine

A rule base for HIP-enabled clients could look something like the following screenshot. Each rule is for the same zone, user, and IP pool, but the HIP matches are different for each rule, so they will apply to different source devices:

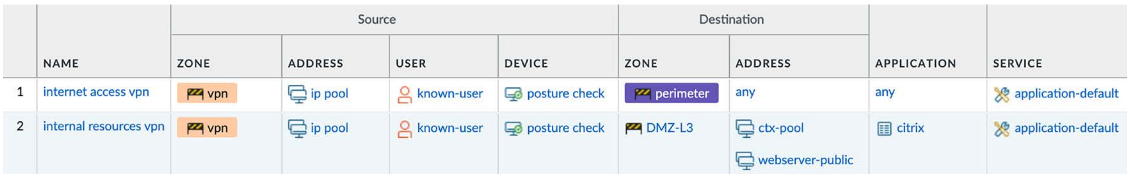


Figure 11.17: HIP-enabled security rules

By default, agents send a HIP update every hour. If you wish to change this interval, it can only be changed from the CLI with the following commands:

```
> debug global-protect portal interval <60-86400>
> configure
# commit
# exit
> debug global-protect portal show
```

In this section, you learned how to set up GlobalProtect components and provide users with a flexible way to work from anywhere from all types of devices.

Summary

In this chapter, you learned how to set up site-to-site VPN tunnels and a client-to-site VPN with GlobalProtect. You can now not only provide connectivity but also scan the client machine for compliance and know how to control the user experience.

If you're preparing for the PCNSE, remember that the clientless VPN is a proxied connection and that applications must be created. You'll need to understand the difference between the GlobalProtect portal and gateway and know which features require an additional license (mobile clients, split tunnels for applications and domains, HIP checks, clientless VPN, IPv6, and split DNS).

Another frequently asked question is how the gateway is selected when multiple gateways are available; see *Setting up the portal* earlier in this chapter for more details. In short, the answer is a combination of TLS responsiveness and gateway priority in the agent configuration. Highest priority will always be preferred unless its responsiveness is lower than the average of all other gateways.

In the next chapter, we will learn about creating custom applications and custom signatures for threat prevention, and how to apply zone protection and protect individual services using DoS protection profiles and policies.

12

Advanced Protection

In this chapter, we will learn about advanced configuration features, such as custom applications and custom threats, and apply them to a policy. We will also learn how to create our own threat signatures so we can block certain payloads, and we will review how zone protection and **Denial-of-Service** (**DoS**) protection can defend the network and individual resources from attackers.

In this chapter, we're going to cover the following main topics:

- Creating custom applications and application overrides
- Creating custom threat signatures
- Implementing zone protection and DoS protection

Technical requirements

In this chapter, we will be covering remote connections and protection from inbound connections. If you have a lab environment where you can simulate custom applications and incoming scans or floods, this will help greatly in visualizing what is being explained.

Creating custom applications and application overrides

Every once in a while, an application may not be found in Applipedia (available applications downloaded via content updates, which you can find in **Objects** | **Applications**, or `applipedia.paloaltonetworks.com`). This could be due to it being a new application that has not been used much in the wild or it could be something a developer created in-house for which it is not reasonable to expect there to be signatures to identify the session.

In these cases, it is possible to create custom applications that use custom signatures and can trigger an App-ID to positively identify the previously unknown application.

The need for a custom application usually starts with the discovery of an abnormality in the traffic log. In the following screenshot, I have discovered my solar power converter, an IoT device, is communicating with its home server over an **unknown-tcp** connection:

(app eq unknown-tcp) and (addr.src in 192.168.27.4)

		RECEIVE TIME	TYPE	FROM ZONE	TO ZONE	SOURCE	DESTINATION	TO PORT	APPLICATI...	ACTION
		06/14 17:09:28	end	LAN	outside	192.168.27.4	78.	22222	unknown-tcp	allow
		06/14 14:43:08	end	LAN	outside	192.168.27.4	79.	22222	unknown-tcp	allow
		06/14 13:25:23	end	LAN	outside	192.168.27.4	46.	22222	unknown-tcp	allow

Figure 12.1: An unknown-tcp application in the traffic log

There are two ways to address this issue through the creation of a custom application:

- Implementing an application override that forcibly sets all these sessions to a specific custom application
- Creating a custom application using signatures to positively identify these sessions, and still performing security scans on the sessions

Let's take a look at the easiest solution first.

Application override

Implementing an app override is "quick and dirty"; it forcibly replaces the application identification process with a custom application. The advantage is that you simply set a few simple parameters and you are done. The drawbacks are that there is no granularity, there is room for mistakes, and, most importantly, if you set a custom application, the security profiles will no longer apply to the sessions (packets will no longer be scanned for threats and malware).

> **Important note**
>
> Setting a predefined application could help "fix" an otherwise broken App-ID process if the data flow is somehow different than what would normally be expected from the application, causing the regular App-ID to fail. This will only work if the application flow exactly matches the application being set in the override, with the *rare* condition of some key packets being out of order. I wouldn't recommend this as a fix-all, but keep it in your pocket for a rainy day.

In the following examples, I will use my solar converter, which uses a custom data flow, as an example of how to go about creating custom applications. It could help to see whether there is an IoT device in your network to follow along with, creating a custom application for it.

The first step is to create a custom app that will be used to identify the session. Create a new application in **Objects** | **Applications** as follows:

1. Click on **Add** and set a descriptive name for the new application. In this case, we will call the application `Solar`.

2. In the **Configuration** tab, set the **Properties** and **Characteristics** settings. For my solar converter, we'll set the following:

    - **Category:** `business-systems`
    - **Subcategory:** `iot-management`
    - **Technology:** `client-server`

    We'll leave all the characteristics blank as this is a simple app, calling home to report on my solar gains.

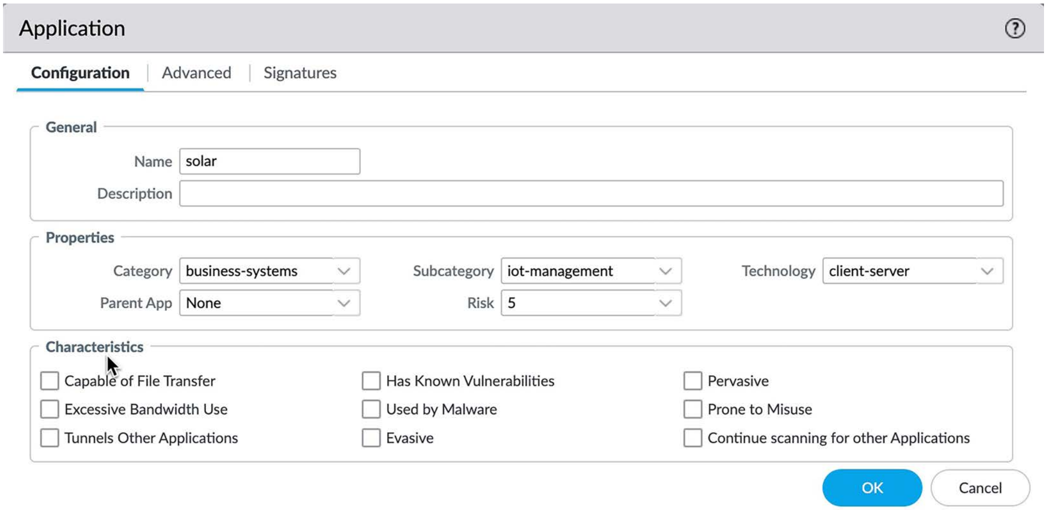


Figure 12.2: Custom application for solar converter

3. In the **Advanced** tab, you can select to use TCP and UDP ports by checking the **Port radio** button or select an **IP Protocol**, **ICMP Type**, **ICMPv6 Type**, or **None**.
4. In the port settings, you can add `tcp/` or `udp/` followed by a port number (such as `tcp/88`), a port range (such as `udp/50-100`), or dynamic (such as `tcp/dynamic`) for dynamically assigned ports. We will set the following:

    - Set **Port** to `TCP/22221-22222`
    - I'll leave all the **Timeouts** settings blank to indicate that I wish to use the system default timeouts for TCP
    - There's an option for scanning **File Types**, **Viruses**, and **Data Patterns**, but this will only work if there is no override in place, so I will leave these blank for now as well

5. We do not need the **Signatures** tab right now, so we can click **OK**.

The application now looks as in the following screenshot:

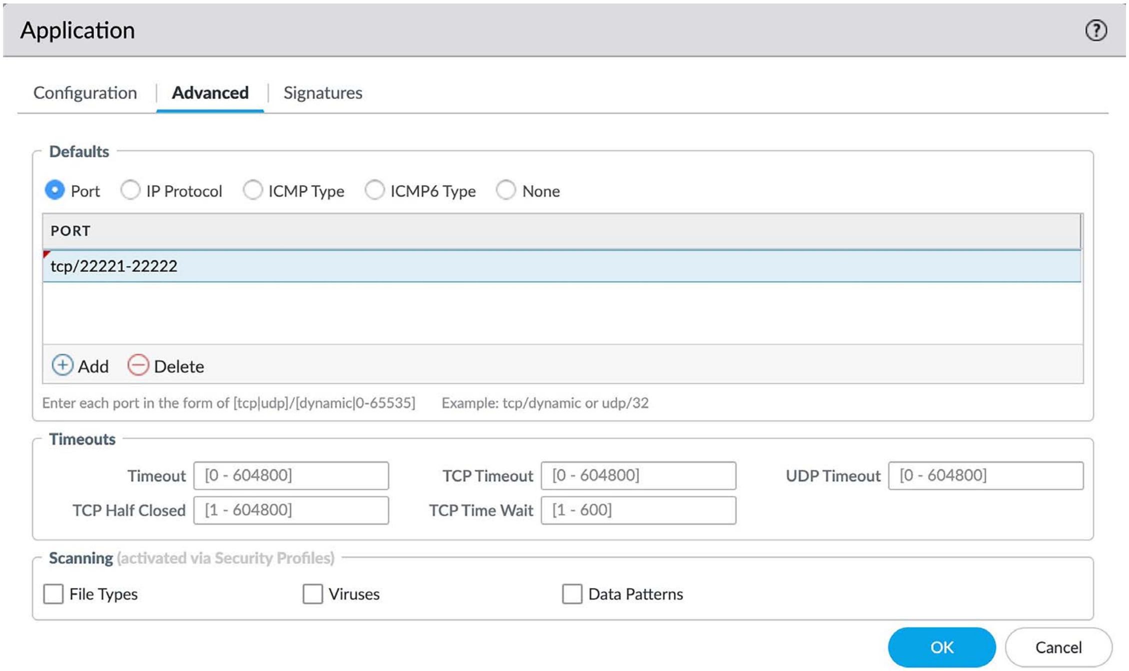
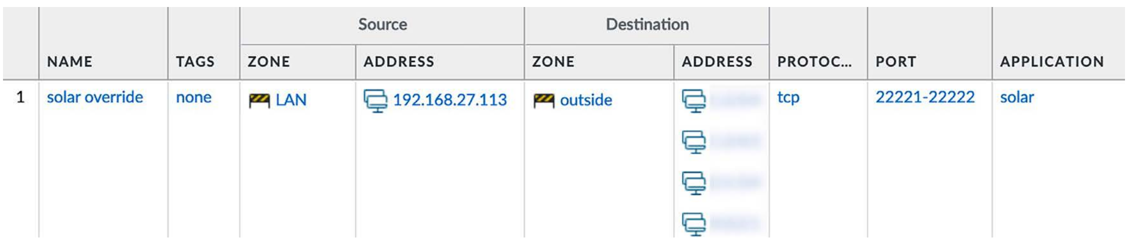


Figure 12.3: Custom application Advanced tab

To create the override, go to **Policies** | **Application Override** and create a new override policy:

1. Click on **Add** and set a descriptive name.
2. In the **Source** tab, I'll set the source zone to **LAN** and the source IP to **192.168.27.113** for my solar converter.
3. In the **Destination** tab, I will set the destination zone to **Outside** and the IP addresses associated with my converter's cloud interface.
4. In the **Protocol/Application** tab, set the destination ports to **tcp 22221-22222** and the **solar** custom application that we created earlier.
5. Click **OK** and **Commit**.

The **override** rule will look as in the following screenshot:



			Source		Destination				
	NAME	TAGS	ZONE	ADDRESS	ZONE	ADDRESS	PROTOC...	PORT	APPLICATION
1	solar override	none	LAN	192.168.27.113	outside		tcp	22221-22222	solar

Figure 12.4: Application override rule

Once the changes are committed, you should start seeing the sessions show up as a different application in your session table and traffic log, as you can see in the following screenshot:

		RECEIVE TIME	TYPE	FROM ZONE	TO ZONE	SOURCE	DESTINATION	TO PORT	APPLICATION	ACTION
		06/17 00:57:18	end	LAN	outside	192.168.27.2...	80.	22222	solar	allow
		06/16 19:49:46	end	LAN	outside	192.168.27.2...	80.	22222	solar	allow
		06/16 16:29:25	end	LAN	outside	192.168.27.2...	18.	22222	solar	allow

Figure 12.5: The session identified as a custom application

While this is a great solution for simple applications, especially internal ones where you have control over the endpoints and can leverage host-based security to make up for the lack of scanning capabilities on the TCP flow, it is better to use signature-based identification and let the App-ID and Content-ID fully scan the flow.

Don't forget to disable the application override policy before moving on to the next section.

Signature-based custom applications

Identifying applications based on a signature or signatures provides far more accuracy when identifying custom sessions. Any sessions that do not match the signatures you set to identify the traffic will still be identified as unknown, which should either be blocked or raise an alarm if you have accounted for all possible signatures.

We first need to do some research into the application we want to identify before we can create custom signatures. Packet captures provide the best information for this.

To set up a basic packet capture, go to **Monitor** | **Packet Capture** and click on **Manage Filters**.

In **Packet Capture Filter**, you can add up to four lines that tell your system what you want to capture, based on **Ingress Interface**, **Source IP or Port**, **Destination IP**, or **Port and Protocol Number**. You can also opt to include, exclude, or exclusively capture non-IP protocols, which is helpful if you're trying to capture DHCP. So, for example, in my case, I'd do the following:

1. Click on **Clear All Settings** to remove any pre-existing filters.
2. Click **Manage Filters** to start adding filters.
3. Click on **Add** and set the filter ID to `1`.
4. Select **Source** and set the IP of my solar converter, `192.168.27.113`.
5. Set the destination port to `22221`.
6. Click on **Add** and set the filter ID to `2`.
7. Select **Source** and set the IP of my solar converter, `192.168.27.113`.
8. Set the destination port to `22222`.
9. Click **OK**.
10. Enable filtering by setting the **Filtering** toggle to **ON**.

The packet capture filter will look like the following screenshot:

Figure 12.6: Packet Capture Filter

Then, configure capturing by doing the following:

1. Click on **Add**.
2. Set a capture stage:
    - **receive** captures packets on the incoming interface
    - **transmit** captures packets on the outgoing interface
    - **drop** captures packets that are being discarded
    - **firewall** captures packets while they are being processed

    For this exercise, we will use the **firewall** stage and call the file `solar.pcap`.
3. Click **OK**.
4. Enable capturing by switching the **Packet Capture** toggle to **ON**.

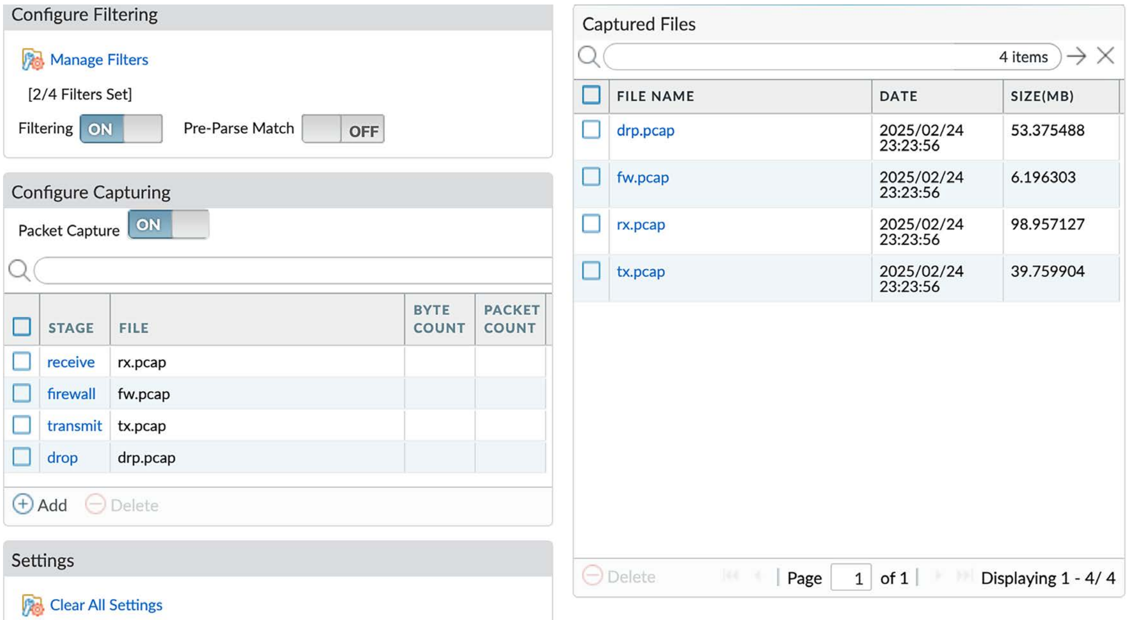


Figure 12.7: Packet Capture

Once packets have been captured, the file will appear in **Captured Files**, where you can click on the file to download it. Wait a sufficient amount of time and then, if possible, restart the session. Once enough data is collected, click on the file and open it with Wireshark to start looking for signatures.

In my case, I discovered that my solar converter will always sign in using the same fingerprint, as you can see in the **Data** field of the fourth packet in the following screenshot:

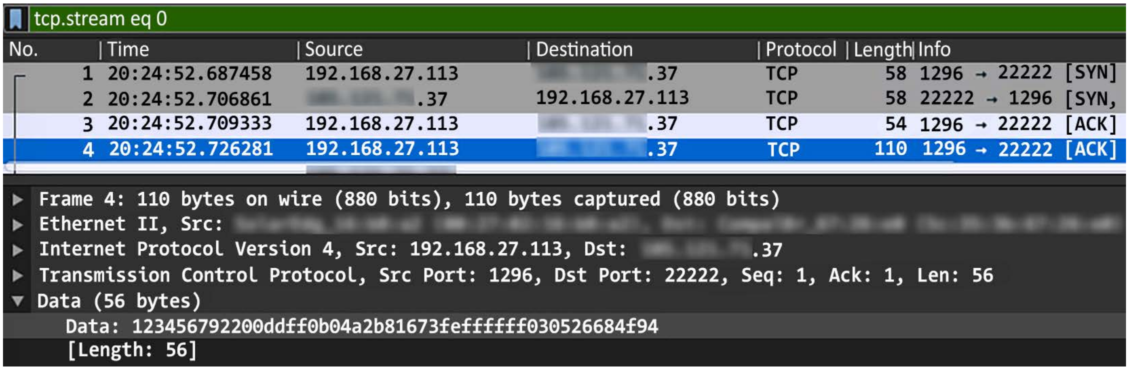


Figure 12.8: Packet capture in Wireshark

We can now add this to the custom application we created earlier. Go to **Objects** | **Applications** and open the custom application (`solar`). In the **Signatures** tab, click on **Add**:

1. Set a descriptive signature name.
2. Set the scope. **Transaction** is used to match a signature in a single packet and **Session** is used to match signatures across multiple packets.
3. I'll set my scope to **Transaction** since the fingerprint identification happens in the fourth packet and we don't need the signature engine to keep analyzing after it identifies the fingerprint.
4. **Ordered Condition Match** requires multiple conditions to be matched in order from top to bottom. With this option unchecked, they can be matched in any order. By default, this setting is enabled, which does not influence a single condition, but keep this in mind if you add multiple conditions that should be matched in differing order.
5. Add an `OR` condition:
   - Set **Operator** to `Pattern Match`.
   - The context for this signature is `unknown-req-tcp-payload` as there is no decoder that *claimed* this session (`unknown-tcp`).
   - Many different contexts are available depending on the decoder that picks up on a session. If the custom app is a sub-application to `web-browsing`, for example, the context could be `http-req-host-header`.
   - Set the pattern. To match ASCII, just add the ASCII text in the field, and to match the hexadecimal value, you must enclose the hex between two `\x` tokens, which lets the signature engine know that this is a hexadecimal value.
   - We'll use `\x123456792200dd\x` to match the fingerprint, which meets the 7-byte minimum for a custom signature.

- Some contexts can have **qualifiers** that filter where a string can be matched (for example, for `http-req-host-header`, you could add the `http-method` qualifier with the GET value).

6. Click **OK** twice.

The custom application will now look as in the following screenshot:

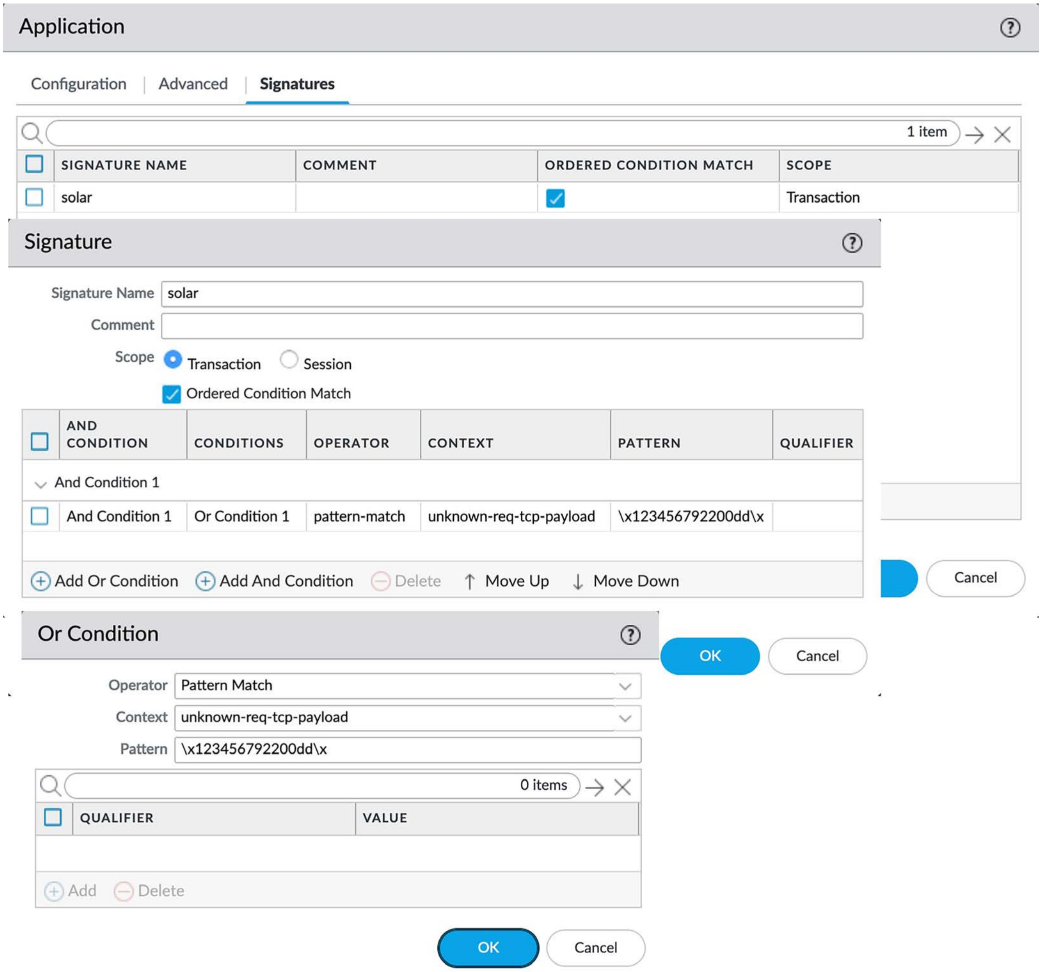


Figure 12.9: Custom application with a signature

Once you commit this, you should start seeing the sessions being picked up as the custom application.

A few notes on creating signatures:

- A signature pattern must contain at least a 7-byte string with fixed values.
- Enclose hexadecimal strings in `\x`.
- Be mindful of uppercase and lowercase letters in ASCII. You may need to include a signature for both if there could be instances where one is used versus the other (for example, `GOOGLE.COM` versus `google.com`).
- Outside of the 7-byte string, you can add **Regular Expressions** (**RegExes**) to match more complex patterns.

The following characters can be used as wildcards in a RegEx string:

RegEx	Example	Description
.	`1.3`	Matches a single character (e.g., `123`, `133`)
?	`dots?`	Matches string with or without last character (e.g., `dot`, `dots`)
*	`dots*`	Matches string with or without last character, and multiple repeats of last character (e.g., `dot`, `dots`, `dotssss`)
+	`dots+`	Matches single or multiple repetitions of the preceding letter (e.g., `dots`, `dotssss`)
|	`((exe)|(msi))`	OR function to match multiple possible strings (e.g., `dot.exe`, `dot.msi`)
[]	`X[abc]`	Matches preceding string followed by any character between square brackets (e.g., `xa`, `xb`, `xc`)
-	`X[a-z]`	Matches any character in a range (e.g., `xa,xm`)
^	`X[^AB]`	Matches any character except the ones listed (e.g., `xC`, `x5`)
{}	`X{1,3}`	Matches anything after x as long as it is 1 to 3 bytes in length (e.g., `x1`, `x123`)
\	`X\.y`	Escape character to exactly match a special character (e.g., `www\pangurus\.com`)
&amp		Used to match & in a string

Table 12.1: RegEx wildcard characters

A list of all contexts and qualifiers can be found in the following Palo Alto Networks Knowledge Base document. It is somewhat outdated but can still serve as good reference material: `https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClOFCA0`.

You can now analyze packets to find identifiable patterns and apply them to signatures of custom applications. You can apply this same knowledge to custom threats!

Creating custom threat signatures

If you need to take a more complex approach to a certain data pattern than allowing or blocking through a simple App-ID-driven security rule, you can also create custom threats that can block or reset a client or server or both, or block the IP of an attacker if a specific pattern is detected in a session.

You can create either a custom vulnerability or custom spyware. Both profiles have the same options but fall into different security profiles and reporting categories.

We will build a custom vulnerability, but the process for creating custom spyware is identical.

In **Objects** | **Custom Objects** | **Vulnerability**, create a custom vulnerability by following these steps:

1. Click **Add.**
2. In the **Configuration** tab, you need to set a threat ID and a descriptive name. All threats are identified by their ID, and a window from `41000` to `45000` and `6800001` to `6900000` is reserved for custom threats (`15000–18000` for custom spyware).

   Let's set an ID of `41000` and give it the name `BlockBrowser`.
3. Set **Severity**. If your vulnerability profile has a specific action other than **default** for the severity, that action will be applied unless you create an exception in the profile. Let's set **high**.
4. For **Direction**, you can set whether this vulnerability should only match if the packet is traveling in a specific direction – from client to server or from server to client – or if it can be detected in both directions. We will set **client2server**.
5. Define a **Default Action**. Set **Reset Client**.
6. **Affected System** is the only unique setting for vulnerabilities that are not also found in spyware; it indicates who is involved with a certain signature. As we're going to capture outgoing browsing sessions, we'll set this to **client**.
7. If there's any CVE, vendor bug ID, or Bugtraq information you'd like to add for completeness, there are fields available to add this information.

The **Configuration** tab should look as in the following screenshot:

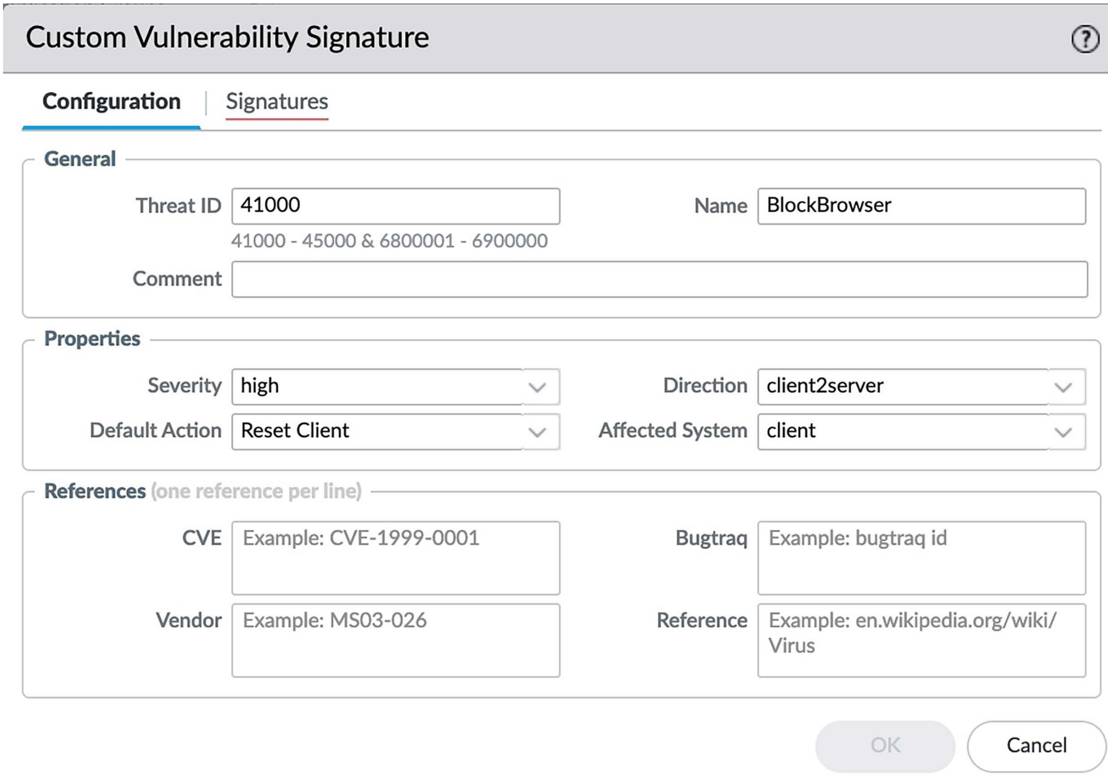


Figure 12.10: Custom Vulnerability Signature

In the **Signatures** tab, we can add the patterns as we did before with custom applications, but there are two signature types:

- **Standard** is the same type of pattern match as the custom application.
- **Combination** adds a timing attribute that lets you define a number of hits over a specified amount of time, and the aggregation criteria (hits from the source, the destination, or the source to the destination are counted). This can help identify brute-force conditions where one or two signature matches in a timeframe could be normal, but five is suspicious. **Combination** can only be applied to predefined vulnerability IDs.

We've already covered how to identify the payload, so let's create a **standard** signature that can block Firefox from being used by a user.

As shown in the following screenshot, if you packet capture a web-browsing session from a regular browser, it will advertise its User-Agent, which is the software used to retrieve the web page. We can use this information in a signature to prevent certain browsers from accessing web pages:

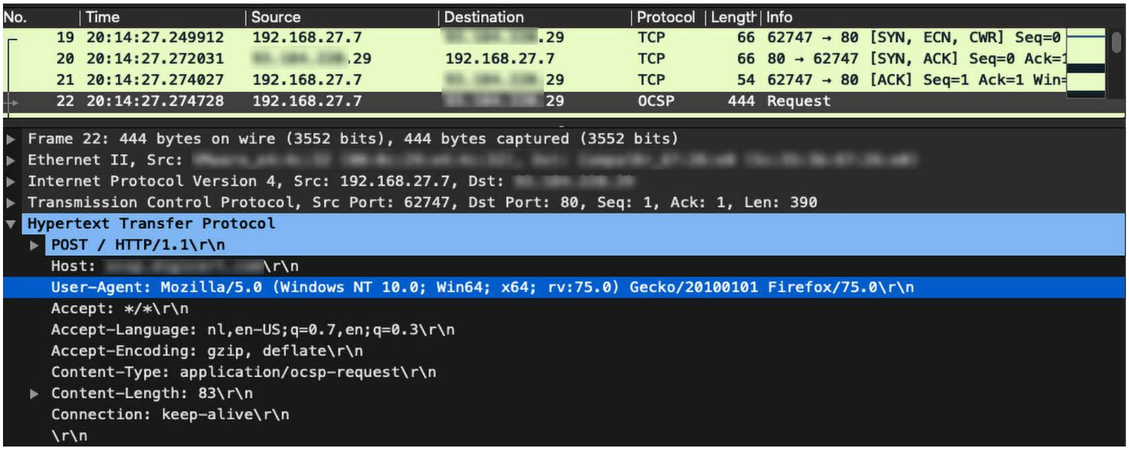


Figure 12.11: A packet capture web session from Firefox

Add the details learned from the packet capture to the custom threat:

1. In the `BlockBrowser` custom threat's **Signatures** tab, click on **Add** and set a name, `Firefox`.
2. Set **Scope** to **Transaction.**
3. Add an `OR` condition:
    - Set **Operator** to `Pattern Match`
    - Set **Context** to `http-req-headers`
    - Set **Pattern** to `Firefox/`
    - Add a qualifier and set it to `http-method` with a value of `POST`
4. Click **OK.**

> If you want to add multiple User-Agents, you can add more `OR` conditions, each matching a different browser type, for example:
>
> 1. Add an OR condition:
>   - Set **Operator** to `Pattern Match`
>   - Set **Context** to `http-req-headers`
>   - Set **Pattern** to `Chrome/`
>   - Add a qualifier and set it to `http-method` with a value of `POST`
> 2. Click **OK** twice.

The **Signatures** tab should look as follows:

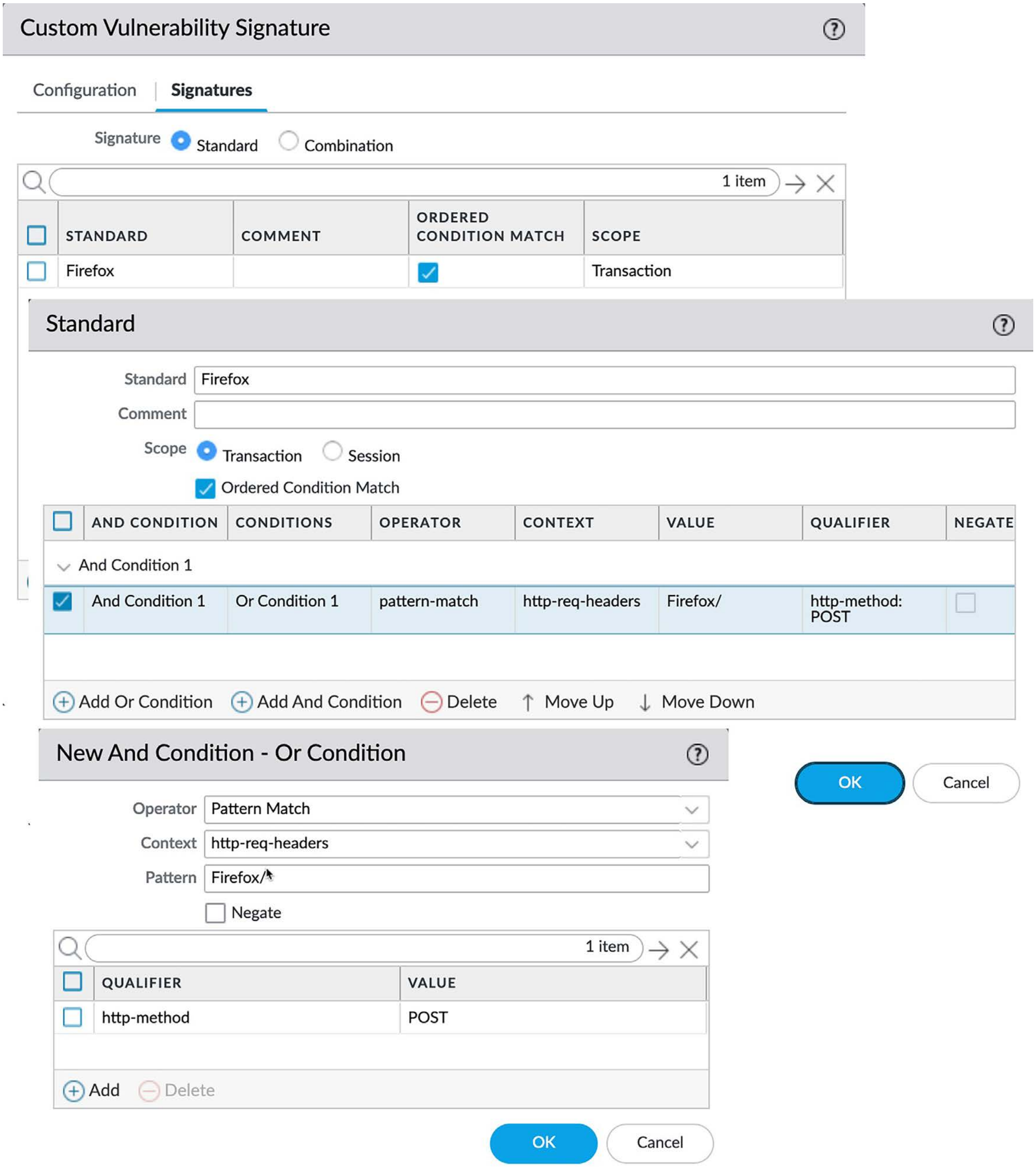


Figure 12.12: Custom Vulnerability Signature

Once this new vulnerability is committed, you will start to see it show up in the threat logs once someone uses a Firefox browser.

Important note

SSL decryption needs to be enabled for patterns to be matched in encrypted payloads or headers.

Pay close attention to the action in the threat prevention rule, as it may differ from the one we set in the custom vulnerability itself. This is because, for high- and critical-severity threats, we usually set an action that replaces all the default actions. If the custom threat action differs from the **Security Profile** settings, add an exception in the **Exceptions** tab.

To add an exception, open the profile where the action needs to be changed:

1. In the **Exceptions** tab, type the threat ID into the search field.
2. Check the **Show All Signatures** box at the bottom.
3. Check the **Enable** box to activate the override for this signature.
4. Make sure the action is set to **default**.

The result will look similar to the following screenshot:

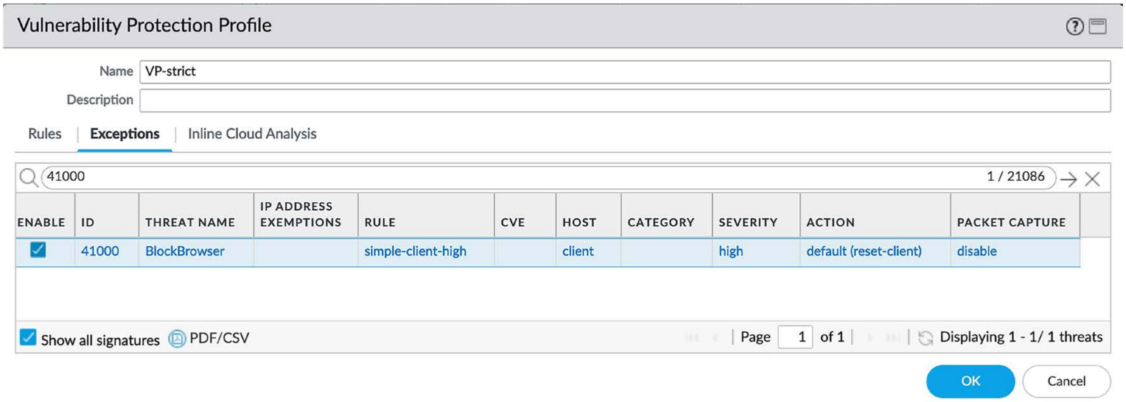


Figure 12.13: Adding a custom vulnerability to the exceptions

You can now use the information you find in a packet capture to create custom applications or custom threats, depending on your needs. In the next section, we're going to protect our network from floods and other low-level attacks.

Implementing zone protection and DoS protection

While layer 7 threats generally revolve around stealing data, blackmailing users through sophisticated phishing, or infecting hosts with complex and expensive zero-day vulnerabilities, protecting the network layer against DoS and low-level packet attacks is equally important. Protecting the system and the network is achieved in three different ways:

- System-wide settings that defend against maliciously crafted packets or attempts at evasion through manipulation
- Zone protection to protect the whole network against an onslaught of packets intended to bring the network to its knees
- DoS protection to more granularly protect resources from being overwhelmed

The system-wide settings are, unfortunately, not all neatly sorted in one place. I'll go over the most important ones.

System protection settings

A good deal of the global session-related settings can be accessed through the **Device** | **Setup** | **Session** tab. In **Session settings**, you can control several nice features such as jumbo frames, IPv6, and accelerated aging.

Packet Buffer Protection

An important setting here that should be enabled is **Packet Buffer Protection**. The firewall has buffers to process traffic while it is coming in and may need to rely on these buffers when CPU usage is high or a session requires extra attention. Attack methods exist that try to exploit such buffers and could cause DoS conditions if they manage to flood the buffer. **Packet Buffer Protection** will keep track of these sessions and discard them if their abuse threatens legitimate sessions.

There are two types of **Packet Buffer Protection** that can be enabled: **Buffer Based Activation** and **Latency Based Activation**. Buffer-based activation reads the usage state of the packet buffer and starts acting when the buffer size reaches a preconfigured threshold. Latency-based activation is only available for hardware firewalls running PAN-OS 10.0 or later and monitors CPU processing latency.

The options for **Buffer Based Activation** are as follows:

- **Monitor Only** can be enabled so the potential impact of enabling **Packet Buffer Protection** can be ascertained before actively implementing a blocking action on abusive sessions.
- **Alert** is the level at which logs will start to be generated, indicating the packet buffer has reached the configured threshold.
- **Activate** is the level of buffer usage where the protection will start monitoring sessions that are heavily taxing the buffers and discard the session if needed.
- **Block Countdown Threshold** is when the **Block Hold Time** value starts to decrease.
- **Block Hold Time** is the amount of time for which a session is allowed to *act abusively* without being blocked immediately. This allows a bit of a buffer in case a legitimate application temporarily misbehaves.

Block Duration is the amount of time for which the blocked IP will be blocked if the behavior lasts longer than the block hold time.

The options for **Latency Based Activation** are as follows:

- **Latency Alert (ms)** is the latency in milliseconds at which logs will start to be generated, indicating CPU processing has reached the configured latency.

- **Latency Activate (ms)** is the level of CPU processing latency where the protection will start applying **RED** (**Random Early Drop**) on incoming packets and will generate an **Activate** log every 10 seconds.
- **Latency Max Tolerate (ms)** is the level of CPU processing latency where **Packet Buffer Protection** will start applying RED with a nearly 100% drop rate.

> The rate at which RED is applied to incoming packets increases the closer latency gets to the **Max Tolerate** level and is calculated as such:
>
> (current latency – **Latency Activate** threshold) / (**Latency Max Tolerate** threshold – **Latency Activate** threshold).
>
> For example, (400-200)/(500-200) = 200/300 = 66%

- **Block Countdown Threshold** is when the **Block Hold Time** value starts to decrease.
- **Block Hold Time** is the amount of time for which a session is allowed to *act abusively* without being blocked immediately. This allows a bit of a buffer in case a legitimate application temporarily misbehaves.
- **Block Duration** is the amount of time the blocked IP will be blocked if the behavior lasts longer than the block hold time.

Enable **Packet Buffer Protection** in each security zone individually.

From the CLI, you can check whether **Packet Buffer Protection** has been engaged:

```
> show session packet-buffer-protection
```

You can also check which zones have been enabled:

```
> show session packet-buffer-protection zones
```

TCP settings

In **TCP Settings**, all protections are enabled by default, but some may need to be disabled (temporarily) to fix an issue. Most commonly, **Asymmetric Path**, which refers to TCP packets arriving out of the window or containing out-of-sync `ACK`, is useful for troubleshooting. Packets dropped by this protection would show up as follows:

```
> show counter global filter category tcp | match out_of
tcp_out_of_sync     0 0 warn tcp  pktproc   can't continue tcp reassembly
because it is out of sync
tcp_drop_out_of_wnd 0 0 warn tcp  resource  out-of-window packets dropped
```

The TCP settings can be verified by running the following command in the CLI:

```
> show running tcp state
```

In the following screenshot, you can see the default values, which, in most cases, should be sufficient:

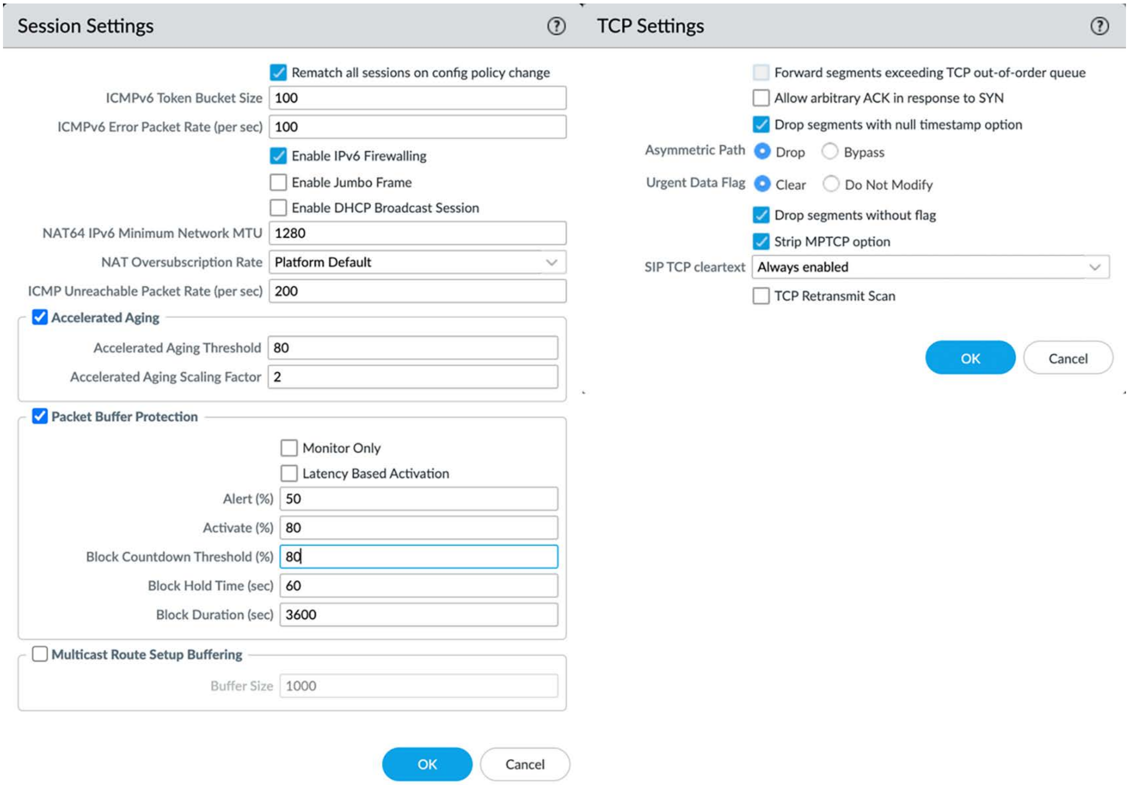


Figure 12.14: The session and TCP settings

The **Session Setup** configuration can only be checked and changed from the CLI:

```
> show session info
    --------------------------------------------------------
    Session setup
      TCP - reject non-SYN first packet:               True
      Hardware session offloading:                     True
      Hardware UDP session offloading:                 True
      IPv6 firewalling:                                True
      Strict TCP/IP checksum:                          True
      Strict TCP RST sequence:                         True
      Reject TCP small initial window:                 False
      Reject TCP SYN with different seq/options:       True
      ICMP Unreachable Packet Rate:                    200 pps
    --------------------------------------------------------
```

TCP - reject non-SYN prevents ACK packets from getting through without first having received a SYN packet to initiate a session.

There's an operational command and a configuration command to change this setting:

```
> set session tcp-reject-non-syn yes|no
# set deviceconfig setting session tcp-reject-non-syn yes|no
```

Strict TCP/IP checksum requires the checksum header to be accurate and unaltered; otherwise, a corrupted checksum will be discarded.

This setting can only be controlled through an operational command:

```
> set session strict-checksum yes|no
```

Strict TCP RST sequence will only accept an RST packet if it has a sequence number that matches the session's flow. RST packets with a mismatching sequence number will be discarded (this could be used to inject reset packets in an attempt to provoke a DoS). This protection can only be controlled through an operational command:

```
> set session tcp-strict-rst yes|no
```

Reject TCP small initial window is disabled by default but lets you set a discard option for SYN packets where the *window size value* in the TCP header is lower than the value you set:

```
> set session tcp-reject-small-initial-window-enable yes|no
> set session tcp-reject-small-initial-window-threshold <0-1024>
```

Reject TCP SYN with different seq/options blocks duplicate SYN packets with different sequence numbers or options:

```
> set session tcp-reject-diff-syn yes|no
```

Now that we've covered the system settings, let's move on to protecting zones.

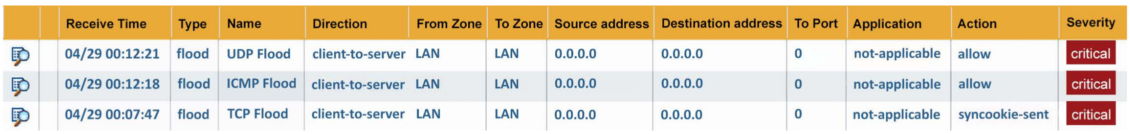
Configuring zone protection

Zone protection does exactly what its name states: protects a zone. This means that each zone needs to be enabled individually and different settings may apply to different zones.

It is important that you have a good understanding of what traffic volumes are to be expected and where the limits of your infrastructure lie for you to be able to set certain flood protections so that they function efficiently. You may want to perform an audit before enabling zone protection.

You can create new zone protection profiles by going to **Network** | **Network Profiles** | **Zone Protection** and following these steps:

1. Click on **Add** and set a descriptive name.
2. In the **Flood Protection** tab, we can enable protection for **UDP**, **ICMP**, **ICMPv6**, and **Other IP**. There are three settings per protocol:
    - **Alarm Rate** is when a log entry is created, alerting the admin that a threshold has been reached. This will be a critical log entry in the threat log, as we can see in *Figure 12.15*.
    - **Activate** is the rate at which **RED** will start randomly discarding packets. This should ideally start happening at a higher rate than what is normal for your network in the appointed zone.
    - **Maximum** is the upper limit of the connections/seconds the system will accept. Anything over this limit will be discarded. The maximum is also used to calculate the progressive rate at which RED discards packets; the closer the connections/seconds get to the limit, the more packets get discarded.
3. **SYN** has one additional setting, called **Action**, where RED can be switched to SYN cookies instead. When SYN cookies are enabled, the firewall does not add SYN queue entries and it discards the SYN packet instead, but it does reply with a SYN/ACK containing a particular sequence number that allows it to reconstruct the original SYN if the client is able to reply with an appropriate ACK to the sequence number. This prevents the SYN queue from getting flooded (as no entries are added).
4. When SYN cookies are used, it is fine to set **Activate** to `0`. When the maximum is reached, all excess SYN packets will still be dropped:



		Receive Time	Type	Name	Direction	From Zone	To Zone	Source address	Destination address	To Port	Application	Action	Severity
		04/29 00:12:21	flood	UDP Flood	client-to-server	LAN	LAN	0.0.0.0	0.0.0.0	0	not-applicable	allow	critical
		04/29 00:12:18	flood	ICMP Flood	client-to-server	LAN	LAN	0.0.0.0	0.0.0.0	0	not-applicable	allow	critical
		04/29 00:07:47	flood	TCP Flood	client-to-server	LAN	LAN	0.0.0.0	0.0.0.0	0	not-applicable	syncookie-sent	critical

Figure 12.15: Flood alert logs

The **Flood Protection** tab should look as in the following screenshot. Make sure to baseline your network before applying aggressive protection.

If no tools or services are available, try setting **Alarm Rate** fairly low and monitor your threat log. Gradually increase the alarm rate until you stop receiving alarms, which should be your peak. At this point, you can set your **Activate** rate for RED and make an educated estimate of where the maximum should be:

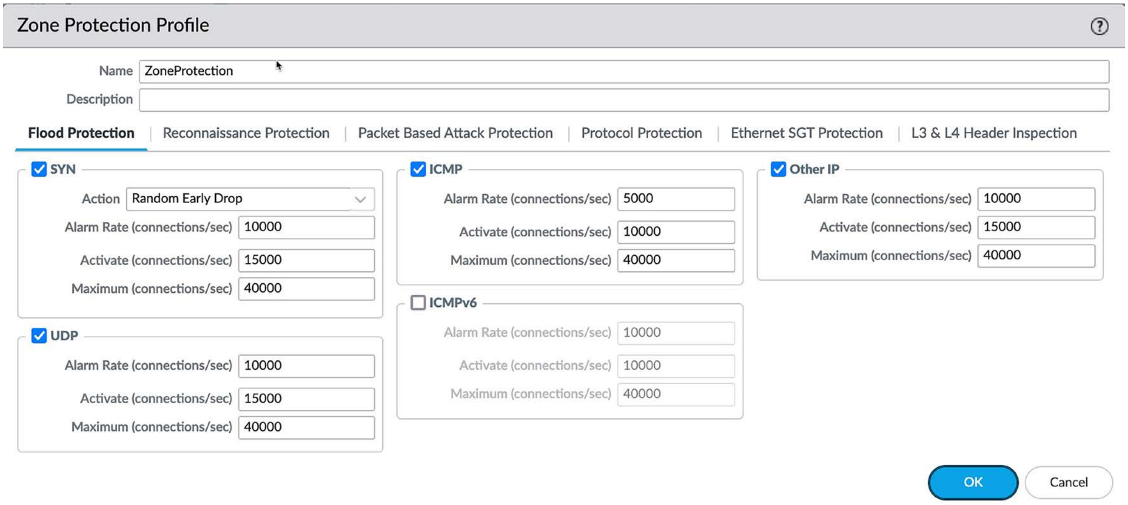


Figure 12.16: Flood Protection tab

In the **Reconnaissance Protection** tab, we can set protection against discovery scans directed at hosts to find out what services are running, or the entire network to map the environment. In Figure 12.17, you can see the four types of scans that can be intercepted:

- **TCP Port Scan** detects TCP connections on many different ports to a single destination from a single source
- **Host Sweep** detects whether a single source is making many connections to many destinations
- **UDP Port Scan** detects UDP connections on many different ports to a single destination from a single source
- **IP Protocol Scan** detects scans against other protocols than TCP/UDP

A source address exclusion can be set in case a known server, such as a PRTG or Nmap server, needs to be able to perform scans for legitimate reasons.

For all scans, the threshold and interval indicate the number of events detected in a certain amount of time, before the action is applied to the source. Actions include **allow**, which disables the scan protection; **alert**, which simply logs detected scans; **block**, which drops new packets that match the type of scan after the threshold was reached; and **block-ip**, which adds the IP to a block list and, depending on whether **Track By** is set to **source** or **source-and-destination**, will block packets from the source or all packets from the source to the destination, regardless of whether the packets are directly associated with the detected scan:

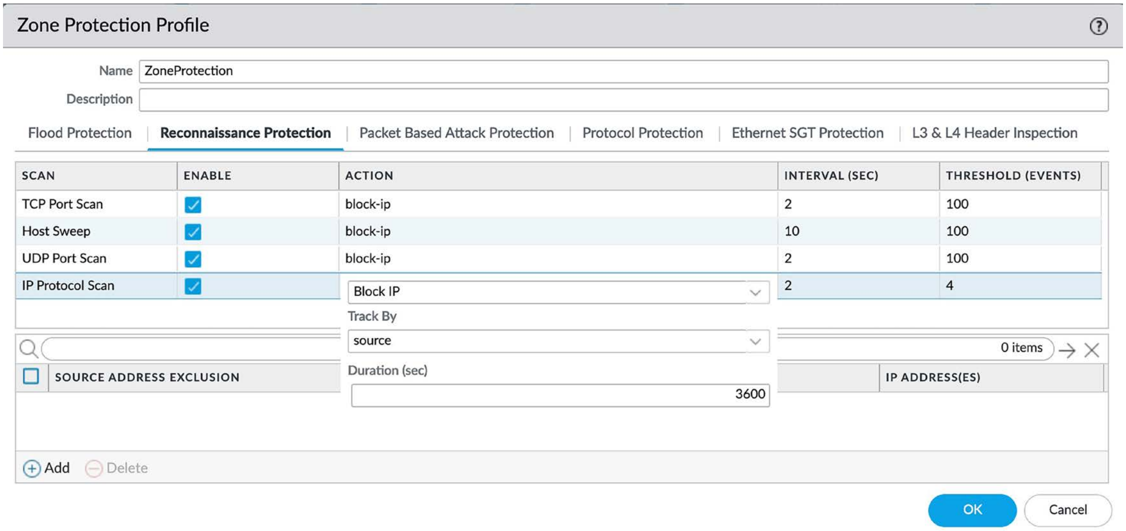


Figure 12.17: Reconnaissance Protection tab

Consider using **alert** as the action for all reconnaissance protection as a crafty attacker could use a spoofed scan to force blocking of legitimate IP addresses.

In the **Packet Based Attack Protection** tab, we find several sub-tabs with a couple of important protection mechanisms. As the following screenshot shows, in the **IP Drop** tab, we will find the following options:

- **Spoofed IP address** will look up the routing table and only accept packets that are ingressing on an interface that has a route associated with the source IP
- **Strict IP Address Check** checks that an IP is not the broadcast IP of a subnet and that the source IP is routable over the source interface
- **Fragmented Traffic** lets you drop fragmented packets (handle this with care as some links may need fragmentation due to **MTU** (**Maximum Transmission Unit**) restrictions)
- **Strict Source Routing** and **Loose Source Routing** are the datagram header options that allow the sender to set the route a packet should take
- **Timestamp** prevents the sender from requesting timestamps from any machine processing the packet
- **Record Route** is an IP header that lets the sender collect the IP from every host processing the packet
- **Security** and **Stream ID**, with IP options `2` and `8`, respectively, can also be blocked
- **Unknown** is the packets that have an unknown class or number
- **Malformed** is the packets with inconsistent combinations of their length, class, and number (based on RFCs `791`, `1108`, `1393`, and `2113`)

Enabling any of these will discard the packet that matches the misbehavior.

The **IP Drop** tab should look similar to the screenshot below:

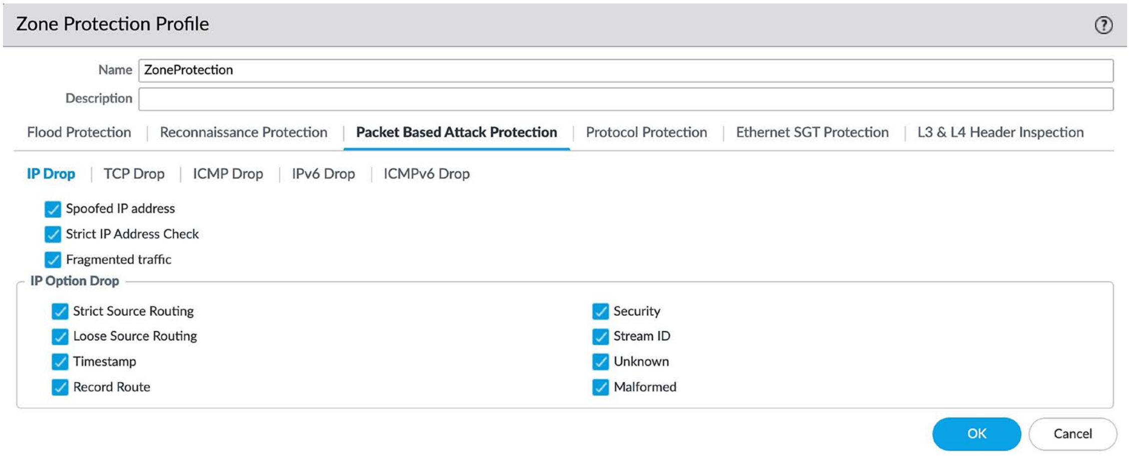


Figure 12.18: IP drop packet-based attack protection

In the **TCP Drop** tab, we can protect against TCP-based malformations or irregularities that could be abused to gain access or exploit systems:

- **Mismatched overlapping TCP segment** blocks packets that are using an incorrect sequence number and could have been injected into a flow.
- **Split Handshake** prevents TCP handshakes that have been fragmented or split over more than three packets.
- **TCP SYN with Data** and **TCP SYNACK with Data** block SYN and SYNACK packets that contain data, since these packets should only be used to establish a handshake and not to transport data.
- **Reject Non-SYN TCP** and **Asymmetric Path** are normally set globally but can be set differently per zone to accommodate some zones needing either of these TCP anomalies without compromising the other zones by changing the global setting.
- The **TCP Timestamp** option should be stripped from the TCP header to prevent timestamp DoS attacks.
- The **TCP Fast Open** option can be stripped. If this check is left disabled (the default), SYN or SYNACK data will be allowed for the purpose of **TCP Fast Open**, even if **TCP SYN with Data** and **TCP SYNACK with Data** are set to blocking.
- **Multipath TCP (MPTCP) Options** can also be left as the global setting or controlled per zone to allow exceptions to the global setting as some zones may need to support multipath.

The **TCP Drop** tab will look similar to the screenshot below:

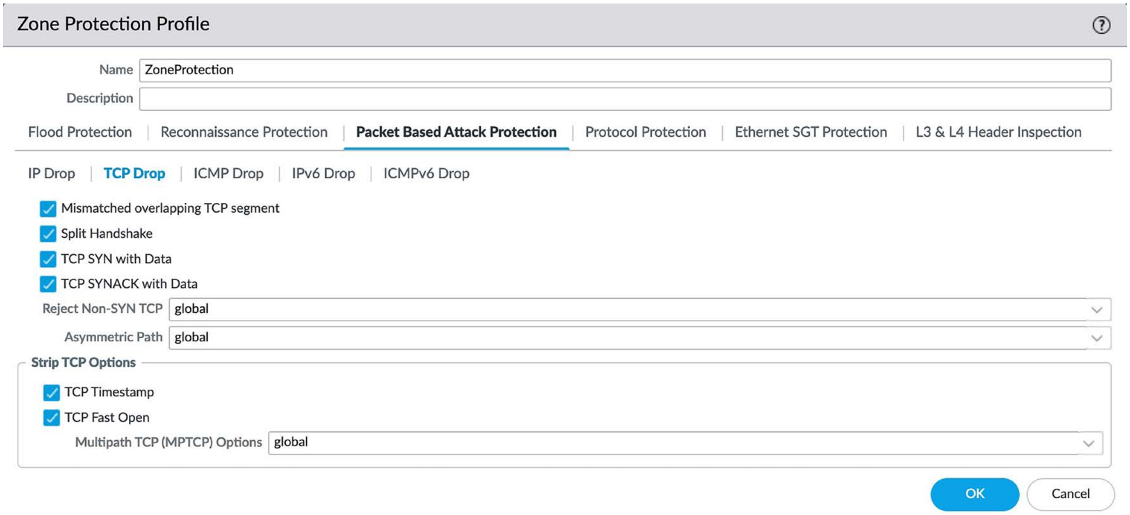


Figure 12.19: TCP drop packet-based attack protection

As you can see in the following screenshot, all the **ICMP** and **ICMPv6** options are disabled by default. Because ICMP is commonly used for troubleshooting, most options may be desirable from a support perspective. ICMP settings can only be set to discard packets, while any options checked in ICMPv6 can be overruled by adding explicit security rules that allow the options:

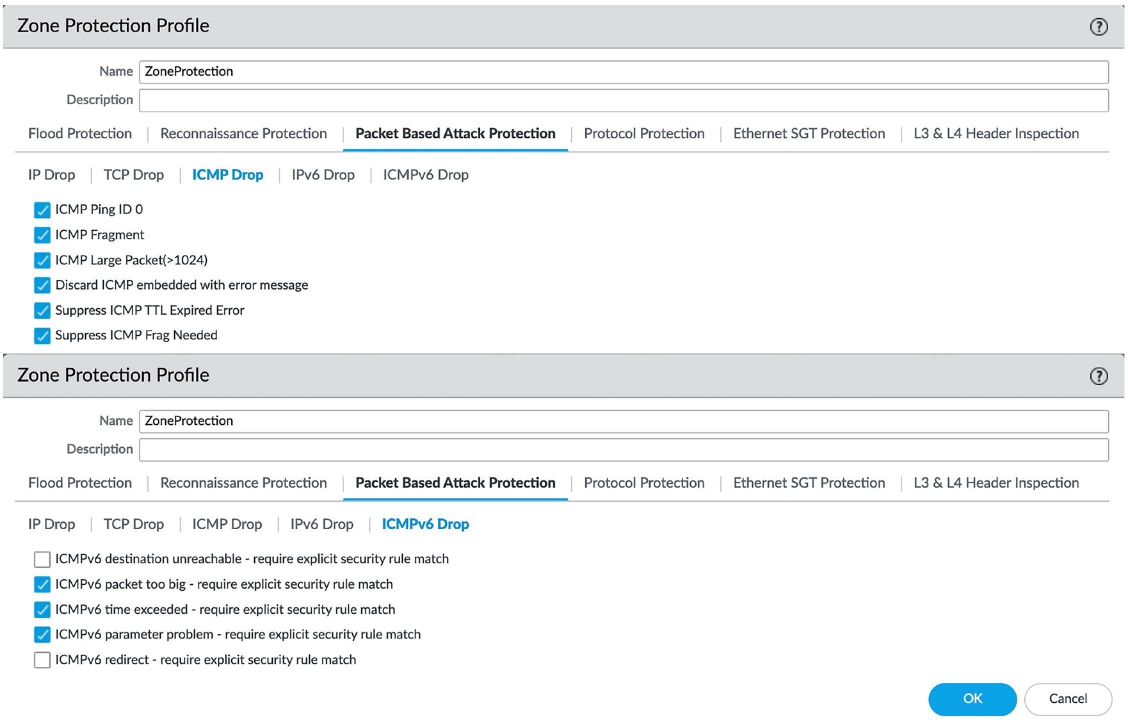


Figure 12.20: Default ICMP and ICMPv6 drop settings

By default, all routing headers, except types 3, 253, and 254, are dropped in **IPv6 Drop**, as you can see in the following screenshot:

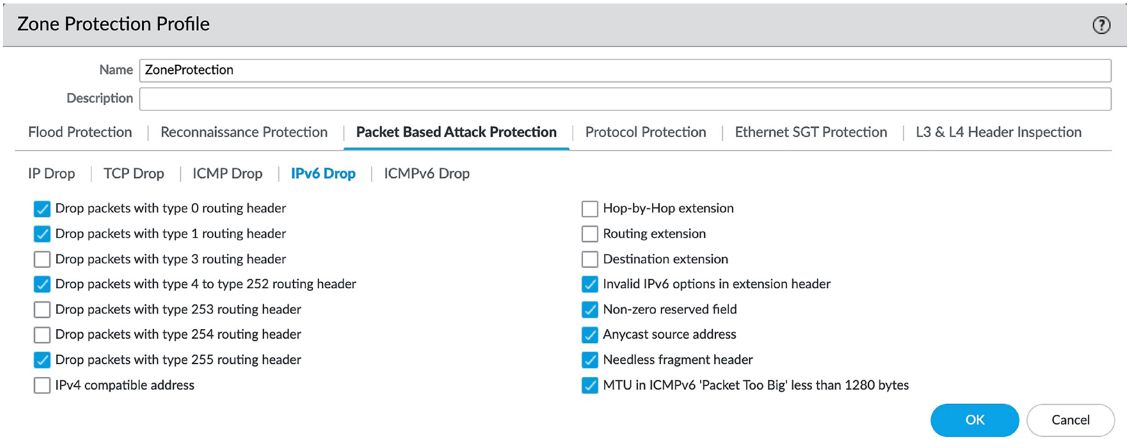


Figure 12.21: Default IPv6 drop options

You can find more information on what each routing type in IPv6 represents here: `https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml`.

In the **Protocol Protection** tab, you can add other protocols outside of IPv4, IPv6, ARP, and VLAN-tagged frames by their hex EtherType value. You can find a list of protocols and their hex EtherType at `http://standards-oui.ieee.org/ethertype/eth.txt`.

As you can see in *Figure 12.22*, this section is fairly straightforward; we can add several different protocols, but we need to choose whether we set this to be an include or exclude list:

- **Exclude List** will drop all the protocols listed. As the following screenshot shows, an EtherType of `0x890d` would be blocked while all the other protocols would be allowed.
- **Include List** allows only the protocols listed in addition to IPv4, IPv6, ARP, and VLAN-tagged frames. All other protocols will be dropped.

In the following screenshot, I've set up a protocol exclude example as I wanted to block the `802.11` management protocol:

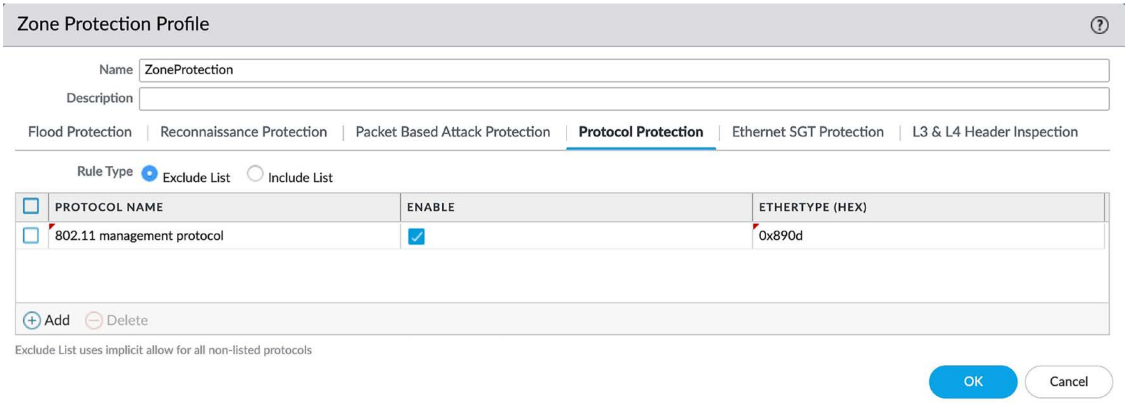


Figure 12.22: Protocol Protection

In **Ethernet SGT Protection**, you can add Cisco TrustSec **Security Group Tags** (**SGTs**). If an incoming packet has an `802.1Q` header that contains an SGT that matches one of the tags in the list, the packet will be dropped.

To enable zone protection, go to **Network** | **Zones** and add a zone protection profile to all the zones by selecting the appropriate one from the dropdown. Make sure to also enable **Packet Buffer Protection**.

Let's take a closer look at both **Packet Buffer Protection** and **L3 & L4 header inspection.**

Packet Buffer Protection and L3 & L4 Header Inspection

Where **Zone Protection** works to protect your firewall from many sources flooding your interface, **Packet Buffer Protection** (**PBP**) works to protect from a single source flooding your firewall packet buffers.

L3 & L4 Header Inspection, a new feature in PAN-OS 11.0, enables the firewall to inspect the layer 3 and layer 4 (ISO) headers of supported IPv4 and IPv6 UDP, TCP, and ICMP headers, and log or block packets that match the custom rules in the zone protection profile.

To be effective, they need to be enabled in the ingress zone. **Enable L3 & L4 Header Inspection** only on the zones where packets matching the custom rules are expected (see the zone protection profile example in *Figure 12.23* at the end of this section). Enable PBP on all zones:

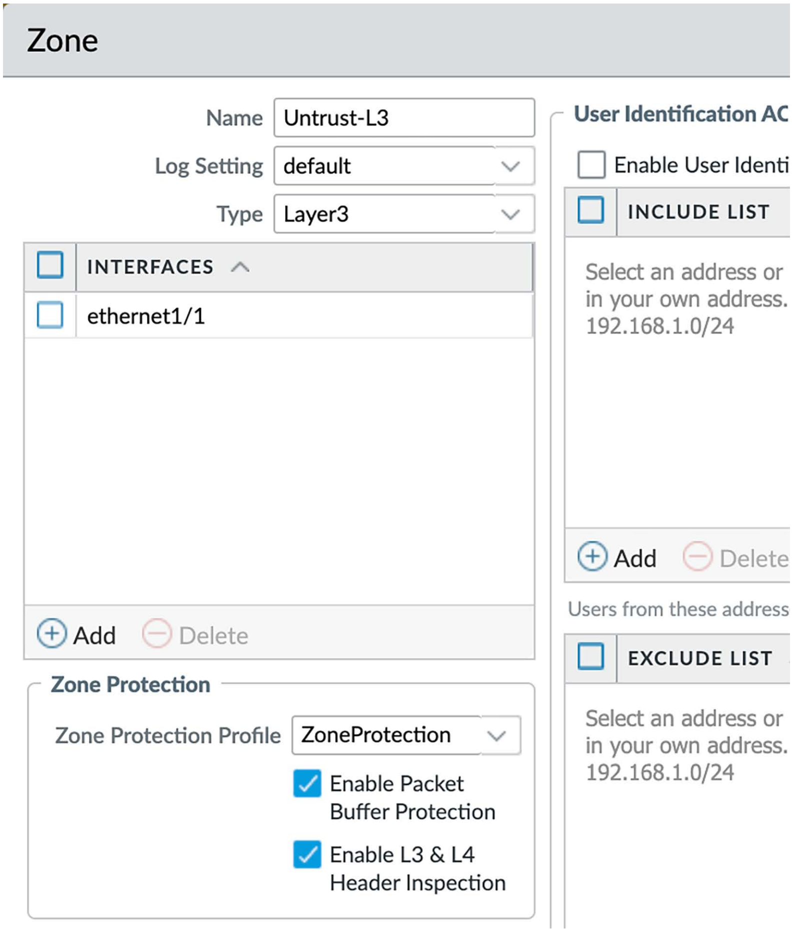


Figure 12.23: Zone protection in zones

Next, in **Device | Setup | Session | Session Settings**, we must enable **L3 & L4 Header Inspection** (this requires a reboot) and **Packet Buffer Protection** (enabled by default but can be disabled).

PBP comes with a few additional configurable options:

- **Monitor Only** can be enabled so PBP only logs potential incidents without interfering with any sessions. This may come in handy to "test the water" before rolling out active protection.
- **Buffer Based Activation** takes baseline measurements of the firewall packet buffer to determine whether there are offending sessions that back up the packet buffer. We looked at the different configurations for buffer-based activation earlier in the chapter.

- **Latency Based Activation** is an alternative to Buffer Based Activation that uses CPU processing latency instead. We looked at the configurations for this earlier in the chapter.

The session settings will look similar to the following screenshot. Decide whether you want to use **Buffer Based Activation** or **Latency Based Activation** (remember, **Latency Based Activation** applies to hardware firewalls on PAN-OS 10.0 or later only):

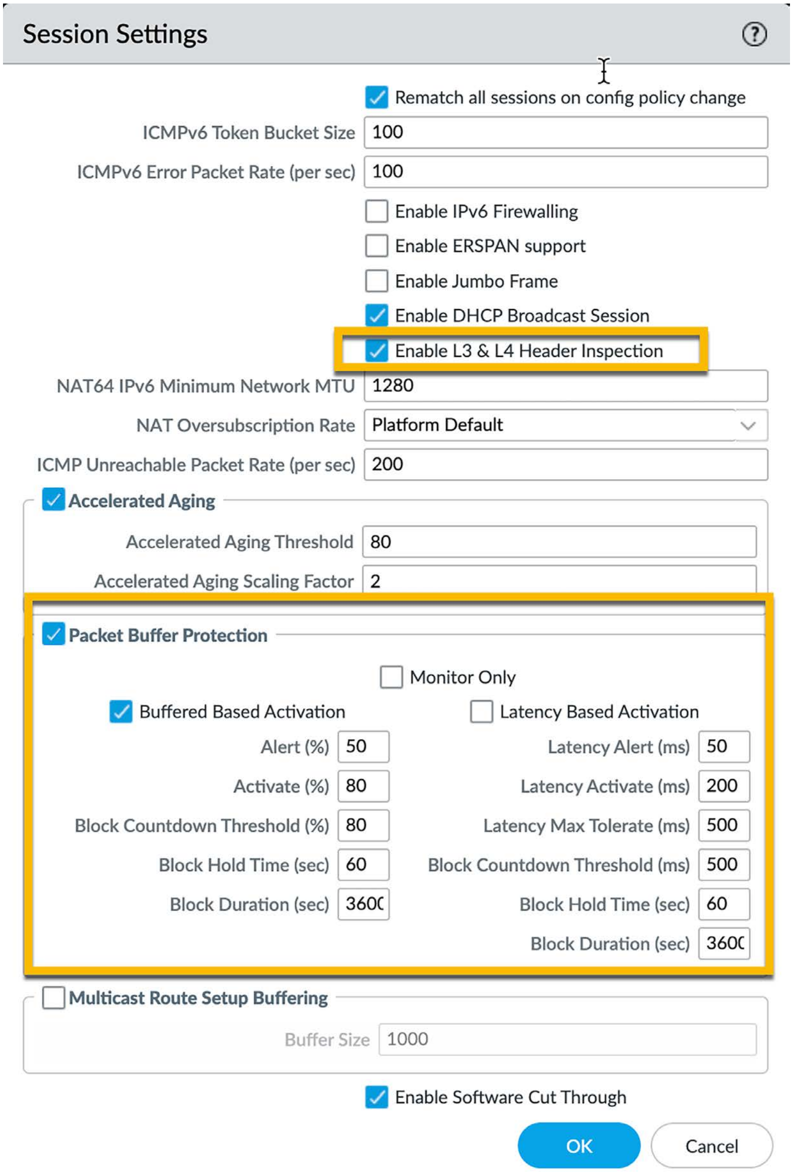


Figure 12.24: System session configuration

Lastly, you can add custom L3 and L4 rules in the zone protection profile to block or log certain vulnerabilities:

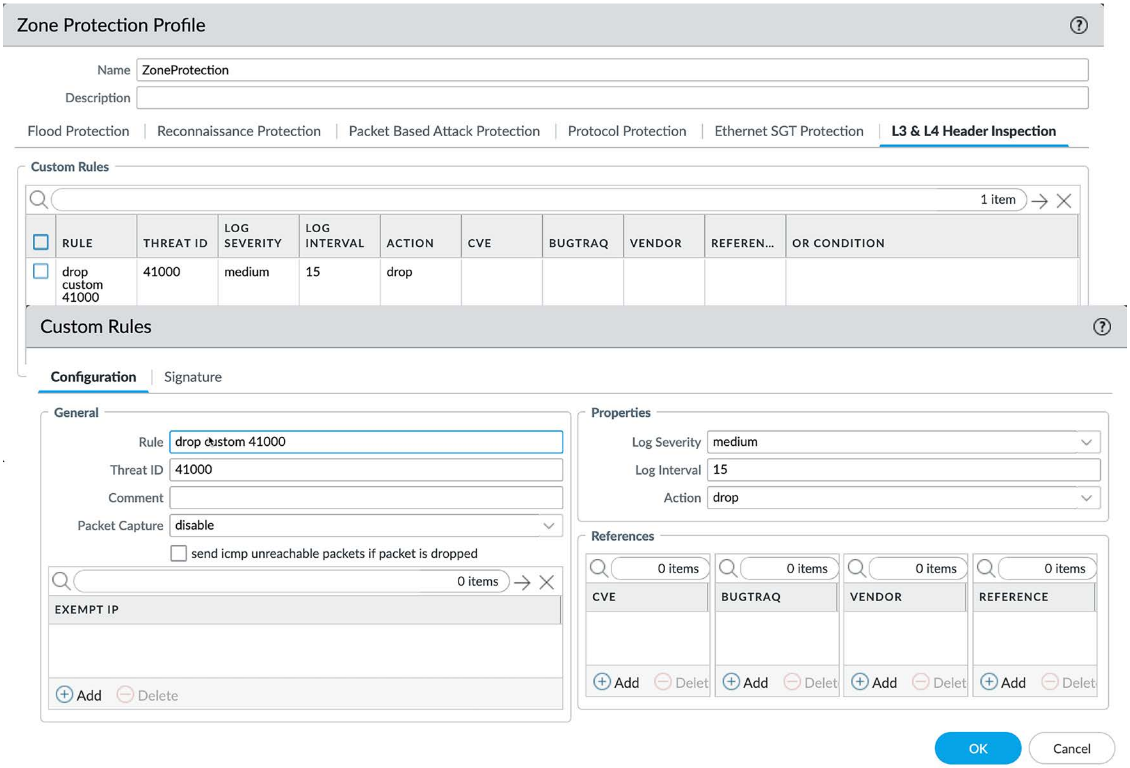


Figure 12.25: L3 & L4 Header Inspection

Now that we've set up protection for our zones, we can add protection for specific resources by setting up DoS profiles and creating a DoS protection policy.

Configuring DoS protection

A DoS protection profile is similar to zone protection but it applies resource limitations at a smaller scale. A server may have limited resources and could be easily flooded by a focused attack leveraging a volume of traffic much lower than what the zone protection profile permits.

New profiles can be created in **Objects** | **Security Profiles** | **DoS Protection**.

As you can see from the following screenshot, the DoS profiles are simpler than those for zone protection. There are two types:

- The **Aggregate** profiles count the total number of connections matching the rule and profile.
- The **Classified** profiles count individual sessions based on the source, the destination, or the source and destination.

There are only two tabs:

- **Flood Protection** contains all the same settings as the **Flood Protection** tab in the **Zone Protection** profile, but someone decided to break it up into smaller topical tabs. The only difference is the addition of **Block Duration**, which will be used in the DoS protection policy.
- **Resources Protection** can be enabled to limit the maximum number of concurrent sessions to a resource.

The **Aggregate** profile will look similar to the following screenshot:

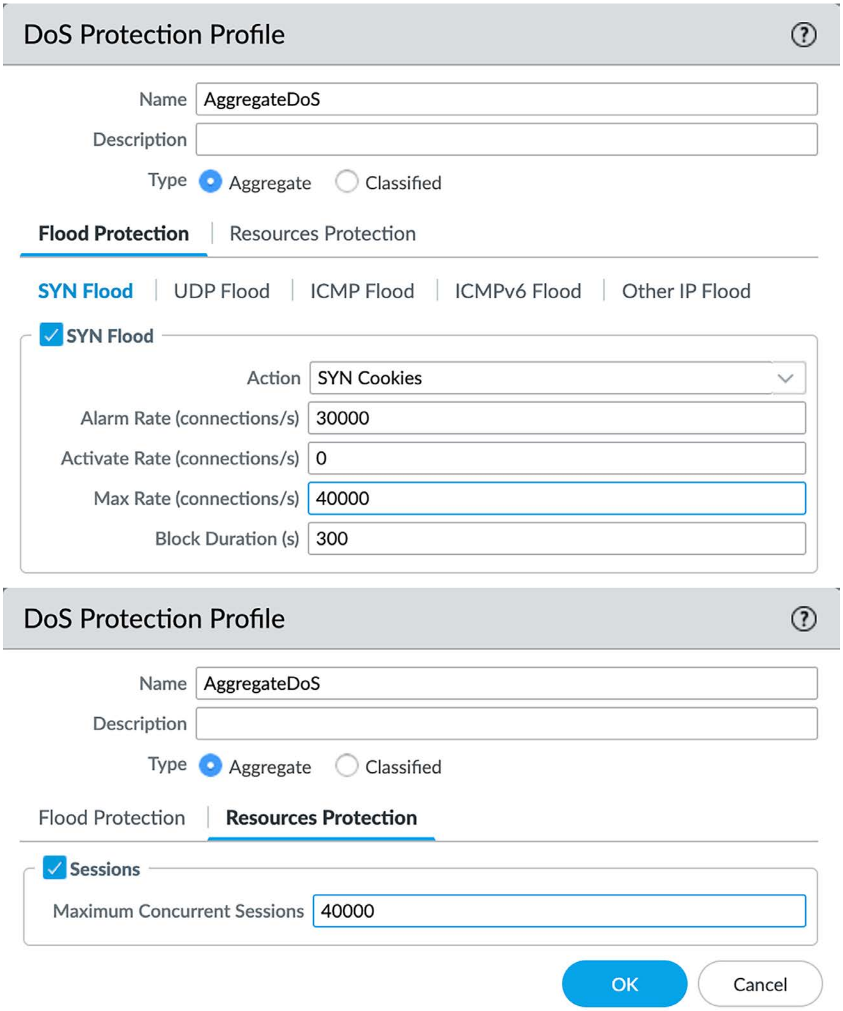


Figure 12.26: Aggregate DoS protection profiles

The **Classified** profile will look similar to the screenshot below:

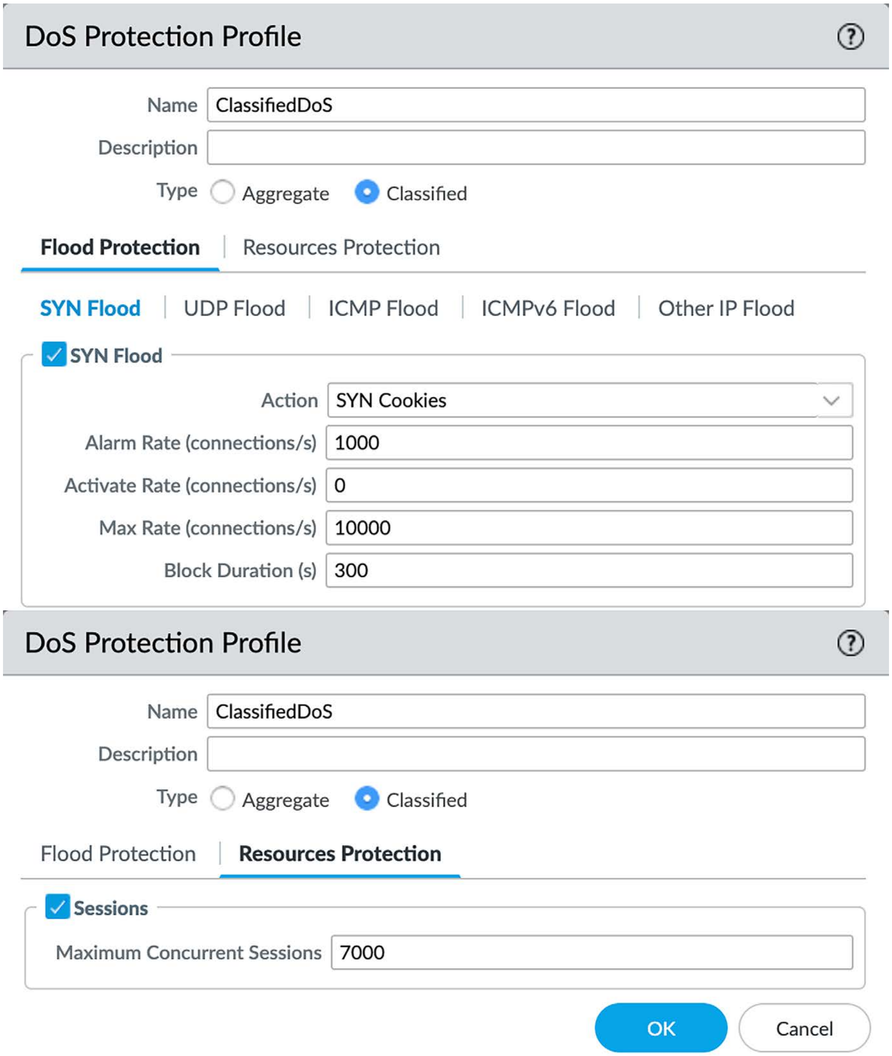


Figure 12.27: Classified DoS protection profiles

To apply these profiles to a resource, create a new rule in **Policies** | **DoS Protection**:

1. Click on **Add** and set a descriptive name.
2. In the **Source** tab, set **Type** to **Zone** or **Interface** and select the appropriate zone or interface. Add a source IP/subnet if needed.
3. In the **Destination** tab, set **Type** to **Zone** or **Interface** and add the destination zone or interface. Set the destination IP address(es) of the resource you are going to protect. Use the public IP address if the connection will come in from the internet and go through the destination NAT.

4. In the **Option/Protection** tab, add the service ports that need to be protected.
5. Then, select one of the following actions:
    - **deny** will block all sessions matching the rule
    - **allow** will allow but not protect all sessions matching the rule
    - **protect** will apply DoS profiles to all sessions matching the rule
    - **deny** and **allow** can be used to create exceptions above a more generic **protect** rule
6. Complete the rule with the following settings:
    - Set a schedule if the rule should only be active at certain moments.
    - Set the appropriate **Log Forwarding** profile if alarm settings need to translate into an email being sent or a syslog sent out to SIEM. If you created a `default` log forwarding profile, it will be added automatically.
    - Select the appropriate **Aggregate** profile.
    - If more granular protection is needed, check **Classified** and select the classified profile.
7. Then, set the address classification as **source-ip-only**, **destination-ip-only**, or **src-dest-ip-both**.

Important note

Address classification takes up resources to keep track of sessions. You should be careful or defer to using **source-ip-only** or **destination-ip-only** for internet-facing protection rules.

Your rule should look similar to the following:

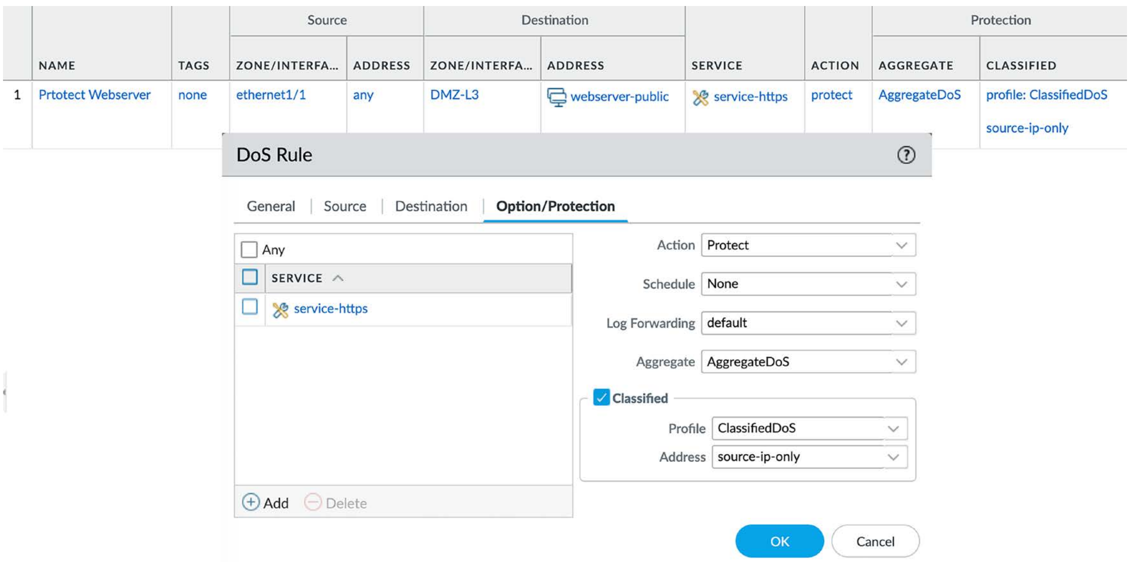


Figure 12.28: A DoS protection rule

With this information, you are now able to protect your network and individual servers from getting flooded. Remember: there's only so much a firewall can do. If the ISP uplink is physically flooded, only alternative paths can make resources available to the outside. The firewall's job is to contain the attack to one zone while all other zones can continue working.

Summary

In this chapter, you learned how to set up site-to-site VPN tunnels and a client-to-site VPN with GlobalProtect. You can now not only provide connectivity but also scan the client machine for compliance and know how to control the user experience. You've also learned how to create custom applications and custom threats that will allow you to identify packets unique to your environment and take affirmative action, and you've learned how to set up zone and DoS protection to defend against all kinds of packet-based attacks.

If you're preparing for the PCNSE, remember that QoS rules are applied on the egress interface. You also need to remember how the classes apply to different profiles on different interfaces, as well as the implications of using an app override and what the benefits are of a custom application or custom threat.

In the next chapter, we will be getting hands-on with some basic troubleshooting. We will learn about session details and how to interpret what is happening in a session.

13

Troubleshooting Common Session Issues

In this chapter, we will learn how to read a session output and how to troubleshoot basic session issues. We will learn how to use the tools available in the web interface to find problems and test policies. We will go over the steps to collect all the information we need to find out why a session may not be working as expected or predict how a new rule will react to certain sessions. We will also look at a user tool called **Maintenance Mode** or the **Maintenance Recovery Tool** (**MRT**), which allows for some very powerful system-level interactions with the firewall.

In this chapter, we're going to cover the following main topics:

- Using the tools in the web interface
- Interpreting session details
- Using the troubleshooting tool
- Using Maintenance Mode to resolve and recover from system issues

At the end of this chapter, you'll be able to perform basic troubleshooting. You will be able to quickly determine which logs you may need for a specific situation, collect packets to review what may be going wrong, and interpret sessions on the firewall.

Technical requirements

Since we're going to be doing some troubleshooting, having a lab available so that you can reproduce some of the steps explained here will greatly help you understand the materials we will cover.

Using the tools in the web interface

Knowing your way around the web interface is a great start if you need to troubleshoot an issue. There are plenty of spaces where valuable information is stored, and knowing just where to look can be the difference between quickly checking and fixing an issue versus spending hours trying to figure out why something isn't working.

As we saw in *Chapter 10*, *Logging and Reporting*, the **Monitor** tab is such a place where knowing where to look can make all the difference. Logs are maintained for just about any event, from sessions passing through or being blocked by the firewall or a security profile, to things happening on the firewall itself. In most cases, the log files will be the first place to look if something unexpected has happened.

Log files

There are many different log databases that collect specific information, which can be found under **Monitor** | **Logs**. Knowing where to look is essential if you want to quickly find information relating to the issue you are investigating:

- **Traffic** holds all the logs related to sessions. This includes the source and destination IP, the port, the zones and users, the application (or lack thereof), bytes, packets sent and received, and the reason for an action applied by a security policy and session end. You can enable a column to indicate whether a session was decrypted or intercepted for captive portal authentication. For each session start or session end log action, an entry is created as determined by the matching security rule log settings.
- **Threat** also logs the log source and destination IP, the port, the zones and users, and the application, but these logs are created as a result of a detected vulnerability or malware. A log will contain the name of the threat and the direction in which it was detected – **client-to-server** (**c2s**) or **server-to-client** (**s2c**). The action listed is what the content engine performed in response to detecting a threat, so it may not correlate with the traffic log; a traffic log may indicate that a session was allowed because it hit a security rule that permitted the connection, but the threat response may have been to send an RST packet or simply create an alert log. In the case of an RST package being sent, the traffic log end reason would read `threat`.

  If packet captures were enabled in the security profiles, any threats that triggered a packet capture will have a little green arrow associated with the log entry, which can be clicked on to download the packet capture.

- **URL Filtering** holds a log of all the URL filtering profile actions except `allow`, which does not generate a log entry. These logs contain the basic source and destination information and the URL and URL category accessed. Actions taken in **URL filtering** will not reflect at all in the traffic log, as the TCP session will simply have been allowed, but the content engine may have returned a `block` or `continue` page.
- **WildFire Submissions** contains a log entry for every file that was intercepted and forwarded to WildFire. The log will contain all the basic source and destination information, as well as a verdict. Grayware and benign verdicts must be enabled in **System** | **Setup** | **WildFire** if you want to keep track of all the files, or else these two verdicts will not be reported. It may take a while for the WildFire log to appear after a file is uploaded, as the log is written when the verdict is learned. The full report can be accessed on the WildFire portal by clicking on the **Detailed log view** icon (the little page with a magnifying glass at the left side of each log entry; see *Figure 13.1*) and clicking **WildFire Analysis Report**.

- **Data Filtering** contains logs for any events that were triggered where keywords were detected in a data filtering profile. The log will contain the basic source and destination information, the filename, and/or the URL accessed.
- **HIP Match** maintains a log for all HIP profiles matched to users logging in through GlobalProtect.
- **GlobalProtect** keeps a record of every user logging in or retrieving a configuration, and which portal or gateway they connected to. A neat feature to help troubleshoot can be seen by applying the `eventid eq gateway-tunnel-latency` filter, which provides pre- and post-tunnel latency for all connected clients. Another useful filter is `tunnel_type eq SSLVPN` to see who is using SSL instead of IPSec.

  Ensure the **Description** column is enabled.

- **IP-Tag** keeps a log each time a tag is assigned to a particular IP address.
- **User-ID** keeps track of all the user-to-IP mappings and the source from which the information was learned.
- **Decryption** contains detailed information regarding sessions that hit a decryption policy. In the case that decrypted sessions fail to connect due to a certificate issue, unsupported cipher suite, or other issues, a log entry will be written here to help troubleshoot any issues.
- **Tunnel Inspection** writes a log for each inspected tunnel, the start and end time, the application used to tunnel, the session and tunnel ID, and the security and tunnel inspection rules matched for the session.
- **Configuration** contains all the configuration changes and information about the administrator who made the change, as well as the time and date and the source address that the admin was connecting from.
- **System** contains all the logs relating to events happening at the system level: any dynamic updates that were downloaded and installed, IPSec tunnels that were established or torn down, commit jobs, admin authentications, daemons reporting on commit outcome, syslog events, satellite connection events, high-availability events, hardware alarms, DoS alarms, and LACP and LLDP events.
- **Alarms** contains specific logs relating to alarms. Default alarms include fan speed/fan tray, temperature issues, and power supply issues. Additional alarms can be configured in **Device** | **Log Settings**. If you enable an alarm, set the log quotas higher as log pruning happens at around 95% capacity.
- **Authentication** contains logs for users authenticating against an authentication (captive portal) rule in **Policies** | **Authentication**.
- **Unified** displays the **Traffic**, **Threat URL Filtering**, **WildFire**, and **Data Filtering** logs all in the same view. When proper filtering is applied, this log view supplies a great single-pane overview.

There is a little magnifying glass to the far left of each log entry that opens a detailed log view:

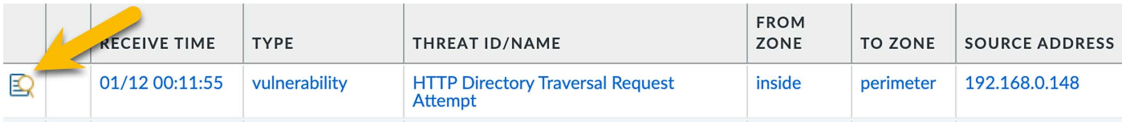


Figure 13.1: Log entry details

This view opens a treasure trove of information, as you can see in the following screenshot:

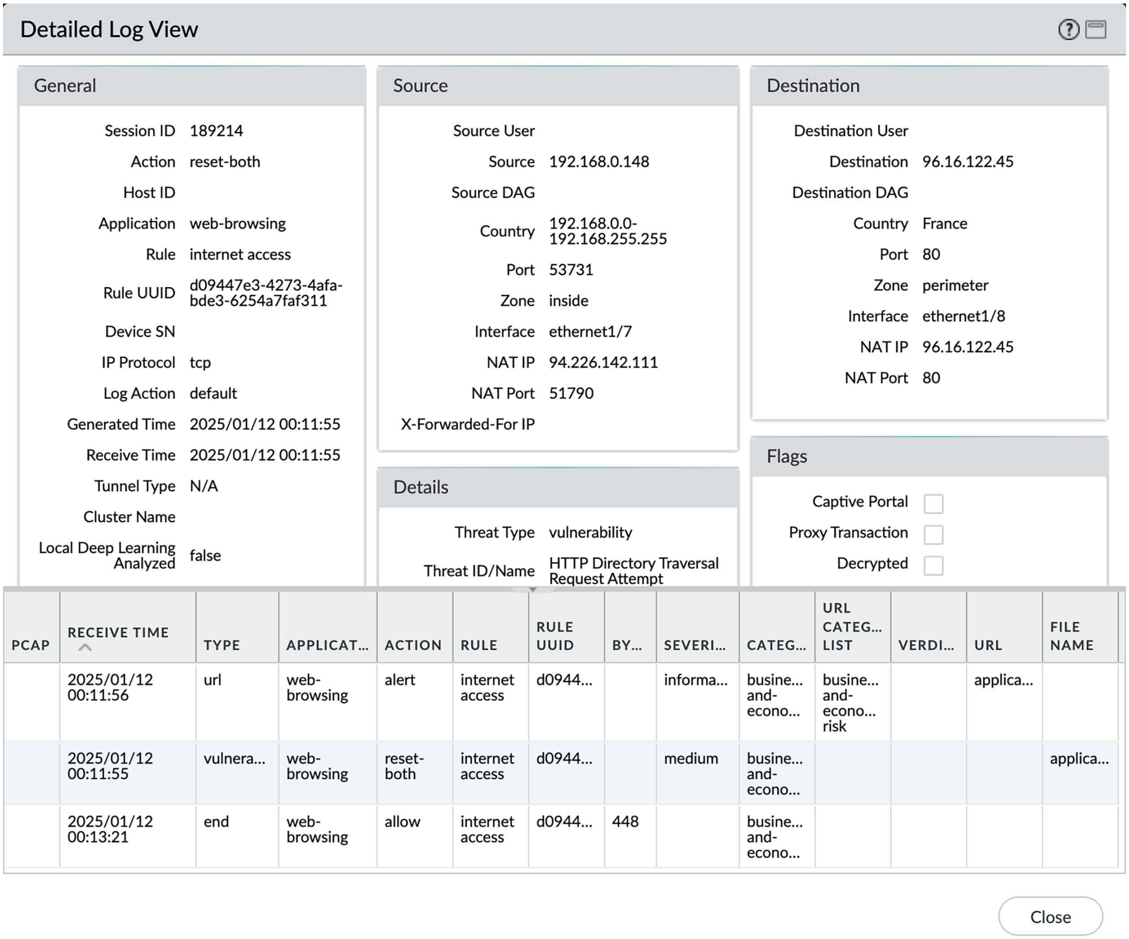


Figure 13.2: Detailed log view

At the bottom of **Detailed Log View**, there is a clickable list of related log entries, which allows you to review **Traffic** (**start or end**), **Vulnerability**, **URL**, and other related log types. Each individual related log can be opened from this view so you can review all the actions taken by the related process or decoder and see what information was collected in that stage.

Any associated packet captures are listed here as well. If, for example, a vulnerability is detected that matches a security profile that has **packet captures** enabled, the packet capture will appear next to the **Vulnerability** log.

In Panorama, logs are displayed depending on the selected device group (see *Figure 13.3*):

- The **System** log will only show if the device group is set to **Shared**/**All**. In older Panorama PAN-OS versions, the configuration log may also only show up if the device group is set to **Shared**.
- When switching the **Device Group** dropdown to a specific device group (e.g., **HQ Firewalls**), the **Traffic**, **Threat**, and other logs will be filtered for only the selected device group while the **Shared**/**All** level will display all logs for all device groups.

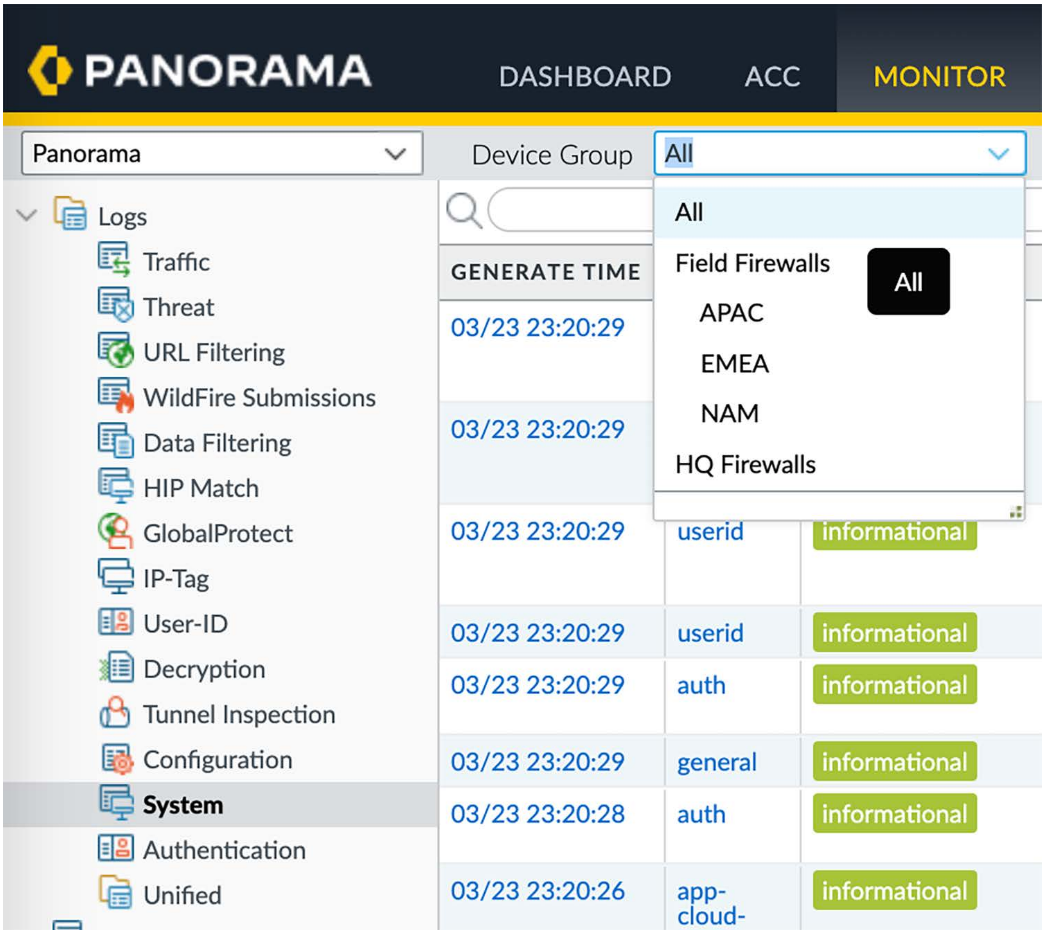


Figure 13.3: Panorama System logs

Logs provide an abundance of information, but for some troubleshooting sessions, more information will need to be collected and a deeper look at the actual packets will be required to find out what is going on. In the next section, we will learn how to capture packets.

Packet captures

The real fun begins in **Monitor** | **Packet Capture**, as we can set up packet capturing for sessions crossing or bouncing off the data plane. Packet captures will intercept the actual packets flowing from the client to the server, and the other way around, and write them to a convenient `.pcap` file, which you can load into a tool such as Wireshark to investigate everything that is happening at the packet level.

There are several areas that can be configured.

Configuring filters

In the upper-left quadrant, you can configure filters by clicking on **Manage Filters** to add up to four filter rules. Each filter rule has several fields that can be used to narrow down the scope of the packet capture:

- **ID**: This is required and must be `1`, `2`, `3`, or `4` with no duplicate IDs. There can only be four filters set at a time.
- **Ingress Interface**: This can be set to only capture whether a matching packet is received on a specific interface.
- **Source**: This is the source IP of the packets being captured.
- **Destination**: This is the destination IP of the packets.
- **Src Port**: This is the source port of the packet that needs to be captured.
- **Dst Port**: This is the destination port to filter for.
- **Proto**: This is the IP protocol – common protocols are `1` for ICMP, `6` for TCP, and `17` for UDP. There's a handy list on the **Internet Assigned Numbers Authority** (**IANA**) website at `https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml`.
- **Non-IP**: This can be set to **Exclude** so that only IP protocol packets will be captured, **Include** to capture both IP and non-IP protocols, or **Only** to exclusively filter non-IP protocols. Non-IP protocols include, for example, NetBEUI, AppleTalk, IPX, and so on.
- **IPv6**: This must be checked to include IPv6 packets that match the filters.

To activate the filters, **Filtering** needs to be toggled to the **ON** position, as in the following screenshot:

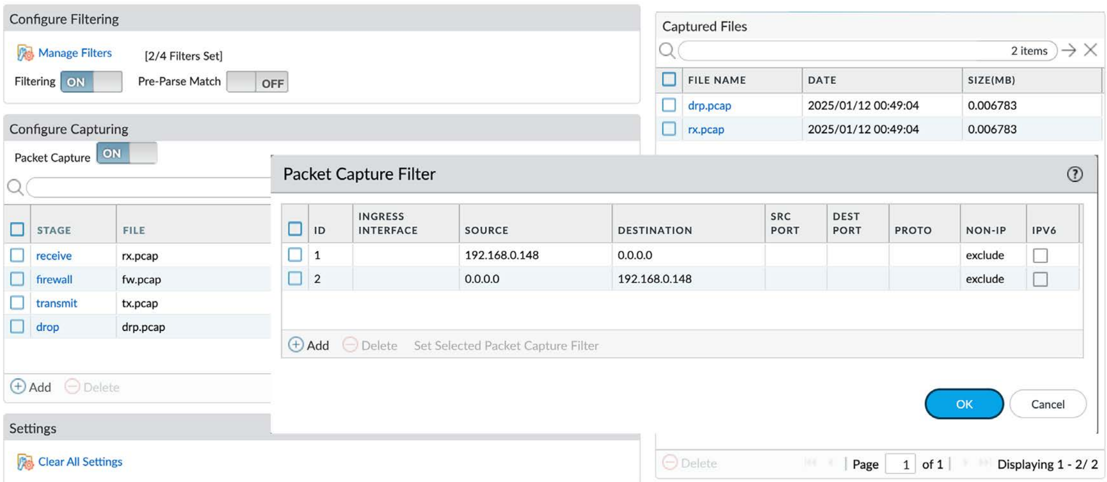


Figure 13.4: Packet captures

The filters are session-aware, which means if you set a filter for one direction of traffic, return packets will also be captured. It is good practice, however, to also include a returning traffic filter in case the packets do not match the session (for example, if the sequence number is somehow completely wrong or the ports have changed somehow). Remember, when setting a filter for returning packets, the destination IP may be the NAT source of the outbound packet, and the original destination port will be the source port.

Any field not filled in will count as a wildcard for that filter value.

Pre-Parse Match is an (advanced troubleshooting) toggle that captures packets before they reach the filtering stage. Some packets may not reach the filtering stage due to them being discarded beforehand. This could be due to a failed routing lookup for the packet. Enabling **Pre-Parse Match** will capture all packets coming into the firewall, essentially bypassing the set filters, so *proceed with caution*. Enabling **Pre-Parse Match** could lead to a negative impact on the system due to extremely high load (i.e., do not enable this feature unless instructed to do so, or in a lab environment).

Configuring capturing

In **Configure Capturing**, four stages can be designated to capture packets:

- Receive captured packets as they are received on the data plane processor
- Transmit captured packets as they leave the data plane processor
- Firewall captured packets while they are being matched to a session in the firewall process
- Drop captured packets as they are being discarded by a policy action or an error

Each stage can be set individually, and it is not necessary to set all stages. Each stage needs to have a unique filename set so that it can write to its own file. Each stage can be limited to how many bytes or packets can be captured; the capture will stop for each stage once the limit is reached.

The maximum size of a single packet capture file is 200 MB. Once that size is reached, the file is renamed with a `.1` extension, and a fresh file is started. Once the new file reaches 200 MB, the old `.1` file is purged, the new file is renamed to have the `.1` extension, and a fresh file is generated to continue the capture.

Once the capture stages have been set, you can enable capturing by setting the **Capturing** toggle to **ON**.

If you then hit the **Refresh** button in the top-right corner, files will start appearing and increasing in size once matching packets are received.

A couple of important considerations should be taken into account when capturing packets: **sessions are marked by the filter** and **offloaded sessions can't be captured.**

Sessions are marked by the filter

The system knows which packets to capture and write to the file by the filters marking sessions to be captured by the processor when the packets reach the designated capture stage during processing. These markings are added to the session when it is created after the filter is made active, so when a packet capture is started, sessions that existed before the filter was activated will not be included in the capture.

Existing sessions can be added to the marked sessions manually by using the following command:

```
reaper@pa-440> debug dataplane packet-diag set filter-marked-session id
<session ID>
```

All the marked sessions can be reviewed using the following command:

```
reaper@pa-440> debug dataplane packet-diag show filter-marked-session
```

If you are done capturing but need to start another capture for a different set of filters, previously marked sessions may inadvertently be captured as they are still marked. Before setting new filters and configuring capture stages, you can delete markings from existing sessions with the following commands:

```
reaper@pa-440> debug dataplane packet-diag clear filter-marked-session id
<session ID>
reaper@pa-440> debug dataplane packet-diag clear filter-marked-session all
```

Offloaded sessions can't be captured

On platforms that have hardware offloading (`pa-3000`, `pa-3200`, `pa-5000`, `pa-5200`, and `PA-7000`), packets will be put into a fast path once processing has completed, which bypasses data plane processing and puts the packets directly onto the networking chip. This will prevent further capturing, as the captures happen on the data plane processors rather than the physical interfaces. If a session needs to be captured that is being offloaded, offloading can be disabled; this could cause additional load on the data plane CPUs, so do not disable offloading when the load is high.

You can check whether offloading is enabled with the following commands (the default is `True`):

```
reaper@pa-3220> show session info | match offload
          Hardware session offloading:                       True
          Hardware UDP session offloading:                   True
```

To disable offloading, issue the following command:

```
reaper@pa-3220> set session offload no
```

Capturing packets on the management interface

Packet capture on the management interface can only be performed from the CLI using the `tcpdump` command. Keep this in mind if you want to inspect sessions between the management interface and, for example, an LDAP server. The capture output file can be read and exported from the CLI.

To start a capture on the management interface, use the `tcpdump` command and add parameters as needed. To end the capture, press *Ctrl+C*.

Setting `snaplen 0` ensures full packets are captured; without setting this option, the capture size per frame may be limited to 96 bytes in older PAN-OS versions.

The filters that can be added are similar to the ones used by `tcpdump` in a Linux system. Some examples are `"src 192.168.27.2"` or `"net 192.168.27.0/24 and not port 22"`. Another example is:

```
reaper@pa-440> tcpdump snaplen 0 filter "host 192.168.27.130 and not port 22"
Press Ctrl-C to stop capturing
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535
bytes
30 packets captured
30 packets received by filter
0 packets dropped by kernel
```

To read the capture output from the CLI, use the following command:

```
reaper@pa-440> view-pcap mgmt-pcap mgmt.pcap
```

You can add several options that will influence the way the output packet capture is displayed. By default, the destination port and IP addresses will be resolved to a friendly name, but this may be undesirable for troubleshooting, so you can disable these options:

- `no-dns-lookup yes`: This will disable DNS lookups for source and destination IPs
- `no-port-lookup`: This will display the destination port as a number rather than a friendly name
- `verbose++ yes`: This adds extra verbosity to the output

Thus, the full command could look like this:

```
reaper@pa-440> view-pcap no-dns-lookup yes no-port-lookup yes verbose++ yes
mgmt-pcap mgmt.pcap
```

To export the file, use either TFTP or SCP to `localhost`:

```
reaper@pa-440> tftp export mgmt-pcap from mgmt.pcap to 192.168.27.7
reaper@pa-440> scp export mgmt-pcap from mgmt.pcap to reaper@192.168.27.7:/
pcaps/
```

Or it can be downloaded from the **Device | Support** tab:

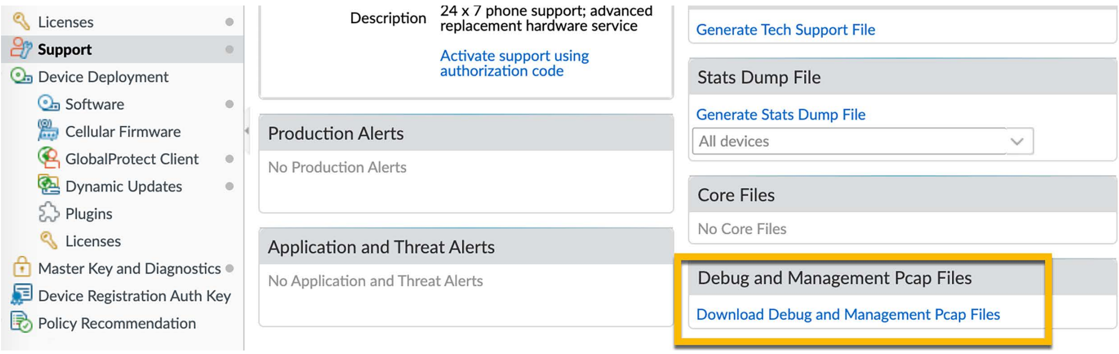


Figure 13.5: Exporting management packet capture files

In addition to using log files and packet captures to review the information that you know, botnet reports collect behavioral information that can help find suspicious hosts in the network.

Botnet reports

In **Monitor** | **Botnet**, there is a log consolidation tool that will keep track of sessions that, when encountered by themselves, are not suspicious at all, but when seen combined with other events, may indicate something is going on that may need some extra attention.

As you can see from the following screenshot, you can edit the configuration for the triggers for botnet reporting by clicking on the **Configuration** link below the calendar.

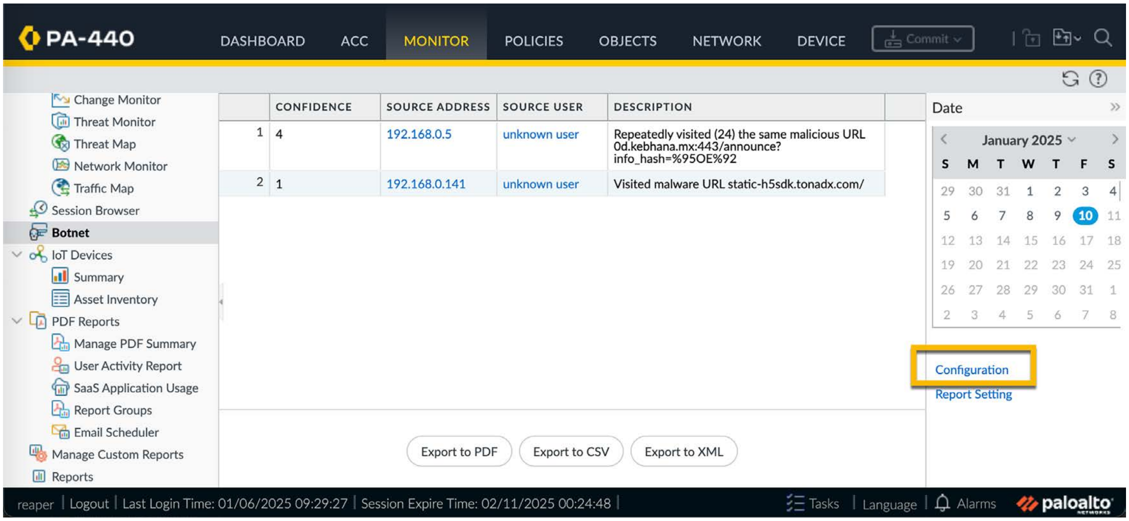


Figure 13.6: Botnet report

Detection is based on the repetition of certain events within a specified timeframe. You can tweak how many occurrences need to happen before something is reported in the botnet report:

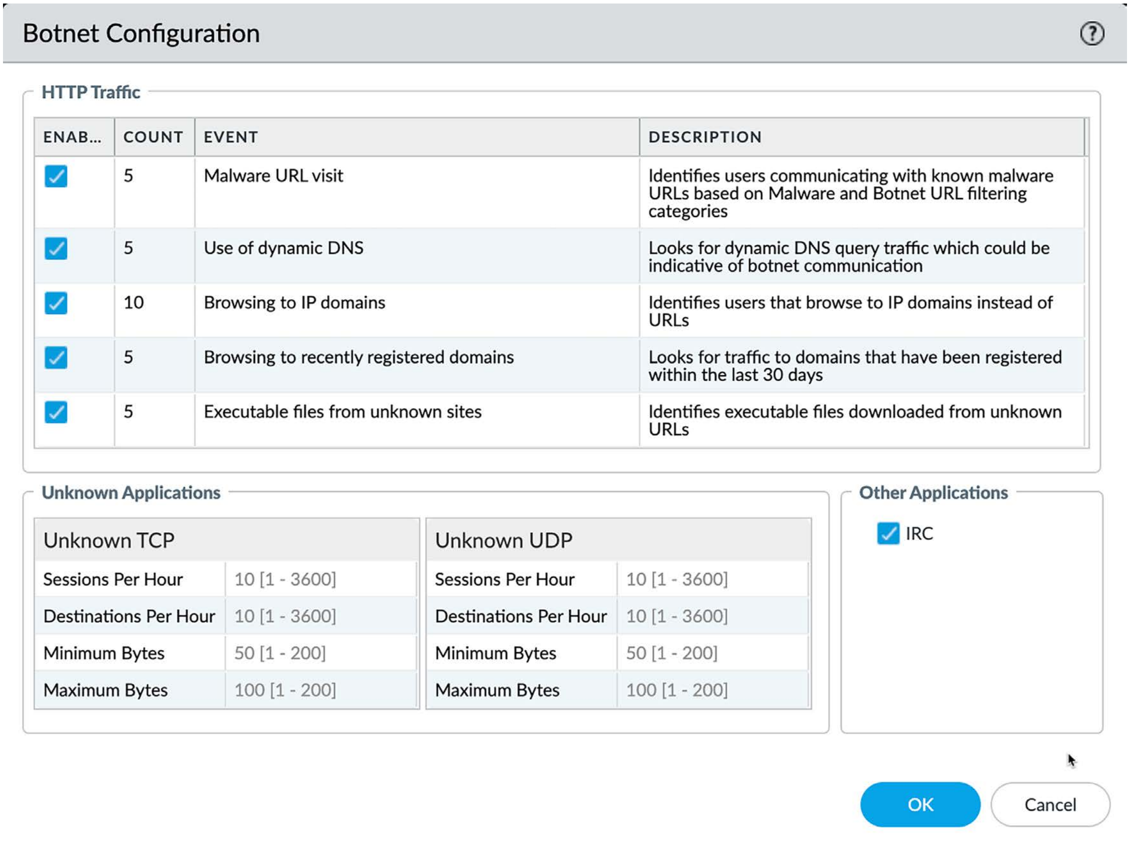


Figure 13.7: Botnet report configuration

Now that you have a good understanding of how to filter logs and capture traffic, we'll take a look at what a session is made up of to help you interpret it.

Interpreting session details

The log details tell you a lot about a session, but not everything. Sessions, while being processed, have several different parameters that only translate to how they are being processed at a particular moment in time.

One such caveat is when **Log at Session Start** is enabled on a security rule, a log will only appear once the first data packet is received rather than when the TCP handshake is completed. This means a session could already exist in the session table because the handshake was completed successfully, without a log entry being generated as no data has been received yet.

> The session table is made up of a finite number of session IDs, so session IDs end up getting reused after the available IDs have been cycled through.

Understanding session states and types

There are seven different states that a session can be in:

- **Initial or INIT**: A session that is ready and waiting to be used by a new flow is in the `INIT` state.
- **Opening**: This is a transient state in which a session ID is assigned to a flow while it is being evaluated to become a full session. This stage accounts for half-open TCP connections, so it has more aggressive timers that close the session if the handshake is not completed within due time.
- **Active**: This is the state in which everything happens – the flow is up and packets are being passed back and forth.
- **Closing**: This is a transient state. If a flow has reached its time to live or idle timeout, this means the session is set to expire soon but has not been removed from the aging process or the session lookup table.

  During this stage, new packets will no longer be matched against this session and will be queued to create a new session, or they will be discarded because they are **ACK** packets that no longer match an active session (non-SYN TCP).
- **Discard**: Here, the flow is hitting a drop/deny rule or is hitting a threat set to block. All packets matching the session will be discarded for the duration of the discard phase.
- **Closed**: This is a transient state. The session has been removed from the aging process but not from the session lookup table. No new packets can match this session, so they are either queued for a new session or are dropped.
- **Free**: This is a transient state. The session has been closed and removed from the session lookup table but still needs to be made available for a new flow.

  Once the **Free** state has completed, the session is returned to the `INIT` state.

Transient states are usually very short and could be hard to spot. `INIT`, `ACTIVE`, and `DISCARD` are stable states and will represent most of the sessions you would be able to see.

All the timers associated with session creation, time to live, and session teardown can be consulted with the following command:

```
reaper@pa-440> show session info
------snip------
Session timeout
TCP default timeout:                               3600 secs
TCP session timeout before SYN-ACK received:          5 secs
TCP session timeout before 3-way handshaking:        10 secs
TCP half-closed session timeout:                    120 secs
TCP session timeout in TIME_WAIT:                    15 secs
TCP session delayed ack timeout:                    250 millisecs
TCP session timeout for unverified RST:              30 secs
UDP default timeout:                                 30 secs
ICMP default timeout:                                 6 secs
SCTP default timeout:                              3600 secs
SCTP timeout before INIT-ACK received:                5 secs
SCTP timeout before COOKIE received:                 60 secs
SCTP timeout before SHUTDOWN received:               30 secs
other IP default timeout:                            30 secs
Captive Portal session timeout:                      30 secs
Session timeout in discard state:
TCP: 90 secs, UDP: 60 secs, SCTP: 60 secs, other IP protocols: 60 secs
```

All of these timers can also be changed to suit your environment through **Configuration** mode or in **Device** | **Setup** | **Session** | **Session Timeouts**:

```
reaper@pa-4400# set deviceconfig setting session timeout-
+ timeout-captive-portal       set captive-portal session timeout value in
seconds
+ timeout-default              set session default timeout value in seconds
+ timeout-discard-default      set timeout of non-tcp/udp session in discard
state
+ timeout-discard-tcp          set timeout of tcp session in discard state
+ timeout-discard-udp          set timeout of udp session in discard state
+ timeout-icmp                 set icmp timeout value in seconds
+ timeout-scan                 application trickling timeout value in seconds
+ timeout-tcp                  set tcp timeout value in seconds
+ timeout-tcp-half-closed      set session tcp half close timeout (after
receiving first FIN/RST) value in seconds
+ timeout-tcp-time-wait        set session tcp time wait timeout (after
```

```
receiving second FIN/RST) value in seconds
+ timeout-tcp-unverified-rst   set session tcp timeout value after
receiving a RST with unverified sequence number in seconds
+ timeout-tcphandshake         set tcp handshake session timeout (before 3-way
handshaking is completed) value in seconds
+ timeout-tcpinit              set tcp initial session timeout (before SYN-ACK
is received) value in seconds
+ timeout-udp                  set udp timeout value in seconds
```

There are also five session types:

- **FLOW**: These are all the regular sessions.
- **FORW** (forward): This is used when a captive portal is used to intercept and redirect browsing sessions to a login page, or when **policy-based forwarding** (**PBF**) is applied to a flow.
- **PRED** (predict): **Application layer gateway (ALG**) protocols that require a return session to be set up outside of the established session (SIP, FTP, and so on) will set up predict sessions to anticipate the inbound connection. If the return session is received, the predict session will be transformed into a flow session. Predict sessions are based on the control information detected in the outbound session.
- **Tunnel:** VPN connections will be set up in a tunnel session.
- **VNI:** If VXLAN **tunnel content inspection** (**TCI**) is enabled in **Policies | Tunnel Inspection**, VXLAN tunnels will be **VNI**-type sessions.

The sessions can be displayed from **Monitor | Session Browser** and, as you can see in *Figure 13.8*, there's a lot of information regarding the session shown in this display that is not in the logs. There are a couple of interesting fields that can help you understand the state that a session is in:

- **Timeout** is the amount of time a session is allowed to exist.
- **Time To Live** is the amount of time left on the timeout.

The session timeout can tell you a lot about what is going on. An established TCP session may get a timeout of `3600` seconds, while a UDP session may only get `30` seconds. A `DISCARD` stage session will also only get a short timeout.

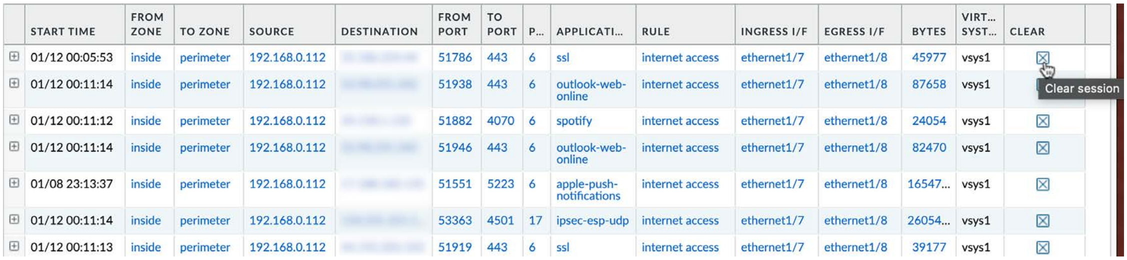
When troubleshooting sessions that go out to the internet, incorrectly configured NAT is often the root cause:

- The NAT source and destination are indicated by **True** or **False**.
- The name of the NAT rule is used by the session.
- Flow 1 is the **client-to-server (c2s)** flow and shows the original source IP (10.0.0.8) and the port destined for the server.
- Flow 2 is the **server-to-client (s2c)** flow and shows the server's IP to the NAT's IP (192.168.27.251) that the client is translated behind, and the NAT's source port as the destination port (44666) for the returning flow.

Reviewing all of these things can help in spotting NAT issues early on.

Terminating and clearing sessions

Sessions can be forcibly terminated by clicking on the **X** mark under the **CLEAR** column, as you can see in the following screenshot. This will set the session in the `INIT` state immediately. Any packets still arriving on the firewall will not have any sessions to match against, so will either be discarded as non-SYN TCP or evaluated for a new session to be created:



START TIME	FROM ZONE	TO ZONE	SOURCE	DESTINATION	FROM PORT	TO PORT	P...	APPLICATI...	RULE	INGRESS I/F	EGRESS I/F	BYTES	VIRT... SYST...	CLEAR
01/12 00:05:53	inside	perimeter	192.168.0.112		51786	443	6	ssl	internet access	ethernet1/7	ethernet1/8	45977	vsys1	
01/12 00:11:14	inside	perimeter	192.168.0.112		51938	443	6	outlook-web-online	internet access	ethernet1/7	ethernet1/8	87658	vsys1	
01/12 00:11:12	inside	perimeter	192.168.0.112		51882	4070	6	spotify	internet access	ethernet1/7	ethernet1/8	24054	vsys1	
01/12 00:11:14	inside	perimeter	192.168.0.112		51946	443	6	outlook-web-online	internet access	ethernet1/7	ethernet1/8	82470	vsys1	
01/08 23:13:37	inside	perimeter	192.168.0.112		51551	5223	6	apple-push-notifications	internet access	ethernet1/7	ethernet1/8	16547...	vsys1	
01/12 00:11:14	inside	perimeter	192.168.0.112		53363	4501	17	ipsec-esp-udp	internet access	ethernet1/7	ethernet1/8	26054...	vsys1	
01/12 00:11:13	inside	perimeter	192.168.0.112		51919	443	6	ssl	internet access	ethernet1/7	ethernet1/8	39177	vsys1	

Figure 13.8: Session browser

Sessions can also be cleared from the CLI using the following commands.

To clear a single session, use the ID:

```
reaper@pa-440> clear session id <ID>
```

To clear every single session, use the following command:

```
reaper@pa-440> clear session all
```

You can also add a filter to the previous command to delete all sessions that match the filter:

```
reaper@pa-440> clear session all filter source 192.168.0.1
```

Viewing session data from the CLI

From the CLI, the same information we saw in the GUI session browser and more can be collected, as you can see with the following command. The CLI allows more flexible use of filter options, so this will usually be the preferred way to review sessions:

```
reaper@pa-440> show session id 256
Session             256
        c2s flow:
                source:      10.0.0.8 [trust]
                dst:         204.79.197.222
                proto:       6
                sport:       49710           dport:      443
                state:       DISCARD         type:       FLOW
                src user:    unknown
                dst user:    unknown
```

```
        s2c flow:
                source:      204.79.197.222 [untrust]
                dst:         192.168.27.251
                proto:       6
                sport:       443             dport:      44666
                state:       DISCARD         type:       FLOW
                src user:    unknown
                dst user:    unknown
        start time                          : Tue Mar 25 23:20:13
        timeout                             : 90 sec
        time to live                        : 79 sec
        total byte count(c2s)               : 316
        total byte count(s2c)               : 66
        layer7 packet count(c2s)            : 3
        layer7 packet count(s2c)            : 1
        vsys                                : vsys1
        application                         : ssl
        rule                                : block push
        service timeout override(index)     : False
        session to be logged at end         : True
        session in session ager             : True
        session updated by HA peer          : False
        address/port translation            : source
        nat-rule                            : outbound hide(vsys1)
        layer7 processing                   : enabled
        URL filtering enabled               : True
        URL category                        :
        session via syn-cookies             : False
        session terminated on host          : False
        session traverses tunnel            : False
        session terminate tunnel            : False
        captive portal session              : False
        ingress interface                   : ethernet1/2
        egress interface                    : ethernet1/1
        session QoS rule                    : N/A (class 4)
        tracker stage firewall              : appid policy lookup deny
        end-reason                          : policy-deny
```

The CLI also shows `tracker stage firewall`, which indicates why a session was closed. In the case of the preceding session, an application was detected that was denied by the security policy, and the session was put into the `DISCARD` state.

Other tracker stages are as follows:

- **Aged out**: The session has reached its timeout.
- **TCP FIN**: FIN packet received to terminate the session.
- **TCP RST client or – server**: The client or server has sent an RST packet.
- **Appid policy lookup deny**: Policy lookup sets an application to deny or drop.
- **Mitigation tdb**: Threat detected that terminates the session.
- **Resource limit**: Rollup of many errors that could happen in a flow (exceeded packets out of order in a flow, and so on).
- **Host service**: Sessions set up toward the firewall for a service that is not allowed from this source or not enabled on this interface.
- **L7 proc**: Processing of `layer7` ongoing. In the case of a `DISCARD` session, this could be a child application that requires additional App-ID effort to identify (as opposed to **Appid policy lookup deny**).
- **ctd decoder bypass**: A session has reached the end of its content inspection and was offloaded to hardware.
- **Session rematch**: This session was previously allowed but new security has been pushed that now blocks this session.

Other session attributes can include the following. Some attributes that are not relevant to a session will not be displayed:

- **Layer7 processing**: If an application override is in place or the protocol in the session does not have a decoder, **Layer7 processing** will be `False`.
- **Session via SYN cookies**: This shows whether SYN cookies were used when the session was set up (these are controlled from the zone protection profile).
- **To Host Session**: This is `true` when the session is connecting to a service running on the firewall, such as a DNS proxy or a management profile.
- **Session traverses tunnel**: These are sessions that are going into an IPSec, SSL, or GRE tunnel.
- **Session terminates tunnel**: These are sessions that terminate a tunnel on the firewall.
- **Session QoS rule**: This indicates whether a **quality of service** (**QoS**) rule is used for this session, and the class assigned to the session.
- **Captive Portal**: This is set to `true` if a session was created that intercepted and redirected a client session to the captive portal page. The s2c flow will indicate whether the original destination was replaced by a captive portal redirect, while the c2s flow has the captive portal as the destination.

A captive portal-type session will look similar to the following output:

```
reaper@pa-440> show session id 865
Session             865
        c2s flow:
                source:      10.0.0.8 [trust]
                dst:         10.0.0.1
                proto:       6
                sport:       50311           dport:      6081
                state:       INIT            type:       FLOW
                src user:    unknown
                dst user:    unknown
        s2c flow:
                source:      127.131.1.1 [captive-portal]
                dst:         10.0.0.8
                proto:       6
                sport:       6181            dport:      50311
                state:       INIT            type:       FLOW
                src user:    unknown
                dst user:    unknown
```

Applying filters

To get a list of all the active sessions, you can use the following command:

```
reaper@pa-440> show session all
```

There are many filters that can be applied to narrow down the output of the above command. An easy trick is to use the *Tab* key to see which options are available:

```
reaper@pa-440> show session all filter <tab>
+ application          Application name
+ count                count number of sessions only
+ ctd-ver              ctd version
+ decrypt-forwarded    session is decrypt forwarded
+ decrypt-mirror       session is mirrored
+ destination          destination IP address
+ destination-port     Destination port
+ destination-user     Destination user
…
```

In the following screenshot, you can see several filters being applied to narrow my search. The resulting output lists each session in two rows: the top row is the c2s flow and the bottom row is the s2c flow. `Flag` indicates whether the session is applying source NAT (`NS`), destination NAT (`ND`), or both (`NB`):

```
reaper@PA-VM> show session all filter protocol 6 nat source from trust type flow state active

--------------------------------------------------------------------------------
ID          Application    State   Type Flag  Src[Sport]/Zone/Proto (translated IP[Port])
Vsys                                          Dst[Dport]/Zone (translated IP[Port])
--------------------------------------------------------------------------------
261          ss1            ACTIVE  FLOW  NS   10.0.0.8[49915]/trust/6  (192.168.27.251(35448])
vsys1                                          [redacted].122.2[443]/untrust  ([redacted].122.2[443])
353          web-browsing   ACTIVE  FLOW  NS   10.0.0.8[50011)/trust/6  (192.168.27.251(43839])
vsys1                                          [redacted].4.52[80]/untrust  ([redacted].4.52[80])
356          web-browsing   ACTIVE  FLOW  NS   10.0.0.8[500101/trust/6  (192.168.27.251[54552])
vsys1                                          [redacted].4.52[80]/untrust  ([redacted].4.52[80])
253          ss1            ACTIVE  FLOW  NS   10.0.0.8[49918]/trust/6  (192.168.27.251(64354])
vsys1                                          [redacted].37.44[443]/untrust  ([redacted].37.44[443])
267          ss1            ACTIVE  FLOW  NS   10.0.0.8[49919]/trust/6  (192.168.27.251[3751])
vsys1                                          [redacted].38.49[443]/untrust  ([redacted].38.49[443])
231          ss1            ACTIVE  FLOW  NS   10.0.0.8[49917]/trust/6  (192.168.27.251(16008])
vsys1                                          [redacted].121.44[443]/untrust  ([redacted]..121.44[443])
```

Figure 13.9: The output of the show session all command with a filter applied

By default, the system view in the command line is `VSYS1`. For most commands, this does not matter, but if you need to list sessions in `VSYS2`, you will first need to change the system perspective to `VSYS2` so that the commands relate to the correct `VSYS`. Use the following command to switch to the `VSYS` perspective:

```
reaper@pa-3220> set system setting target-vsys ?
none none
vsys1 prod
vsys2 beta
```

You should now be able to find a session and correlate it to the expected behavior. You can see whether the session is being allowed or blocked and whether NAT, QoS, or PBF are being applied as expected. In the next section, we will review the troubleshooting tools that allow us to see how a session will behave before it has taken place.

Using the troubleshooting tool

The web interface is a very convenient way to configure the firewall, but it also holds several tools that you can use to troubleshoot issues you might encounter. The troubleshooting tool, which you can find in **Device** | **Troubleshooting**, lets you run several tests past your configuration to see what the system is expected to do in the given situation.

Some of the available tests let you verify whether the system can connect to cloud services, as illustrated in the following screenshot.

Click on the test result to see the **Result Detail** pane on the right-hand side:

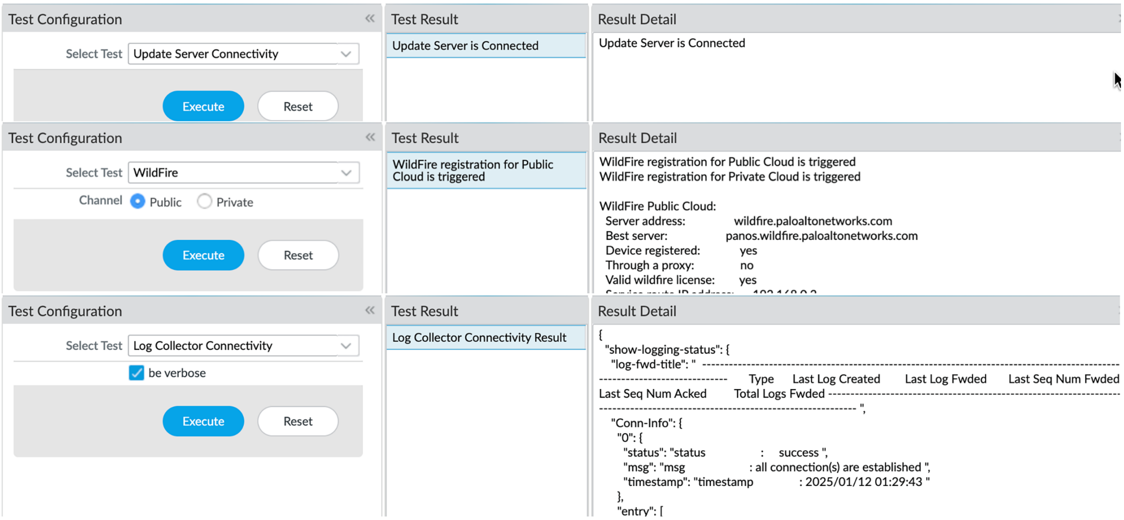


Figure 13.10: A cloud connectivity test

Testing policies

The troubleshooting tool lets you test several policies to see whether they will behave as you expect. The following policies can be tested:

- Security policy match
- QoS policy match
- Authentication policy match
- Decryption/SSL policy match
- NAT policy match
- Policy-based forwarding policy match
- DoS policy match

The following screenshot shows a security policy match test; you can put in some parameters, such as `source IP`, `destination IP`, `destination port`, `protocol`, `application`, or `URL Category`. The system will match your set of parameters against the entire security rulebase to see which rule matches:

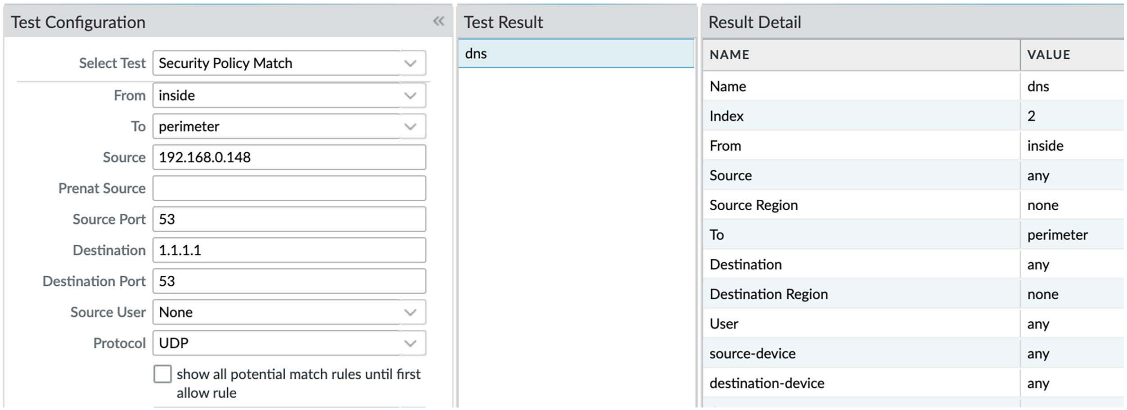


Figure 13.11: Security Policy Match

Note

The **URL Category** parameter in **Security Policy Match** only reflects rules that have a category set in the destination. This is not matched against **URL filtering profiles**.

Testing connectivity

The troubleshooting tool can also be used to test connectivity. The ping test lets you send ICMP echo requests to a host. You can define some typical parameters, such as the following:

- **Count:** This is the number of ping requests to send.
- **Don't Fragment:** This can be checked to add the DF bit to ping packets. Combined with the **Size** option, this can be used to test MTU toward a target destination.
- **Interval:** This is the time between requests in seconds.
- **Source:** This is the data plane interface to send the packets from.
- **Host:** This is the destination to be pinged.
- **Size:** This is the payload size of the ping packet. Changing this can be useful to test whether larger packets take longer to return or get dropped along the route.
- **Tos:** This lets you set a **Type of Service** (**ToS**) IP option to verify whether upstream devices apply.
- **Ttl:** This is the maximum number of hops the packet can pass before being discarded. The default is 58.

There are also a few special features you can set:

- **Bypass routing table, use specific interface** lets you put packets directly into an interface instead of performing a routing lookup. This can be useful to test a redundant path.
- **Don't fragment echo request packets (IPv4)** lets you set the **Don't Fragment** bit in the IP header of ping packets, which is useful if you want to discover `Path MTU` by sending ever-increasing-sized ping packets. When you reach the size where the packets are dropped, you have found the maximum MTU that your path will allow, as packets that are too large and are not allowed to be fragmented must be discarded.
- **Pattern** lets you add a specific pattern to the payload, which can help identify the packet in an upstream device:

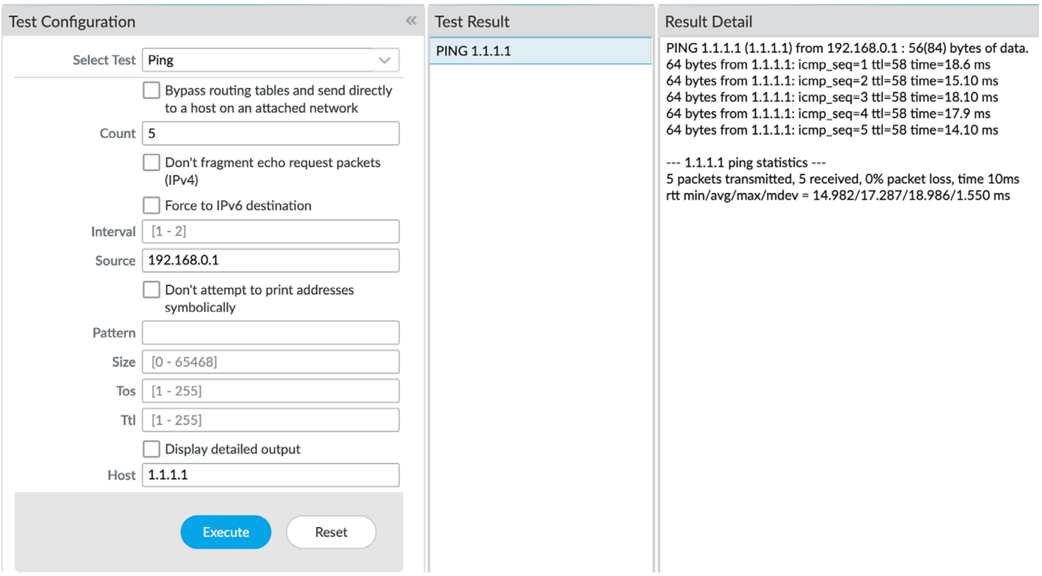


Figure 13.12: The ping tool

All of these options are available from the CLI as well:

```
reaper@pa-440> ping
+ bypass-routing    Bypass routing table, use specified interface
+ count             Number of requests to send (1..2000000000 packets)
+ do-not-fragment   Don't fragment echo request packets (IPv4)
+ inet6             Force to IPv6 destination
+ interval          Delay between requests (seconds)
+ no-resolve        Don't attempt to print addresses symbolically
+ pattern           Hexadecimal fill pattern
+ size              Size of request packets (0..65468 bytes)
+ source            Source address of echo request
+ tos               IP type-of-service value (0..255)
```

```
+ ttl              IP time-to-live value (IPv6 hop-limit value) (0..255 hops)
+ verbose          Display detailed output
* host             Hostname or IP address of remote host
```

The output should look similar to this:

```
reaper@pa-440> ping count 2 interval 1 source 192.168.27.2 host 1.1.1.1
PING 1.1.1.1 (1.1.1.1) from 192.168.27.2 : 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=58 time=10.9 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=58 time=15.1 ms
--- 1.1.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1017ms
rtt min/avg/max/mdev = 10.972/13.073/15.174/2.101 ms
```

Testing with traceroute

Where there is a ping, there is traceroute. A traceroute test lets you send UDP traceroute packets to identify hops along the path toward a remote host. This is a very practical tool to find routing issues; packets are sent out with an ever-increasing TTL value starting from `1`, and each hop along the path is required to decrease the TTL counter by 1 before sending the packet to the next hop. If the counter reaches `0`, the hop must discard the packet and send back an ICMP option `11` (time exceeded) packet to the sender. The sender will, theoretically, receive a notification from all the hosts along the path to the final destination, revealing the routing involved in getting packets to the final destination.

The following options can be set to tailor traceroute to your needs:

- Both IPv4 and IPv6 can be tested.
- **First Ttl** lets you set a starting TTL higher than `1`, which could be useful if the first few hops are not to be included in the test or the resulting output.
- **Max Ttl** is the maximum number of hops taken before giving up.
- **Port** lets you set a static destination port used for the UDP packet. By default, a random high port is chosen at the start of the test, sequentially increasing with every packet sent.
- **Tos** lets you set the ToS IP option.
- **Wait** is the number of seconds that the firewall should wait for a reply message to arrive.
- **Pause** is the amount of time, in milliseconds, that the firewall should wait between probes.
- **Gateway** lets you set up to `8` **loose source routing** gateways.
- **Don't attempt to print addresses symbolically** prevents a reverse lookup of the IP against the DNS.
- **Bypass routing tables and send directly to a host** puts the packets directly onto the wire.
- **Source** is the data plane interface to use as the source. By default, the management interface is used.
- **Host** is where the traceroute should try to reach.

As you can see, there are plenty of options to make the test more thorough:

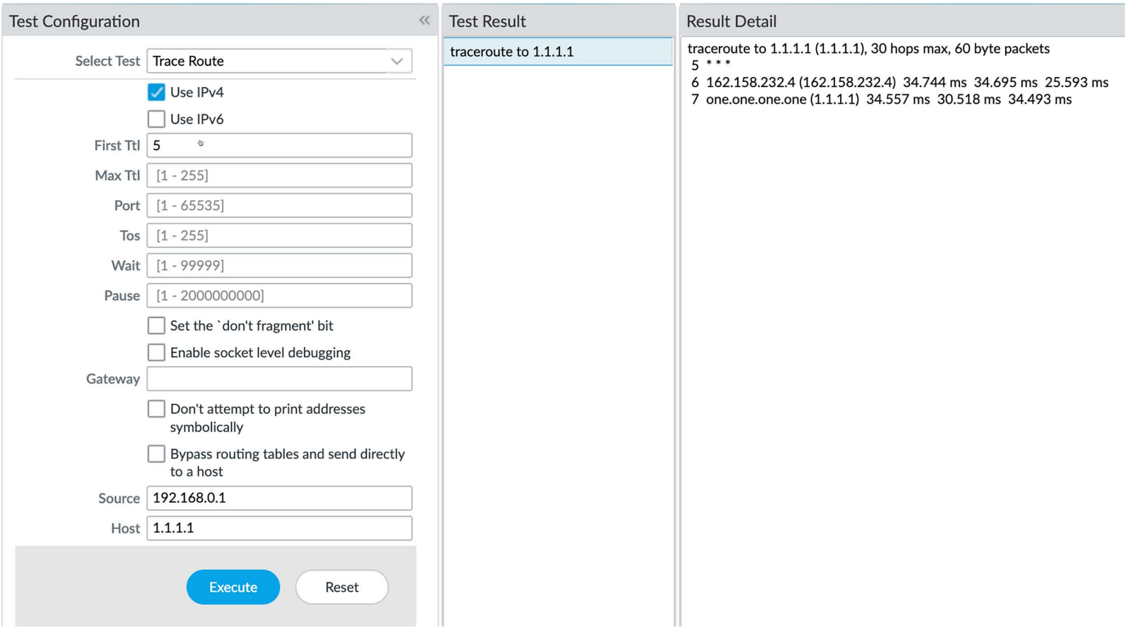


Figure 13.13: A traceroute test

Traceroute can also be executed from the CLI with all the same options:

```
reaper@pa-440> traceroute
+ bypass-routing     Bypass routing tables and send directly to a host
+ debug-socket       Enable socket level debugging
+ do-not-fragment    Set the 'don't fragment' bit
+ first-ttl          time-to-live used in the first outgoing probe packet
+ gateway            Specify a loose source route gateway (8 maximum)
+ ipv4               Use IPv4
+ ipv6               Use IPv6
+ max-ttl            Set the max time-to-live (max number of hops)
+ no-resolve         Don't attempt to print addresses symbolically
+ pause              Set the time (in milliseconds) to pause between probes
+ port               Set the base port number used in probes (default udp/33434
tcp/80 icmp/1)
+ source             Use specified source address in outgoing probe packets
+ tos                IP type-of-service value (0..255)
+ wait               Set number of seconds to wait for a response
* host               Hostname or IP address of remote host
```

The output for a traceroute test should look similar to this:

```
reaper@pa-440> traceroute first-ttl 4 ipv4 yes source 192.168.27.2 no-resolve
yes host 1.1.1.1
traceroute to 1.1.1.1 (1.1.1.1), 30 hops max, 60 byte packets
4  * * *
5  213.224.125.31  20.784 ms  21.148 ms  20.968 ms
6  81.20.71.70  20.383 ms  20.179 ms  21.393 ms
7  1.1.1.1  19.391 ms * *
```

You can now determine whether the firewall has proper connectivity to its services and whether an expected session will hit all the appropriate policies.

Every once in a while, a system may encounter unexplained issues and require some low-level debugging or may need to be reset to factory settings to be configured from scratch. This can be achieved from Maintenance Mode.

Using Maintenance Mode to resolve and recover from system issues

The MRT, also called **Maintenance Mode**, resides on a separate bootable partition and can be invoked if the system has an unexpected failure. If, for example, the system is unable to complete the auto-commit process, it will reboot to try and rectify what is causing the failure. If, after three reboots, the auto-commit is still failing, the system will boot into Maintenance Mode.

> If the system failed, you can SSH into the device using the `maint` username and the *serial number* of the device as the password. If you connect to the console, you don't need a username and password.

You can force the system to boot into Maintenance Mode from the command line by executing the following command. The system will ask whether you want to reboot after you hit *Enter*:

```
> debug system maintenance-mode
```

You can also manually start **Maintenance Mode**. During the boot process, there are two options.

In older PAN-OS versions, there will be a short window where a dialog asks whether you want to interrupt the boot sequence by hitting any key. If you do, you have 5 seconds to take action. If you type `maint`, you will be taken to the bootloader, where you can choose the maintenance partition.

In newer PAN-OS versions (10.2 and later), you will see the bootable partitions appear for a short time during the boot process. If you use the arrow keys when the partitions appear, the boot process is interrupted and you can select the desired boot partition, as you can see in the following screenshot:

```
      PANOS (maint-sysroot0)
      PANOS (maint-sysroot1)
      PANOS (sysroot0)
      PANOS (sysroot1)



      Use the ^ and v keys to change the selection.
      Press 'e' to edit the selected item, or 'c' for a command prompt.
```

Figure 13.14: The Maintenance Mode bootloader in PAN-OS 10.2 and later

If you select the maintenance partition, you are taken to a welcome page that has details on getting support:

```
                    Welcome to the Maintenance Recovery Tool



     Welcome to maintenance mode. For support please contact Palo Alto
     Networks.

                  866-898-9087 or support@paloaltonetworks.com


< Continue                                                               >



               Q=Quit,  Up/Down=Navigate,  ENTER=Select,  ESC=Back
```

Figure 13.15: The MRT

Once you hit *Enter*, you will be taken to the main menu, as follows:

```
                    Welcome to the Maintenance Recovery Tool


< Maintenance Entry Reason                                                >
< Get System Info                                                         >
< Factory Reset                                                           >
< Set FIPS-CC Mode                                                        >
< FSCK (Disk Check)                                                       >
< Log Files                                                               >
< Disk Image                                                              >
< Select Running Config                                                   >
< Content Rollback                                                        >
< Set IP Address                                                          >
< Diagnostics                                                             >
< Debug Reboot                                                            >
< Reboot                                                                  >


               Q=Quit,  Up/Down=Navigate,  ENTER=Select,  ESC=Back
```

Figure 13.16: Maintenance Mode main menu

Any advanced features that require a password can be accessed using `MA1NT` as the password.

Let's take a look at the options available from the main menu.

If **Maintenance Mode** was invoked by the system (rather than requested by the admin), there should be some additional information in **Maintenance Entry Reason.**

Get System Info returns an overview of all the system information, such as the serial number, installed OS, and content updates.

Factory Reset lets you revert the system to clean factory settings. The configuration files are purged and reset to the default configuration and all logs and reports are wiped from the system.

If you require the logs to be securely purged and not just deleted, you can opt to scrub the system. You can pick between the **NNSA** (overwrite all locations twice with a pseudo-random pattern and once with a known pattern) and **DOD** (overwrite all addressable locations with a character, its complement, and then a random character) scrub.

You can choose which PAN-OS version the system should be set to during the reset. By default, the system is reset to the most currently installed PAN-OS version; in the **Advanced** menu option, you can choose an older PAN-OS version to install:

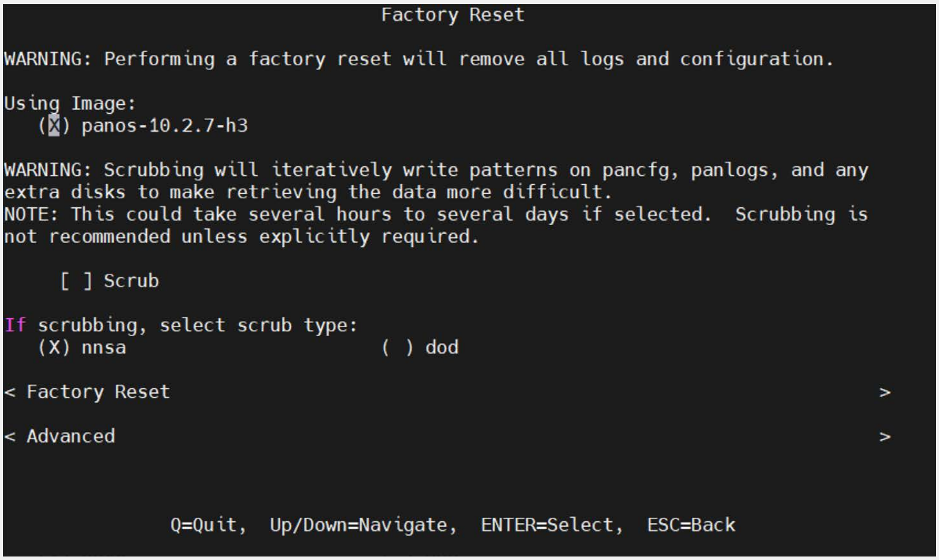


Figure 13.17: Factory reset

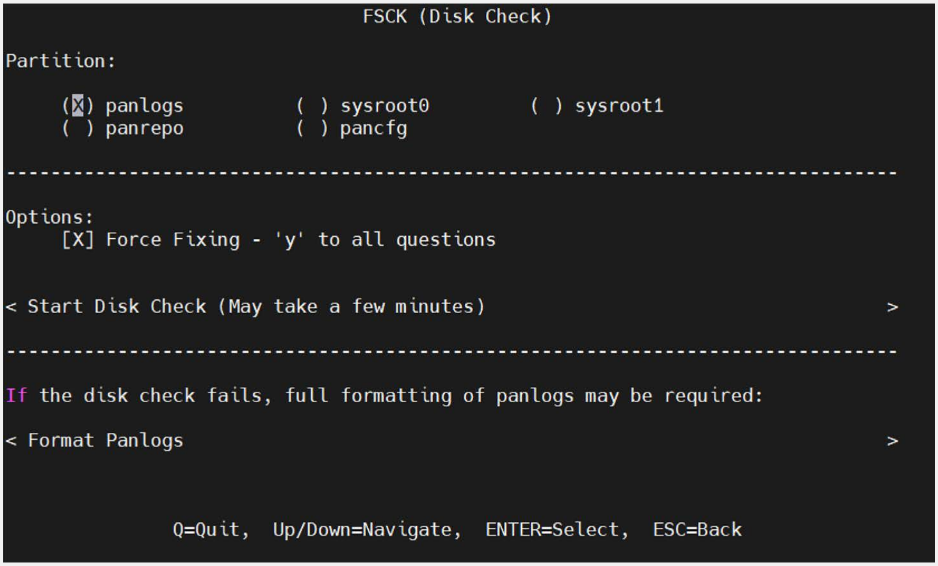
Set FIPS-CC Mode converts the system into FIPS compliance; it will take the following actions:

- It will disable all weak crypto ciphers
- It will disable the console port as the CLI, only allowing it to function as an output port
- It will set the minimum password length to `6`
- Weak management protocols (such as `http`, `ftp`, and `telnet`) are disabled and are no longer available
- Encryption on **HA1** is mandatory

FSCK can be used to scan all the partitions for issues and attempt to repair any bad sectors. You can scan the following partitions:

- `panlogs`
- `panrepo`
- `sysroot0`
- `sysroot1`
- `pancfg`

You can opt to automatically select *Y* for any question to fix a bad sector, and you can format the **panlogs** partition if the disk check fails for that partition.



```
                              FSCK (Disk Check)

Partition:

     (X) panlogs           ( ) sysroot0          ( ) sysroot1
     ( ) panrepo           ( ) pancfg

--------------------------------------------------------------------------------

Options:
     [X] Force Fixing - 'y' to all questions

< Start Disk Check (May take a few minutes)                                    >

--------------------------------------------------------------------------------

If the disk check fails, full formatting of panlogs may be required:

< Format Panlogs                                                               >


               Q=Quit,  Up/Down=Navigate,  ENTER=Select,  ESC=Back
```

Figure 13.18: FSCK can repair partitions

Log Files lets you access all the system logs in case you need to review whether a process was able to write a critical error. The log files can also be exported to an external system for further investigation:

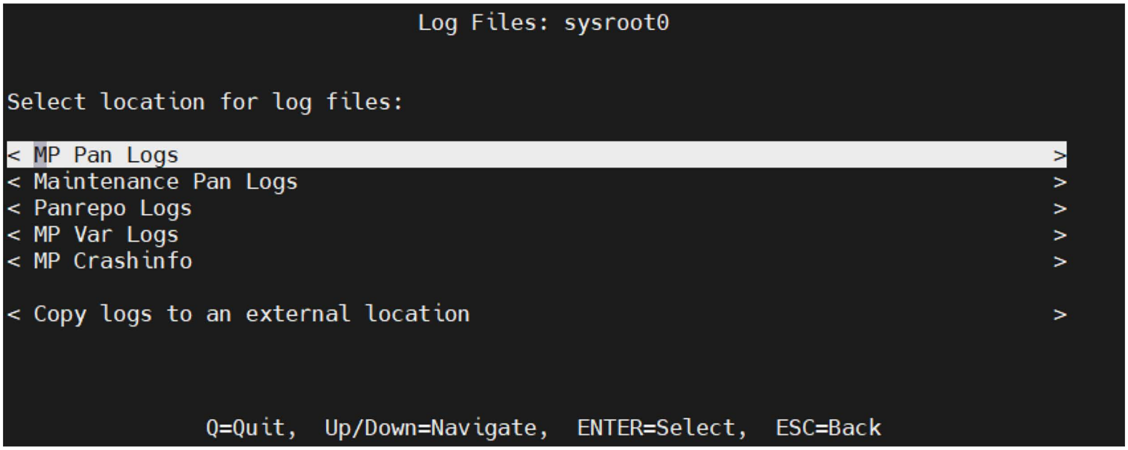


Figure 13.19: Access or export log files

If you want to export the log files to an external location, you need to select **Set IP address** in the top menu before you can start the copy.

Disk Image lets you reinstall the currently installed PAN-OS version, without changing the running configuration, or revert to the previously installed version. In the **Advanced** options, you can do the following:

- Review the install history and current bootable partition status
- Revert to the previously installed PAN-OS version
- Verify the integrity of the currently installed image
- Purge older images from the disk
- Manually select which partition to set as the boot in the bootloader
- Manually boot into a PAN-OS version without changing the bootloader

Select Running Config lets you select previously saved configuration files and set them as the running config, which can come in quite handy if you lose your admin password, as there is no password recovery procedure other than performing a factory reset or loading a saved configuration.

Important note

Don't save a configuration file containing a default `admin/admin` account as that will allow a backdoor for anyone able to boot into **Maintenance Mode**.

Content Rollback lets you revert to an older content version package if something went wrong when a content update was installed.

Set IP Address lets you manually set an IP if the device does not load its management IP or is unable to get a DHCP IP.

Diagnostics runs a disk performance check.

Debug Reboot reboots the system but outputs all boot dialog in verbose mode, which will help if the system fails to boot.

With this knowledge, you will be able to recover from several highly critical failures, or at least collect sufficient information to perform a postmortem and find out what caused the situation in the first place.

Summary

In this chapter, you learned where to find all the different types of log files and how aggregated information can lead to identifying a botnet. You can now perform a packet capture using filters to capture only what you need. You can interpret a session on the firewall and identify key attributes such as the NAT direction, the end reason, and the timeout settings. You can verify whether the firewall has connectivity to all its cloud services and whether an anticipated flow will hit all the intended policies using the troubleshooting tool. You can also perform key tasks such as a factory reset or loading a different configuration file from **Maintenance Mode**.

If you're preparing for the PCNSE, troubleshooting is quite an important part. Remember what information can be found in each log file, and specifically memorize which log databases are available. Carefully review the session states and the information contained in session outputs.

In the next chapter, we will take what you have learned in this chapter to the next level by using packet capture filters to analyze global counters and look at the actual flow as it goes through the firewall and is touched by different processes.

14

A Deep Dive Into Troubleshooting

In this chapter, we will learn how a session is formed and how flows traverse the firewall. We will learn how to interpret global counters and take things a step further to look at all the stages that a packet goes through between entering and leaving the firewall. We will see how sessions are set up and how packets are handled at every step.

In this chapter, we're going to cover the following main topics:

- Understanding global counters
- Analyzing session flows
- Debugging processes
- CLI troubleshooting commands cheat sheet

By the end of this chapter, you'll be able to analyze all the phases a packet and a session go through while traversing the firewall, and figure out what might be going wrong.

Technical requirements

For this chapter, it is strongly recommended that you have a lab environment where you can emulate the steps we will be taking to get a better feel for the commands and the output we will be reviewing. There is a cheat sheet with useful commands and a list of all the global counters at the end of the chapter, also available at `https://github.com/PacktPublishing/Mastering-Palo-Alto-Networks-Third-Edition`.

Understanding global counters

When you are troubleshooting a connectivity issue, the log files and packet captures provide a wealth of information, but sometimes they're not enough to figure out what is happening to a session. All sessions, whether they are traversing the firewall or getting dropped, are tracked by all the processes that touch them. This tracking is done by counters that increment with each step a packet takes or action a process performs for each packet in a session. This can provide a wealth of information if something is not working as expected.

The global counters can be viewed by running the following command:

```
reaper@PA-440> show counter global
```

This will output all of the global counters without context, which is not very useful. You can add a `delta` filter to only show global counters for the period between the previous and the last time that the command was issued. The duration will be indicated in the output:

```
reaper@PA-440> show counter global filter delta yes
```

The output will look similar to the following screenshot:

```
reaper@PA-440> show counter global filter delta yes

Global counters:
Elapsed time since last sampling: 0.937 seconds

name                              value     rate severity  category  aspect    description
--------------------------------------------------------------------------------
pkt_recv                              4        4 info      packet    pktproc   Packets received
pkt_runtc_np                          4        4 info      packet    pktproc   runtc flow_np
pkt_sent                              2        2 info      packet    pktproc   Packets transmitted
pkt_stp_rcv                           2        2 info      packet    pktproc   STP BPDU packets received
flow_qos_pkt_enque                    2        2 info      flow      qos       Packet enqueued to QoS module
flow_qos_pkt_deque                    2        2 info      flow      qos       Packet dequeued from QoS module
flow_dos_ag_buckets_upd               2        2 info      flow      dos       Updated aggregate buckets for aging
flow_sdwan_internal_ager             94      100 info      flow      sdwan     SDWAN FEC/PD ager count
flow_ip_cksm_sw_validation            2        2 info      flow      pktproc   Packets for which IP checksum validation was done
in software
flow_tcp_cksm_sw_validation           2        2 info      flow      pktproc   Packets for which TCP checksum validation was done
 in software
--------------------------------------------------------------------------------
Total counters shown: 10
--------------------------------------------------------------------------------

reaper@PA-440>
```

Figure 14.1: The show counter global delta output

This is far easier to read than simply outputting all of the counters, but the counters are still system-wide. We'll look at adding more specific filters in a little bit, but we first need to learn a little more about the counters themselves.

Let's take a look at the attributes of a global counter:

- `name`: Each counter has a name that usually tries to convey which process saw what. For example, `flow` is used for packet processing and `_fwd` is used to indicate packets that need to be forwarded somewhere, while `_arp` is used for **ARP** (**Address Resolution Protocol**) packets that don't need to be routed, and `_l3` indicates that they were received on a layer 3 (routing) interface.
- `value`: This is the total number of hits on that counter over the full duration of the delta.
- `rate`: This is an approximate progression of the hits per second over the specified duration as seen by the system. If you see a number in the `value` field but the rate is `0`, there hasn't been a hit on the counter for at least a short while. This could indicate a cluster of hits at the beginning of the delta and none near the end. Repeat the command a few times to gain more insight.
- `severity`: There are four levels of severity:
    - `info` is the default severity for all counters.
    - `drop` is used to indicate something that was intentionally discarded. This could be due to a security policy, a threat profile, or an irregularity relating to where a packet is coming from or needs to go.

    - `error` indicates packets that are malformed and are discarded.
    - `warn` is used when something goes wrong at the system level or if there is an abnormality in received packets – for example, a failed reassembly of a fragmented packet or a split handshake.

- `category`: This indicates which process this counter is related to. A few interesting ones include the following:
    - `aho` is a threat- and data-filtering algorithm engine.
    - `appid` is for the counters related to App-ID processing.
    - `cad` is for Cloud App-ID.
    - `ctd` is for content inspection events.
    - `dfa` is the App-ID algorithm engine. Counters indicate packets going into the engine.
    - `dlp` is for data loss prevention events.
    - `fpga` (**field-programmable gate array**) is the hardware offloading chip. This one is only included in `PA-3000` and higher hardware models.
    - `flow` is for packet processing.
    - `nat` is the **Network Address Translation** (**NAT**) action.
    - `packet` is for packet buffering events.
    - `proxy` is for proxy events, such as SSL decryption or DNS proxy.
    - `session` is for session management
    - `uid` is for user ID events.
    - `zip` triggers when `.zip` files are being unpacked.
    - `tcp` is for TCP packet events.
    - `mprelay` is triggered when sessions require interaction with the management plane routing process.
- `aspect`: This adds more detail regarding which stage a packet was in when the counter was incremented. For example:
    - `arp` for ARP packet processing
    - `dos` for packets matching a zone protection profile
    - `forward` for packet forwarding
    - `ipfrag` for fragmentation
    - `-offload` for packets that are offloaded to hardware
    - `parse` for packet parsing
    - `pktproc` for packet processing
    - `qos` for **quality of service** (**QoS**) enforcement
    - `session` for session setup and teardown
- `description`: This helps to identify counters more clearly.

You can use all of these attributes to filter the global counters for more meaningful outputs, as shown in the following screenshot:

```
reaper@PA-440> show counter global filter delta yes severity drop

Global counters:
Elapsed time since last sampling: 58.466 seconds

name                                  value     rate severity  category  aspect    description
--------------------------------------------------------------------------------
flow_fwd_l3_bcast_drop                  104        1 drop      flow      forward   Packets dropped: unhandled IP broadcast
flow_fwd_l3_mcast_drop                  330        5 drop      flow      forward   Packets dropped: no route for IP multicast
flow_mld_rcv_err                          6        0 drop      flow      mld       MLD receive error
flow_host_service_deny                   10        0 drop      flow      mgmt      Device management session denied
--------------------------------------------------------------------------------
Total counters shown: 4
--------------------------------------------------------------------------------
```

Figure 14.2: Global counters with a severity filter

Here, by adding `severity drop` as an attribute in the filter, only counters that indicate that a packet was discarded will be returned. This can be very useful in finding out if and why packets are dropped.

However, this still only reflects system-wide global counters. To narrow down the scope to just the sessions we want to know more about, we can leverage the same filters used for packet captures.

I have set up a lab device that is pinging out to `194.7.1.4`, which is a public DNS server. We will use these pings to show how global counters can be filtered to return information on just one flow we are interested in.

To filter global counters, we first need to set up the same `packet-diag` filters we would use to set up a packet capture.

First, clear any previously configured filters:

```
reaper@PA-440> debug dataplane packet-diag clear all
```

Unmark any sessions that were marked by the previous filter:

```
reaper@PA-440> debug dataplane packet-diag clear filter-marked-session al
```

Important note

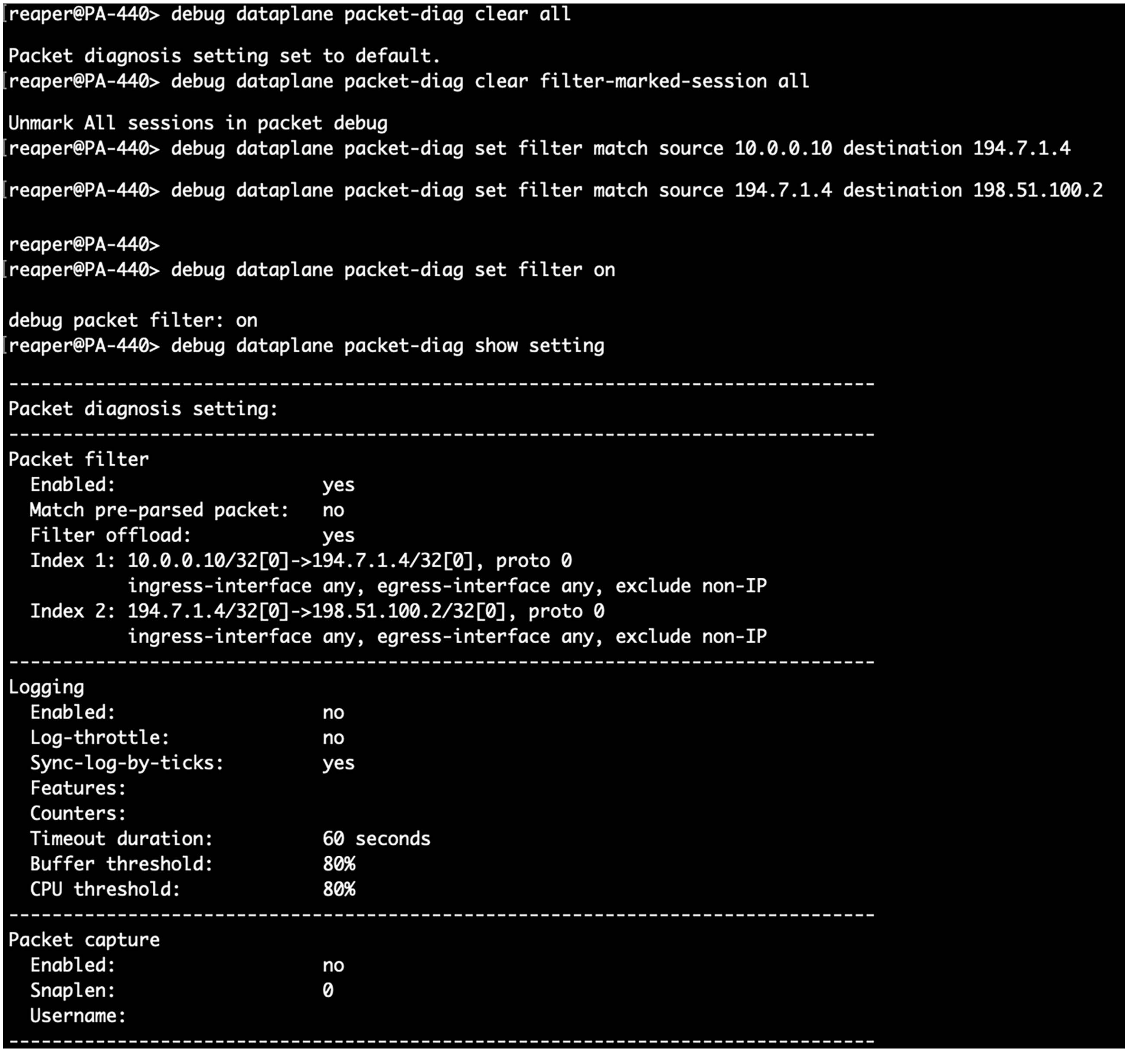
The `packet-diag` filters work by marking sessions and any packets belonging to them. This marking is then used by different processes to keep tabs on packets and to make packet capture, global counter filtering, and flow analysis possible. Any sessions previously marked by a filter will maintain this "tag" for as long as the session is active. If several filter sessions follow one another, old sessions may show up in the debug session due to them still being tagged by the previously configured filters.

Next, add all the filters and turn them on. In the following example, we have one filter from the internal IP of the host to the DNS server for outbound packets, and a returning filter for the DNS server as the source with the NAT address as the destination while the filters are session-aware. This is good practice to catch any packets that somehow manage to escape the original session. One such example could be packets arriving with such latency that the session was already closed.

The filters will use the following IP addresses: `10.0.0.10` as the internal IP of the lab server, `198.51.100.2` as the public IP used by the firewall as the NAT source for outgoing sessions, and `194.7.1.4` as the destination public IP. Adding filters for each possible direction and stage of the session will look like this:

```
reaper@PA-440> debug dataplane packet-diag set filter match source 10.0.0.10
destination 194.7.1.4
reaper@PA-440> debug dataplane packet-diag set filter match source 194.7.1.4
destination 198.51.100.2
reaper@PA-440> debug dataplane packet-diag set filter on
reaper@PA-440> debug dataplane packet-diag show setting
```

Your CLI session will look like the following screenshot:



```
reaper@PA-440> debug dataplane packet-diag clear all

Packet diagnosis setting set to default.
reaper@PA-440> debug dataplane packet-diag clear filter-marked-session all

Unmark All sessions in packet debug
reaper@PA-440> debug dataplane packet-diag set filter match source 10.0.0.10 destination 194.7.1.4

reaper@PA-440> debug dataplane packet-diag set filter match source 194.7.1.4 destination 198.51.100.2

reaper@PA-440>
reaper@PA-440> debug dataplane packet-diag set filter on

debug packet filter: on
reaper@PA-440> debug dataplane packet-diag show setting

--------------------------------------------------------------------------------
Packet diagnosis setting:
--------------------------------------------------------------------------------
Packet filter
  Enabled:                   yes
  Match pre-parsed packet:   no
  Filter offload:            yes
  Index 1: 10.0.0.10/32[0]->194.7.1.4/32[0], proto 0
           ingress-interface any, egress-interface any, exclude non-IP
  Index 2: 194.7.1.4/32[0]->198.51.100.2/32[0], proto 0
           ingress-interface any, egress-interface any, exclude non-IP
--------------------------------------------------------------------------------
Logging
  Enabled:                   no
  Log-throttle:              no
  Sync-log-by-ticks:         yes
  Features:
  Counters:
  Timeout duration:          60 seconds
  Buffer threshold:          80%
  CPU threshold:             80%
--------------------------------------------------------------------------------
Packet capture
  Enabled:                   no
  Snaplen:                   0
  Username:
--------------------------------------------------------------------------------
```

Figure 14.3: Setting up filters

We can now start looking at global counters that only relate to our filters by adding `packet-filter yes` to the global counter filter:

```
reaper@PA-440> show counter global filter delta yes packet-filter yes
```

The output of the global counters should look similar to the following:

```
reaper@PA-440> show counter global filter delta yes packet-filter yes

Global counters:
Elapsed time since last sampling: 5.801 seconds

name                                   value     rate severity  category  aspect    description
--------------------------------------------------------------------------------
pkt_recv                                   5        0 info      packet    pktproc   Packets received
pkt_stp_rcv                                5        0 info      packet    pktproc   STP BPDU packets received
flow_fwd_l3_bcast_drop                     6        1 drop      flow      forward   Packets dropped: unhandled IP broadcast
flow_fwd_l3_mcast_drop                     8        1 drop      flow      forward   Packets dropped: no route for IP multicast
flow_bcast_pkt_rcv                         9        1 info      flow      parse     IP broadcast pkt received
flow_arp_pkt_rcv                          38        6 info      flow      arp       ARP packets received
flow_arp_rcv_gratuitous                    4        0 info      flow      arp       Gratuitous ARP packets received
flow_nd_pkt_submit                         2        0 info      flow      nd        ND packets submited to process
flow_nd_pkt_rcv                            2        0 info      flow      nd        ND packets received
--------------------------------------------------------------------------------
Total counters shown: 9
--------------------------------------------------------------------------------
```

Figure 14.4: Global counters for ping

The screenshot above shows all the counters associated with the outbound ping initiated from the lab device to the public server. The global counters extracted from this session will appear as follows:

- `pkt_sent` tells us four packets were sent in the delta timeframe.
- `session_allocated` means valid sessions were set up to handle the ping request (the opening state).
- `session_installed` means the session was accepted and set to the active state.
- `flow_ip_chksm_sw_validation` is the packets for which the IP checksum was validated in the software.
- `appid_ident_by_icmp` means `App-ID` was able to identify these packets as pings immediately by their ICMP echo request signature.
- `dfa_sw` is the packets identified by `App-ID` in the software.
- `ctd_process` is the number of sessions processed by `content-ID`.
- `ctd_pkt_slowpath` is the number of packets that went through slowpath:
  - **slowpath** is the first stage of a session where packets need to be verified and matched against NAT and security rules before a session can be created. Once the session is set up, packets are processed in fastpath.
  - **fastpath** is the stage where a session has been established in the session table, so the firewall just needs to match a packet to an existing session and perform the appropriate forwarding for said session.

- `nat_dynamic_port_xlt` indicates the packets were translated by a NAT rule set to use dynamic source ports for translation. A ping will be exempted due to the dynamic nature of this type of NAT, due to the need for these types of packets to retain the same source and destination port. This can be verified by looking at the sessions, as you can see in the following screenshot:

```
reaper@PA-440> show session all filter application ping

--------------------------------------------------------------------------------
ID          Application    State   Type Flag  Src[Sport]/Zone/Proto (translated IP[Port])
Vsys                                          Dst[Dport]/Zone (translated IP[Port])
--------------------------------------------------------------------------------
70487        ping          ACTIVE  FLOW       192.168.1.122[15129]/perimeter/1  (192.168.1.122[15129])
vsys1                                         1.1.1.1[20]/perimeter  (1.1.1.1[20])
70476        ping          ACTIVE  FLOW       192.168.1.122[15129]/perimeter/1  (192.168.1.122[15129])
vsys1                                         1.1.1.1[19]/perimeter  (1.1.1.1[19])
70501        ping          ACTIVE  FLOW       192.168.1.122[15129]/perimeter/1  (192.168.1.122[15129])
vsys1                                         1.1.1.1[23]/perimeter  (1.1.1.1[23])
70483        ping          ACTIVE  FLOW       192.168.1.122[15129]/perimeter/1  (192.168.1.122[15129])
vsys1                                         1.1.1.1[18]/perimeter  (1.1.1.1[18])
70477        ping          ACTIVE  FLOW       192.168.1.122[15129]/perimeter/1  (192.168.1.122[15129])
vsys1                                         1.1.1.1[21]/perimeter  (1.1.1.1[21])
70482        ping          ACTIVE  FLOW       192.168.1.122[15129]/perimeter/1  (192.168.1.122[15129])
vsys1                                         1.1.1.1[22]/perimeter  (1.1.1.1[22])
70489        ping          ACTIVE  FLOW       192.168.1.122[15129]/perimeter/1  (192.168.1.122[15129])
vsys1                                         1.1.1.1[24]/perimeter  (1.1.1.1[24])
reaper@PA-440>
```

Figure 14.5: The show session output

In subsequent global counter outputs, all of the counters should indicate that the session is progressing as expected.

Finding issues through counters

There are many counters that indicate what is happening to packets in a session that simply show that everything is progressing as expected. There are also several counters that can help you quickly spot that something is going on that could be causing issues. Many of these will show up when you add a `severity` filter for dropped packets:

```
reaper@PA-440> show counter global filter delta yes packet-filter yes severity drop
```

There are several common counters that indicate a problem:

- `flow_tcp_non_syn` is triggered when an ACK packet is received and no TCP handshake has been established that matches the packet tuples. This is commonly an indication of asymmetric flows where only one path of the client-to-server communication passes by the firewall.

  In this case, review external routing and flow paths to determine what could be causing asymmetry.

- `flow_fw_zonechange` is another indication of asymmetric flows where returning packets are detected in a different zone, or a routing change has happened after the session was started and the destination zone is now on a different interface.

Review the routing table and policy-based forwarding rules to determine how packets could match a different zone.

- `flow_policy_deny` is when the session hits a deny policy in the security rule base. If there are no deny or drop entries in the traffic log, it is likely the session is hitting the default deny rule.
- `flow_fwd_l3_norarp` indicates that a packet can't be forwarded to the final destination because the firewall is unable to get an ARP address for the destination IP. You can review the ARP table using the following commands:

```
reaper@pa-440> show arp <interface e.g. ethernet1/1>
reaper@pa-440> show arp all
```

There are also plenty of counters that indicate whether something has happened that blocked or interrupted a session. A full list of all counters can be found at `https://github.com/PacktPublishing/Mastering-Palo-Alto-Networks-Third-Edition`.

An example of a common configuration issue can be seen in the following screenshot. It is made abundantly clear when using global counters to troubleshoot why internal hosts are unable to reach a server in the DMZ (packets are being discarded due to a LAND condition in NAT):

```
reaper@PA-VM> show counter global filter delta yes packet-filter yes

Global counters:
Elapsed time since last sampling: 2.740 seconds

name                                   value     rate severity  category  aspect    description
--------------------------------------------------------------------------------
session_allocated                          1        0 info      session   resource  Sessions allocated
session_freed                              1        0 info      session   resource  Sessions freed
flow_policy_nat_land                       1        0 drop      flow      session   Session setup: source NAT IP allocation result in LAND attack
nat_dynamic_port_xlat                      1        0 info      nat       resource  The total number of dynamic_ip_port NAT translate called
nat_dynamic_port_release                   2        0 info      nat       resource  The total number of dynamic_ip_port NAT release called
--------------------------------------------------------------------------------
Total counters shown: 5
--------------------------------------------------------------------------------
```

Figure 14.6: The packet dropped due to a LAND attack

This global counter output indicates that packets are being dropped and are ticking the `flow_policy_nat_land` counter. A LAND attack happens when the source and destination IP are identical, which could cause a loop in the system that receives these packets, causing it to reply to itself. In this case, it is caused by a misconfiguration in NAT causing the outbound packet to get translated by the Hide NAT rule, changing the source IP of the session to that of the firewall's external interface.

> Another example of this type of misconfiguration is when a Hide NAT rule is set up with **any** as the source zone. This may cause it to catch inbound sessions toward the public IP of the firewall and the source to be translated to the same IP address (inbound GlobalProtect connections, for example).

This issue can be resolved by creating a **U-Turn** NAT rule above the generic Hide NAT rule, as illustrated in the following example:

		Original Packet						Translated Packet	
	NAME	SOURCE ZONE	DESTINATION ZONE	DESTINATION INTERFACE	SOURCE ADDRESS	DESTINATION ADDRESS	SERVICE	SOURCE TRANSLATION	DESTINATION TRANSLATION
1	U-Turn	Trust-L3	Untrust-L3	ethernet1/1	any	198.51.100.2	any	dynamic-ip-and-port ethernet1/3 10.0.0.1/24	destination-translation address: 10.0.0.5
2	Inbound SSH Server	Untrust-L3	Untrust-L3	ethernet1/1	198.51.100.1	any	any	none	destination-translation address: 10.0.0.5
3	outbound hide NAT	Trust-L3	Untrust-L3	ethernet1/1	any	any	any	dynamic-ip-and-port ethernet1/1	none

Figure 14.7: A U-Turn NAT rule to prevent a LAND condition

Keep in mind that all rules are evaluated from top to bottom, so setting rules in the wrong order can have unexpected consequences. Always be mindful of source and destination zones in the original packet, and consider the effect of broad dynamic IP and port rules.

In this section, we learned how to interpret all kinds of global counters and how to go about applying filters so that the appropriate counters can be collected.

When the global counters indicate that there is an error in processing a packet, you may need to take a deep dive into the flow and look at how a session is being handled by the system. We will learn how to analyze session flows in the next section.

Analyzing session flows

We've all been there: you've reviewed the logs, collected packet captures, and looked at the global counters, but you still can't find out what exactly is happening with a session. The last resort is to look at a session one packet at a time as it goes through the firewall and see what is happening to each packet at every stage and in every process.

Inspecting the flow is a very labor-intensive task for the data plane processor to do because it now needs to write a log for each stage or process a packet goes through, for each and every packet in a flow, for all sessions that match the filter. It is paramount that a very strict filter is set, which will help prevent clutter, as well as ensuring that the data plane is not unnecessarily deprived of resources.

Use the following command while collecting the information to keep an eye on the data plane, and make sure you're not creating more issues by overloading it:

```
reaper@PA-440> show running resource-monitor second

Resource monitoring sampling data (per second):

CPU load sampling by group:
flow_lookup                    :      1%
flow_fastpath                  :      1%
```

```
flow_slowpath                   :     1%
flow_forwarding                 :     1%
flow_mgmt                       :     1%
flow_ctrl                       :     1%
nac_result                      :     0%
flow_np                         :     1%
dfa_result                      :     0%
module_internal                 :     1%
aho_result                      :     0%
zip_result                      :     0%
pktlog_forwarding               :     1%
send_out                        :     1%
flow_host                       :     1%
send_host                       :     1%
qat_result                      :     0%

CPU load (%) during last 60 seconds:
core   0   1   2   3
       *   1   1   1
       *   1   1   1
       *   1   1   1
       *   1   1   1
'------cropped for brevity, this space normally shows 60 lines, 1 for each
second-------'

Resource utilization (%) during last 60 seconds:
session:
 12  12  12  12  12  12  12  12  12  12  12  12  12  12  12
 12  12  12  12  12  12  12  12  12  12  12  12  12  12  12
 12  12  12  12  12  12  12  12  12  12  12  12  12  12  12
 12  12  12  12  12  12  12  12  12  12  12  12  12  12  12

packet buffer:
  4   4   4   4   4   4   4   4   4   4   4   4   4   4   4
  4   4   4   4   4   4   4   4   4   4   4   4   4   4   4
  4   4   4   4   4   4   4   4   4   4   4   4   4   4   4
  4   4   4   4   4   4   4   4   4   4   4   4   4   4   4

packet descriptor:
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

sw tags descriptor:
  5   6   4   3   3   5   5   2   3   7   5   2   4   6   5
  2   3   5   6   5   4   3   4   5   7   4   4   5   5   5
  3   2   5   6   5   5   5   5   3   5   4   4   5   4   5
  7   4   4   5   4   3   3   4   4   3   2   2   6   4   4
```

Be vigilant in ensuring that the data plane cores do not hit 100% consistently and that the packet descriptors, which are on-chip buffers, remain below 85%. Either of these events could mean the firewall is having trouble processing the traffic load and packet loss will occur (consider upgrading the device in this case).

Before we begin inspecting the session flows, we need to determine which processes we want to capture. There are several options available:

```
reaper@PA-440> debug dataplane packet-diag set log feature
> all         all
> appid       appid
> base        base
> cfg         cfg
> ctd         ctd
> flow        flow
> http2       http2
> misc        misc
> module      module
> pow         pow
> proxy       proxy
> ssl         ssl
> tcp         tcp
> tdb         tdb
> tunnel      tunnel
> url_trie    url_trie
> zip         zip
```

Use the following explanations of these options to find all the options needed to troubleshoot any issues. Do be careful about how many options you enable based on the filter you set. It may be wise to start small or capture individual logs in separate sessions so as not to overload your firewall:

- `all` will capture absolutely everything. This option should only be used in a lab (seriously, don't turn this on in a production environment unless you know what you're doing).
- `appid` adds capturing for the `App-ID` process.
- `base` allows deeper logging for **HA** operations.
- `cfg` helps log config changes.
- `ctd` lets you log several content engine processes, including DNS, URL, and credentials.

- `flow` includes packet processing from ingress to egress.
- `http2` can log how `http2` sessions are processed.
- `misc` includes additional services, such as the clientless VPN portal.
- `module` is used to track core engines, such as `aho`, `dfa`, and URL.
- `pow` is the scheduling of work to cores. This could be useful if there appears to be an issue with how packets are distributed among cores.
- `proxy` is the proxy processes (outbound SSL decryption and DNS proxy).
- `ssl` is the inbound SSL decryption.
- `tcp` is any additional TCP action, such as reassembly.
- `tdb` is for threat scanning.
- `tunnel` lets you look more closely at tunnel operations (such as `flow` and `ager`).
- `url_trie` is the URL-matching mechanism.
- `zip` is the unpacking process to scan inside compressed `.zip` files.

As you can see in the following screenshot, each feature (in this case, `flow`) has its own set of sub-features. These can range from child processes to overall log levels:

```
reaper@PA-440> debug dataplane packet-diag set log feature flow
  ager          ager
  all           all
  arp           arp
  basic         basic
  cluster       cluster
  fbo           fbo
  ha            ha
  log           log
  nd            nd
  np            np
  pred          pred
  receive       receive
  sdwan         sdwan
  sdwan_probe   sdwan_probe
  track         track
```

Figure 14.8: Flow features and sub-features

In many cases, a good place to start is to use `basic` (`flow basic`). Add more features if the output from `flow basic` indicates that packets are encountering an issue in a specific process.

> Similar to the filter and packet capture diagnostics, `log` has an `on` and `off` toggle:
>
> ```
> reaper@PA-440> debug dataplane packet-diag set log on
> reaper@PA-440> debug dataplane packet-diag set log off
> ```
>
> Once the on command is executed, the system starts logging at every stage that the packets matching the filter pass through, so keep an eye on the data plane CPU cores.

To run a successful log session, follow the steps outlined in the following *Preparation*, *Execution*, and *Cleanup* subsections.

Preparation

To prepare for the execution stage of the data collection effort, enter these commands in the following order to prepare filters and ensure the stage is set:

1. `debug dataplane packet-diag clear all`
2. `debug dataplane packet-diag clear filter-marked-session all`
3. `debug dataplane packet-diag set filter match <filter settings>`
4. Add up to four filters.
5. `debug dataplane packet-diag set filter on`
6. `debug dataplane packet-diag set feature flow basic`
7. Optional*: `set session offload no`
8. If there are any active sessions, it is best to clear them and hold off on creating new sessions until you are ready to start logging.

* The PA-5000, PA-5200, PA-5400, and PA-7000 platforms support hardware offloading. This happens when a session has passed all inspections and can be put into a state where it no longer needs to pass through the "slower" data plane CPU cores and can be handled by the faster network chips instead, bypassing the data plane. This means that once a session is offloaded, we can no longer collect captures or logs for the session as all this data is captured on the data plane. This is why we should turn off offloading during troubleshooting if possible, but keep in mind that this will have an impact on the data plane core usage as this is a global setting.

Execution

In this section, we're going to dive into a troubleshooting session step by step. It may help to have a session going on a real firewall while you follow along with these steps to see the output on a live system.

It can be helpful to have multiple SSH sessions open so that you can run different commands in different windows so that you don't mix outputs:

1. To clear the global counter delta, run the following command:

   ```
   Show counter global filter delta yes packet-filter yes
   ```

2. To ensure the data plane is in a healthy state before we enable logging, run the following command:

   ```
   show running resource-monitor
   ```

   Ensure the data plane load is not dangerously high before continuing to the next step.

3. Execute the following command to start the log collection:

```
debug dataplane packet-diag set log on
```

4. Initiate the session that you are troubleshooting.
5. In separate SSH windows, periodically run the following command. This allows you to maintain an easier overview:

```
> show session all filter  <appropriate filters for the sessions you are tracking>
> show counter global filter delta yes packet-filter yes
> show running resource-monitor
```

6. Once the session has ended or the issue you are trying to learn more about has occurred, wait for a few seconds to capture any "late" packets, and then turn the log off:

```
> debug dataplane packet-diag set log off
```

Because sessions are assigned to a specific core for processing and each core logs to its own file, the flow logs may be spread over multiple `pan_task_*.log` files, in `dp-log` for larger platforms, and `mp-log` for small and virtual platforms. These can be combined into a single `pan_packet_diag.log` file in the `mp-log` directory with the `aggregate-logs` command.

Once you execute the `aggregation` command, wait for a while for the logs to be merged:

```
reaper@PA-440> debug dataplane packet-diag aggregate-logs
reaper@PA-440> less mp-log pan_packet_diag.log
```

Once all the appropriate data is collected, don't forget to "clean up" after so that all information is collected and no unnecessary data is left on the firewall.

Cleanup

You can export the `pan_packet_diag.log` file, along with all the other management plane log files, so that you can analyze it in your favorite text editor:

```
reaper@PA-440> scp export log-file management-plane to user@host:/path/
```

Or, use `tftp` as an alternative if an SCP server is not available (be mindful that `tftp` is a cleartext protocol and should only be used in a secure network):

```
reaper@PA-440> tftp export log-file management-plane to host:/path/
```

After you are done, you should delete the file from your system due to its potentially sensitive content:

```
reaper@PA-440> debug dataplane packet-diag clear log log
```

This concludes the data collection. If more data is required, start again from the *Preparation* phase.

Now, let's put what we've learned into practice.

A practical example

In the following screenshot, you can see the lab layout. There is a client on a private network with the IP address 10.0.0.10, connecting to the firewall as the default gateway on 10.0.0.1. Outbound connections are source-translated behind the firewall's external IP address of 198.51.100.2.

Session 1 tries to establish an SSH connection to IP address 198.51.100.2.

Session 2 establishes an SSH connection with upstream router 198.51.100.1:

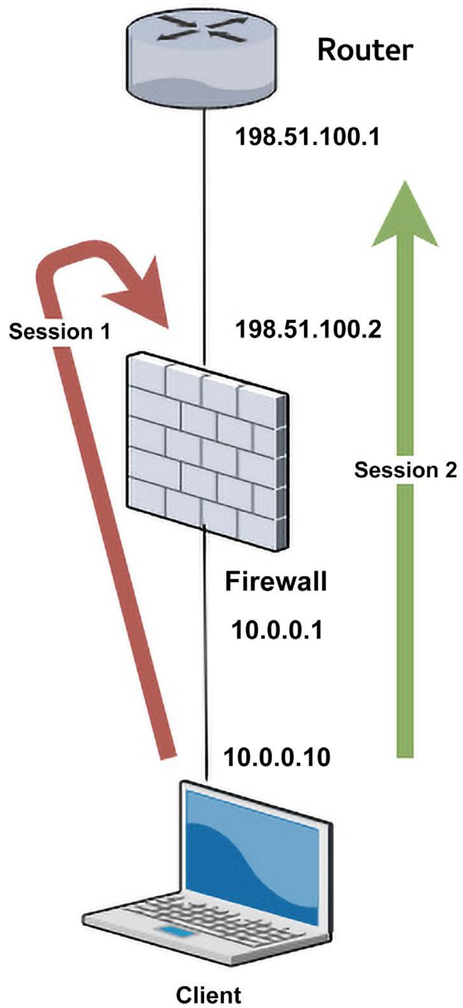


Figure 14.9: Example scenario

We prepare the troubleshooting session by clearing out all the previous filters and ensuring no marked sessions remain by clearing all the markings. We then set up the filters, disable session offloading, and prepare the log feature:

```
reaper@PA-440> debug dataplane packet-diag clear all
Packet diagnosis setting set to default.
reaper@PA-440> debug dataplane packet-diag clear filter-marked-session all
Unmark All sessions in packet debug
reaper@PA-440> debug dataplane packet-diag set filter match source 10.0.0.10
destination 198.51.100.2
reaper@PA-440> debug dataplane packet-diag set filter match source 10.0.0.10
destination 198.51.100.1
reaper@PA-440> debug dataplane packet-diag set filter match source 198.51.100.1
destination 198.51.100.2
reaper@PA-440> debug dataplane packet-diag set filter match destination
198.51.100.2
reaper@PA-440> debug dataplane packet-diag set filter on
debug packet filter: on
reaper@PA-440> show session all filter source 10.0.0.10
No Active Sessions
reaper@PA-440> set session offload no
reaper@PA-440> debug dataplane packet-diag set log feature flow basic
```

Once we're ready to get the session going, open an additional SSH window so that we can keep an eye on the data plane resources. We could influence the outcome of the test and impact other network traffic (or even the system's stability) if we overload the data plane.

It is good practice to log all your SSH sessions while troubleshooting so that you can rebuild your timeline afterward. It helps to occasionally add visual time cues to the SSH output if the troubleshooting session takes an extended length of time. You can do that by using the following command:

```
reaper@PA-440> show clock
Sun Apr  13 23:10:35 CEST 2025
```

Once we're all set to start the collection effort, enable the `log` option and clear the global counter delta:

```
reaper@PA-440> debug dataplane packet-diag set log on
Packet log is enabled
reaper@PA-440> show counter global filter delta yes packet-filter yes
Global counters:
Elapsed time since last sampling: 159.191 seconds
name                                   value     rate severity  category
aspect    description
-----------------------------------------------------------------------------
--
```

```
pkt_recv                                   2        0 info      packet
pktproc   Packets received
pkt_sent                                  14        0 info      packet
pktproc   Packets transmitted
pkt_stp_rcv                                2        0 info      packet
pktproc   STP BPDU packets received
flow_arp_pkt_rcv                           2        0 info      flow      arp
ARP packets received
flow_arp_rcv_gratuitous                    2        0 info      flow      arp
Gratuitous ARP packets received
flow_ip_cksm_sw_validation                 9        0 info      flow
pktproc   Packets for which IP checksum validation was done in software
log_pkt_diag_us                           82        0 info      log
system    Time (us) spent on writing packet-diag logs
------------------------------------------------------------------------------
--
Total counters shown: 7
------------------------------------------------------------------------------
--
reaper@PA-440> show counter global filter delta yes packet-filter yes
Global counters:
Elapsed time since last sampling: 5.588 seconds
------------------------------------------------------------------------------
--
Total counters shown: 0
------------------------------------------------------------------------------
--
```

Now that the global counter delta is cleared and logging is enabled, we can start the first session by launching an SSH session from the client at `10.0.0.10` to the firewall's external interface IP, `198.51.100.2`. We can then check to see whether a session was created and what the output of the global counters is:

```
reaper@PA-440> show counter global filter delta yes packet-filter yes
Global counters:
Elapsed time since last sampling: 24.931 seconds
------------------------------------------------------------------------------
--
Total counters shown: 0
------------------------------------------------------------------------------
--
reaper@PA-440> show session all filter source 10.0.0.10
No Active Sessions
```

```
reaper@PA-440> show counter global filter delta yes packet-filter yes
Global counters:
Elapsed time since last sampling: 13.132 seconds
name                                   value     rate severity  category
aspect    description
------------------------------------------------------------------------------
--
session_allocated                          4        0 info      session
resource  Sessions allocated
session_freed                              4        0 info      session
resource  Sessions freed
flow_policy_nat_land                       4        0 drop      flow
session   Session setup: source NAT IP allocation result in LAND attack
nat_dynamic_port_xlat                      4        0 info      nat
resource  The total number of dynamic_ip_port NAT translate called
nat_dynamic_port_release                   8        0 info      nat
resource  The total number of dynamic_ip_port NAT release called
log_pkt_diag_us                          262       19 info      log
system    Time (us) spent on writing packet-diag logs
------------------------------------------------------------------------------
--
Total counters shown: 6
------------------------------------------------------------------------------
--
reaper@PA-440> show session all filter source 10.0.0.10
No Active Sessions
```

We can see that a session has not been created and the global counters indicate that the packets were dropped.

We can now start the second session by launching an SSH session from the client at `10.0.0.10` to the upstream router at `198.51.100.1`.

This time, a session is created, and we can view the details in the command line:

```
reaper@PA-440> show session all filter source 10.0.0.10
------------------------------------------------------------------------------
--
ID          Application    State   Type Flag  Src[Sport]/Zone/Proto (translated
IP[Port])
Vsys                                          Dst[Dport]/Zone (translated
IP[Port])
------------------------------------------------------------------------------
--
```

```
270          ssh            ACTIVE  FLOW  NS   10.0.0.10[49402]/trust/6
(198.51.100.2[12607])
vsys1                                          198.51.100.1[22]/untrust
(198.51.100.1[22])
reaper@PA-440> show session id 270
Session             270
        c2s flow:
                source:      10.0.0.10 [trust]
                dst:         198.51.100.1
                proto:       6
                sport:       49402           dport:      22
                state:       ACTIVE          type:       FLOW
                src user:    unknown
                dst user:    unknown
        s2c flow:
                source:      198.51.100.1 [untrust]
                dst:         198.51.100.2
                proto:       6
                sport:       22              dport:      12607
                state:       ACTIVE          type:       FLOW
                src user:    unknown
                dst user:    unknown
        start time                           : Thu Jun  4 00:46:11
        timeout                              : 3600 sec
        time to live                         : 3589 sec
        total byte count(c2s)                : 3961
        total byte count(s2c)                : 6143
        layer7 packet count(c2s)             : 22
        layer7 packet count(s2c)             : 27
        vsys                                 : vsys1
        application                          : ssh
        rule                                 : outbound
        service timeout override(index)      : False
        session to be logged at end          : True
        session in session ager              : True
        session updated by HA peer           : False
        address/port translation             : source
        nat-rule                             : outbound hide(vsys1)
        layer7 processing                    : completed
        URL filtering enabled                : True
        URL category                         : any
```

```
        session via syn-cookies               : False
        session terminated on host            : False
        session traverses tunnel              : False
        session terminate tunnel              : False
        captive portal session                : False
        ingress interface                     : ethernet1/2
        egress interface                      : ethernet1/1
        session QoS rule                      : N/A (class 4).
        tracker stage l7proc                  : ctd decoder done
        end-reason                            : unknown
reaper@PA-440>
```

We can see that the session is active and the outbound connection is being source-NATed behind the firewall external IP of `198.51.100.1`. Packets are traveling in both directions.

We can now take a look at the global counters to verify whether everything is working as expected:

```
reaper@PA-440> show counter global filter delta yes packet-filter yes
Global counters:
Elapsed time since last sampling: 55.235 seconds
name                                   value     rate severity  category
aspect    description
------------------------------------------------------------------------------
--
pkt_recv                                   5        0 info      packet
pktproc   Packets received
pkt_sent                                  20        0 info      packet
pktproc   Packets transmitted
session_allocated                          1        0 info      session
resource  Sessions allocated
session_freed                              1        0 info      session
resource  Sessions freed
flow_policy_nat_land                       1        0 drop      flow
session   Session setup: source NAT IP allocation result in LAND attack
flow_ip_cksm_sw_validation                27        0 info      flow
pktproc   Packets for which IP checksum validation was done in software
nat_dynamic_port_xlat                      1        0 info      nat
resource  The total number of dynamic_ip_port NAT translate called
nat_dynamic_port_release                   2        0 info      nat
resource  The total number of dynamic_ip_port NAT release called
dfa_sw                                     4        0 info      dfa
pktproc   The total number of dfa match using software
ctd_sml_exit_detector_i                    1        0 info      ctd
pktproc   The number of sessions with sml exit in detector i
```

```
ctd_run_detector_i                          1        0 info      ctd
pktproc   run detector_i
ctd_sml_vm_run_impl_opcodeexit              1        0 info      ctd
pktproc   SML VM opcode exit
ctd_fwd_err_tcp_state                       1        0 info      ctd
pktproc   Forward to varrcvr error: TCP in establishment when session went away
ctd_pscan_sw                                4        0 info      ctd
pktproc   The total usage of software for pscan
ctd_pkt_slowpath                            4        0 info      ctd
pktproc   Packets processed by slowpath
log_pkt_diag_us                           303        5 info      log
system    Time (us) spent on writing packet-diag logs
------------------------------------------------------------------------------
--
Total counters shown: 16
------------------------------------------------------------------------------
--
```

Once the SSH session is ended on the client, we can verify whether the session still exists on the firewall.

Once the session is closed, we can collect a last global counter output to ensure we have all the details, and then turn off the logging feature and re-enable session offloading (if your firewall supports this):

```
reaper@PA-440> show session all filter source 10.0.0.10
------------------------------------------------------------------------------
--
ID          Application    State   Type Flag  Src[Sport]/Zone/Proto (translated
IP[Port])
Vsys                                          Dst[Dport]/Zone (translated
IP[Port])
------------------------------------------------------------------------------
--
270          ssh            ACTIVE  FLOW  NS   10.0.0.10[49402]/trust/6
(198.51.100.2[12607])
vsys1                                          198.51.100.1[22]/untrust
(198.51.100.1[22])
reaper@PA-440> show session all filter source 10.0.0.10
No Active Sessions
reaper@PA-440> show counter global filter delta yes packet-filter yes
Global counters:
Elapsed time since last sampling: 54.857 seconds
name                                   value     rate severity  category
aspect    description
```

```
--------------------------------------------------------------------------------
--
pkt_recv                                  3        0 info      packet
pktproc   Packets received
pkt_sent                                 14        0 info      packet
pktproc   Packets transmitted
flow_ip_cksm_sw_validation                9        0 info      flow
pktproc   Packets for which IP checksum validation was done in software
log_pkt_diag_us                         100        1 info      log
system    Time (us) spent on writing packet-diag logs
--------------------------------------------------------------------------------
--
Total counters shown: 4
--------------------------------------------------------------------------------
--
reaper@PA-440> debug dataplane packet-diag set log off
Packet log is disabled
reaper@PA-440> set session offload yes
```

The final step is to aggregate all the `pan_task_*.log` files into a single file:

```
reaper@PA-440> debug dataplane packet-diag aggregate-logs
pan_packet_diag.log is aggregated
```

We should wait a minute to give the firewall time to compile the aggregated file. The time needed to accomplish this may vary, depending on the size of the individual log files.

You can review the aggregated file on the firewall by using the `less` command:

```
reaper@PA-440> less mp-log pan_packet_diag.log
```

Alternatively, you can export the log files so that you can read them in Notepad++ or another text editor:

```
reaper@PA-440> scp export log-file management-plane to reaper@192.168.27.16:/
home/reaper/
```

In the resulting `tar.gz` file, `pan_packet_diag.log` is located in the `/var/log/pan` directory.

As you can see from the following screenshot, session 1 is over in just two log entries. The first paragraph shows the ingress stage where the SYN packet is first received and all of its attributes disseminated. You can use this first segment to review whether the initial SYN packet is coming in on the right interface and looks “normal.” The ingress stage will also check whether the packet matches an existing session. In our case, this is a new session, so the packet is enqueued to create a new session:

```
Packet received at ingress stage, tag 0, type ORDERED
Packet info: len 74 port 17 interface 17 vsys 1
  wqe index 33521 packet 0x0xc0013b0900, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:d7:40:22->00:0c:29:7e:38:e5, type 0x0800
IP:     10.0.0.10->198.51.100.2, protocol 6
        version 4, ihl 5, tos 0x00, len 60,
        id 41859, frag_off 0x4000, ttl 64, checksum 63842(0x62f9)
TCP:    sport 43100, dport 22, seq 3116136369, ack 0,
        reserved 0, offset 10, window 64240, checksum 24589,
        flags 0x02 ( SYN), urgent data 0, l4 data len 0
TCP option:
00000000: 02 04 05 b4 04 02 08 0a  69 3f 75 a2 00 00 00 00    ........ i?u.....
00000010: 01 03 03 07                                         ....
Flow lookup, key word0 0x600020016a85c word1 0  word2 0xa00000affff0000 word3 0x0 word4 0x26433c6ffff0000
* Dos Profile NULL (NO) Index (0/0) *
Session setup: vsys 1
No active flow found, enqueue to create session


Packet received at slowpath stage, tag 3223295891, type ATOMIC
Packet info: len 74 port 17 interface 17 vsys 1
  wqe index 33521 packet 0x0xc0013b0900, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:d7:40:22->00:0c:29:7e:38:e5, type 0x0800
IP:     10.0.0.10->198.51.100.2, protocol 6
        version 4, ihl 5, tos 0x00, len 60,
        id 41859, frag_off 0x4000, ttl 64, checksum 63842(0x62f9)
TCP:    sport 43100, dport 22, seq 3116136369, ack 0,
        reserved 0, offset 10, window 64240, checksum 24589,
        flags 0x02 ( SYN), urgent data 0, l4 data len 0
TCP option:
00000000: 02 04 05 b4 04 02 08 0a  69 3f 75 a2 00 00 00 00    ........ i?u.....
00000010: 01 03 03 07                                         ....
Session setup: vsys 1
Session setup: ingress interface ethernet1/2 egress interface ethernet1/1 (zone 1)
NAT policy lookup, matched rule index 1
Policy lookup, matched rule index 0,
Allocated new session 265.
set exclude_video in session 265 0xe03cb10780 0 from work 0xe014f40f80 0
Rule: index=1 name=outbound hide, cfg_pool_idx=1 cfg_fallback_pool_idx=0
NAT Rule: name=outbound hide, cfg_pool_idx=1; Session: index=265, nat_pool_idx=1
Packet dropped, vsys 1 NAT rule index 2 result in LAND attack, same SA/DA 198.51.100.2
```

Figure 14.10: Discarded SYN packet in flow basic

Interface 17 is the ID of the `ethernet1/2` interface, which you can check by issuing the following command:

```
reaper@PA-440> show interface all
```

Paragraph 2 is the slowpath stage. The packet will be matched against the forwarding table to determine the egress interface so a NAT lookup can take place. The egress zone is determined to be zone `1`. You can determine all the zone IDs by issuing the following command:

```
reaper@PA-440> debug device-server dump idmgr type zone all
```

A session ID of `265` is assigned to this SYN packet as it is going to turn into a session. An exact NAT rule match is found and then the NAT logic is verified; it is found that this NAT action will cause a LAND attack, so the packet is immediately discarded. This is why we couldn't see a session in the `show session` command because the session was terminated before it formed.

The second session shows a different story. In the ingress stage, we see a similar packet arriving on interface 17. A forwarding lookup is performed and the egress interface and zone are determined, followed by a NAT lookup. The NAT rule is matched and, this time, there is no conflict, so the NAT action is prepared. A session ID of 941 is assigned to this flow and enqueued to be installed:

```
Packet received at ingress stage, tag 0, type ORDERED
Packet info: len 74 port 17 interface 17 vsys 1
  wqe index 23554 packet 0x0xc0013fdf40, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:d7:40:22->00:0c:29:7e:38:e5, type 0x0800
IP:     10.0.0.10->198.51.100.1, protocol 6
        version 4, ihl 5, tos 0x00, len 60,
        id 29076, frag_off 0x4000, ttl 64, checksum 59796(0x94e9)
TCP:    sport 49404, dport 22, seq 4257280317, ack 0,
        reserved 0, offset 10, window 64240, checksum 17082,
        flags 0x02 ( SYN), urgent data 0, l4 data len 0
TCP option:
00000000: 02 04 05 b4 04 02 08 0a  3d 06 e8 fe 00 00 00 00    ........ =.......
00000010: 01 03 03 07                                         ....
Flow lookup, key word0 0x600020016c0fc word1 0  word2 0xa00000affff0000 word3 0x0 word4 0x16433c6ffff0000
* Dos Profile NULL (NO) Index (0/0) *
Session setup: vsys 1
No active flow found, enqueue to create session


Packet received at slowpath stage, tag 1688519813, type ATOMIC
Packet info: len 74 port 17 interface 17 vsys 1
  wqe index 23554 packet 0x0xc0013fdf40, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:d7:40:22->00:0c:29:7e:38:e5, type 0x0800
IP:     10.0.0.10->198.51.100.1, protocol 6
        version 4, ihl 5, tos 0x00, len 60,
        id 29076, frag_off 0x4000, ttl 64, checksum 59796(0x94e9)
TCP:    sport 49404, dport 22, seq 4257280317, ack 0,
        reserved 0, offset 10, window 64240, checksum 17082,
        flags 0x02 ( SYN), urgent data 0, l4 data len 0
TCP option:
00000000: 02 04 05 b4 04 02 08 0a  3d 06 e8 fe 00 00 00 00    ........ =.......
00000010: 01 03 03 07                                         ....
Session setup: vsys 1
PBF lookup (vsys 1) with application none
Session setup: ingress interface ethernet1/2 egress interface ethernet1/1 (zone 1)
NAT policy lookup, matched rule index 1
Policy lookup, matched rule index 0,
TCI_INSPECT: Do TCI lookup policy - appid 0
Allocated new session 941.
set exclude_video in session 941 0xe03cb3ab80 0 from work 0xe014cd2080 0
Rule: index=1 name=outbound hide, cfg_pool_idx=1 cfg_fallback_pool_idx=0
NAT Rule: name=outbound hide, cfg_pool_idx=1; Session: index=941, nat_pool_idx=1
Packet matched vsys 1 NAT rule 'outbound hide' (index 2),
source translation 10.0.0.10/49404 => 198.51.100.2/63571
Created session, enqueue to install. work 0xe014cd2080 exclude_video 0,session 941 0xe03cb3ab80 exclude_video 0
```

Figure 14.11: Accepted session in flow basic

In the next screenshot, we can see the SYN packet enter the final stage, called fastpath, which means a session has been created; the packet can egress out and the firewall is ready to receive a reply packet. In the egress stage, DSCP tags are added (if any are configured) and NAT is applied.

We also see additional information about the `layer3` routing decisions, and finally, the packet being sent out of the `ethernet1/1` interface with ID `16`:

```
Packet received at fastpath stage, tag 941, type ATOMIC
Packet info: len 74 port 17 interface 17 vsys 1
  wqe index 23554 packet 0x0xc0013fdf40, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:d7:40:22->00:0c:29:7e:38:e5, type 0x0800
IP:     10.0.0.10->198.51.100.1, protocol 6
        version 4, ihl 5, tos 0x00, len 60,
        id 29076, frag_off 0x4000, ttl 64, checksum 59796(0x94e9)
TCP:    sport 49404, dport 22, seq 4257280317, ack 0,
        reserved 0, offset 10, window 64240, checksum 17082,
        flags 0x02 ( SYN), urgent data 0, l4 data len 0
TCP option:
00000000: 02 04 05 b4 04 02 08 0a  3d 06 e8 fe 00 00 00 00    ........ =.......
00000010: 01 03 03 07                                         ....
Flow fastpath, session 941 c2s (set work 0xe014cd2080 exclude_video 0 from sp 0xe03cb3ab80 exclude_video 0)
IP checksum valid
* Dos Profile NULL (NO) Index (0/0) *
* Dos Profile NULL (NO) Index (0/0) *
2020-06-05 00:16:22.030 +0200  pan_flow_process_fastpath(src/pan_flow_proc.c:3928): SESSION-DSCP: set session DSCP: 0x00
NAT session, run address/port translation
Syn Cookie: pan_reass(Init statete): c2s:0 c2s:nxtseq 4257280318 c2s:startseq 4257280318 c2s:win 0 c2s:st 3 c2s:newsyn 0
0 plen 0
CP-DENY TCP non data packet getting through
Forwarding lookup, ingress interface 17
L3 mode, virtual-router 1
Route lookup in virtual-router 1, IP 198.51.100.1
Route found, interface ethernet1/1, zone 1
Resolve ARP for IP 198.51.100.1 on interface ethernet1/1
ARP entry found on interface 16
Transmit packet size 60 on port 16
```

Figure 14.12: SYN packet going into fastpath

The next two log entries represent the returning SYN-ACK from the upstream server.

First, we see the ingress stage, which is similar to the original outbound SYN packet, with the exception that flow `1883` was found, which is the entry in the session table that was created when the SYN packet was accepted and a session was created. The SYN-ACK is immediately forwarded to fastpath.

The second log is the packet arriving in fastpath, and a reverse forwarding lookup is taking place. Reverse NAT is applied and the ARP table is verified if the post-NAT destination can be found. The packet is then egressed back out to the client:

```
Packet received at ingress stage, tag 0, type ORDERED
Packet info: len 74 port 16 interface 16 vsys 1
  wqe index 23554 packet 0x0xc002c89380, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:7a:5e:82->00:0c:29:7e:38:db, type 0x0800
IP:     198.51.100.1->198.51.100.2, protocol 6
        version 4, ihl 5, tos 0x00, len 60,
        id 0, frag_off 0x4000, ttl 64, checksum 20966(0xe651)
TCP:    sport 22, dport 63571, seq 671986244, ack 4257280318,
        reserved 0, offset 10, window 28960, checksum 36020,
        flags 0x12 ( SYN ACK), urgent data 0, l4 data len 0
TCP option:
00000000: 02 04 05 b4 04 02 08 0a  07 57 06 99 3d 06 e8 fe    ........ .W..=...
00000010: 01 03 03 07                                         ....
Flow lookup, key word0 0x60001f8530016 word1 0  word2 0x16433c6ffff0000 word3 0x0 word4 0x26433c6ffff0000
Flow 1883 found, state 2, HA 0
Active flow, enqueue to fastpath process, type 0

* Dos Profile NULL (NO) Index (0/0) *

Packet received at fastpath stage, tag 941, type ATOMIC
Packet info: len 74 port 16 interface 16 vsys 1
  wqe index 23554 packet 0x0xc002c89380, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:7a:5e:82->00:0c:29:7e:38:db, type 0x0800
IP:     198.51.100.1->198.51.100.2, protocol 6
        version 4, ihl 5, tos 0x00, len 60,
        id 0, frag_off 0x4000, ttl 64, checksum 20966(0xe651)
TCP:    sport 22, dport 63571, seq 671986244, ack 4257280318,
        reserved 0, offset 10, window 28960, checksum 36020,
        flags 0x12 ( SYN ACK), urgent data 0, l4 data len 0
TCP option:
00000000: 02 04 05 b4 04 02 08 0a  07 57 06 99 3d 06 e8 fe    ........ .W..=...
00000010: 01 03 03 07                                         ....
Flow fastpath, session 941 s2c (set work 0xe014cd2080 exclude_video 0 from sp 0xe03cb3ab80 exclude_video 0)
IP checksum valid
* Dos Profile NULL (NO) Index (0/0) *
NAT session, run address/port translation
Syn Cookie: pan_reass(Init statete): c2s:1 c2s:nxtseq 4257280318 c2s:startseq 4257280318 c2s:win 28960 c2s:
s2c:newsyn 0 ack 4257280318 nosyn 0 plen 0
CP-DENY TCP non data packet getting through
Forwarding lookup, ingress interface 16
L3 mode, virtual-router 1
Route lookup in virtual-router 1, IP 10.0.0.10
Route found, interface ethernet1/2, zone 2
Resolve ARP for IP 10.0.0.10 on interface ethernet1/2
ARP entry found on interface 17
Transmit packet size 60 on port 17
```

Figure 14.13: Returning SYN-ACK

You may have noticed that the ingress packets have the `ORDERED` type, while the slowpath/fastpath packets are of the `ATOMIC` type. The `ORDERED` type indicates that a session is randomly assigned to a data plane core, which is common for newly ingressing packets, while `ATOMIC` means the session is assigned to a single core, which is common for established sessions.

The next two log entries represent the final ACK packet, completing the handshake:

```
Packet received at ingress stage, tag 0, type ORDERED
Packet info: len 66 port 17 interface 17 vsys 1
  wqe index 23554 packet 0x0xc0013fe900, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:d7:40:22->00:0c:29:7e:38:e5, type 0x0800
IP:     10.0.0.10->198.51.100.1, protocol 6
        version 4, ihl 5, tos 0x00, len 52,
        id 29077, frag_off 0x4000, ttl 64, checksum 61588(0x94f0)
TCP:    sport 49404, dport 22, seq 4257280318, ack 671986245,
        reserved 0, offset 8, window 502, checksum 33324,
        flags 0x10 ( ACK), urgent data 0, l4 data len 0
TCP option:
00000000: 01 01 08 0a 3d 06 e9 00  07 57 06 99                ....=... .W..
Flow lookup, key word0 0x600020016c0fc word1 0  word2 0xa00000affff0000 word3 0x0 word4 0x16433c6ffff0000
Flow 1882 found, state 2, HA 0
Active flow, enqueue to fastpath process, type 0

* Dos Profile NULL (NO) Index (0/0) *

Packet received at fastpath stage, tag 941, type ATOMIC
Packet info: len 66 port 17 interface 17 vsys 1
  wqe index 23554 packet 0x0xc0013fe900, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:d7:40:22->00:0c:29:7e:38:e5, type 0x0800
IP:     10.0.0.10->198.51.100.1, protocol 6
        version 4, ihl 5, tos 0x00, len 52,
        id 29077, frag_off 0x4000, ttl 64, checksum 61588(0x94f0)
TCP:    sport 49404, dport 22, seq 4257280318, ack 671986245,
        reserved 0, offset 8, window 502, checksum 33324,
        flags 0x10 ( ACK), urgent data 0, l4 data len 0
TCP option:
00000000: 01 01 08 0a 3d 06 e9 00  07 57 06 99                ....=... .W..
Flow fastpath, session 941 c2s (set work 0xe014cd2080 exclude_video 0 from sp 0xe03cb3ab80 exclude_video 0)
IP checksum valid
NAT session, run address/port translation
CP-DENY TCP non data packet getting through
Forwarding lookup, ingress interface 17
L3 mode, virtual-router 1
Route lookup in virtual-router 1, IP 198.51.100.1
Route found, interface ethernet1/1, zone 1
Resolve ARP for IP 198.51.100.1 on interface ethernet1/1
ARP entry found on interface 16
Transmit packet size 52 on port 16
```

Figure 14.14: Completed handshake

From this point forward, the session has been established and both sides can start exchanging their payload, as you can see in the next four log entries.

The `PSH ACK` packet is received at the ingress stage:

```
Packet received at ingress stage, tag 0, type ORDERED
Packet info: len 107 port 17 interface 17 vsys 1
  wqe index 23554 packet 0x0xc0013ff2c0, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:d7:40:22->00:0c:29:7e:38:e5, type 0x0800
IP:     10.0.0.10->198.51.100.1, protocol 6
        version 4, ihl 5, tos 0x00, len 93,
        id 29078, frag_off 0x4000, ttl 64, checksum 50836(0x94c6)
TCP:    sport 49404, dport 22, seq 4257280318, ack 671986245,
        reserved 0, offset 8, window 502, checksum 63771,
        flags 0x18 ( ACK PSH), urgent data 0, l4 data len 41
TCP option:
00000000: 01 01 08 0a 3d 06 e9 01  07 57 06 99                ....=... .W..
Flow lookup, key word0 0x600020016c0fc word1 0  word2 0xa00000affff0000 word3 0x0 word4 0x16433c6ffff0000
Flow 1882 found, state 2, HA 0
Active flow, enqueue to fastpath process, type 0

* Dos Profile NULL (NO) Index (0/0) *
```

Figure 14.15: The client PSH ACK at the ingress stage

The packet is processed through fastpath and sent out:

```
Packet received at fastpath stage, tag 941, type ATOMIC
Packet info: len 107 port 17 interface 17 vsys 1
  wqe index 23554 packet 0x0xc0013ff2c0, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:d7:40:22->00:0c:29:7e:38:e5, type 0x0800
IP:     10.0.0.10->198.51.100.1, protocol 6
        version 4, ihl 5, tos 0x00, len 93,
        id 29078, frag_off 0x4000, ttl 64, checksum 50836(0x94c6)
TCP:    sport 49404, dport 22, seq 4257280318, ack 671986245,
        reserved 0, offset 8, window 502, checksum 63771,
        flags 0x18 ( ACK PSH), urgent data 0, l4 data len 41
TCP option:
00000000: 01 01 08 0a 3d 06 e9 01  07 57 06 99                ....=... .W..
Flow fastpath, session 941 c2s (set work 0xe014cd2080 exclude_video 0 from sp 0xe03cb3ab80 exclude_video 0)
IP checksum valid
NAT session, run address/port translation
session 941 packet sequeunce old 0 new 1
Forwarding lookup, ingress interface 17
L3 mode, virtual-router 1
Route lookup in virtual-router 1, IP 198.51.100.1
Route found, interface ethernet1/1, zone 1
Resolve ARP for IP 198.51.100.1 on interface ethernet1/1
ARP entry found on interface 16
Transmit packet size 93 on port 16
```

Figure 14.16: The client PSH ACK at the egress stage

The server ACK is received at the ingress stage:

```
Packet received at ingress stage, tag 0, type ORDERED
Packet info: len 66 port 16 interface 16 vsys 1
  wqe index 23554 packet 0x0xc002c83bc0, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:7a:5e:82->00:0c:29:7e:38:db, type 0x0800
IP:     198.51.100.1->198.51.100.2, protocol 6
        version 4, ihl 5, tos 0x00, len 52,
        id 46303, frag_off 0x4000, ttl 64, checksum 31281(0x317a)
TCP:    sport 22, dport 63571, seq 671986245, ack 4257280359,
        reserved 0, offset 8, window 227, checksum 11152,
        flags 0x10 ( ACK), urgent data 0, l4 data len 0
TCP option:
00000000: 01 01 08 0a 07 57 06 9b  3d 06 e9 01                .....W.. =...
Flow lookup, key word0 0x60001f8530016 word1 0  word2 0x16433c6ffff0000 word3 0x0 word4 0x26433c6ffff0000
Flow 1883 found, state 2, HA 0
Active flow, enqueue to fastpath process, type 0

* Dos Profile NULL (NO) Index (0/0) *
```

Figure 14.17: The server ACK at the ingress stage

The server ACK packet is processed through fastpath and sent out:

```
Packet received at fastpath stage, tag 941, type ATOMIC
Packet info: len 66 port 16 interface 16 vsys 1
  wqe index 23554 packet 0x0xc002c83bc0, HA: 0, IC: 0
Packet decoded dump:
L2:     00:0c:29:7a:5e:82->00:0c:29:7e:38:db, type 0x0800
IP:     198.51.100.1->198.51.100.2, protocol 6
        version 4, ihl 5, tos 0x00, len 52,
        id 46303, frag_off 0x4000, ttl 64, checksum 31281(0x317a)
TCP:    sport 22, dport 63571, seq 671986245, ack 4257280359,
        reserved 0, offset 8, window 227, checksum 11152,
        flags 0x10 ( ACK), urgent data 0, l4 data len 0
TCP option:
00000000: 01 01 08 0a 07 57 06 9b  3d 06 e9 01                .....W.. =...
Flow fastpath, session 941 s2c (set work 0xe014cd2080 exclude_video 0 from sp 0xe03cb3ab80 exclude_video
IP checksum valid
NAT session, run address/port translation
CP-DENY TCP non data packet getting through
Forwarding lookup, ingress interface 16
L3 mode, virtual-router 1
Route lookup in virtual-router 1, IP 10.0.0.10
Route found, interface ethernet1/2, zone 2
Resolve ARP for IP 10.0.0.10 on interface ethernet1/2
ARP entry found on interface 17
Transmit packet size 52 on port 17
```

Figure 14.18: The server ACK at the egress stage

You can take a look at the full `pan_packet_diag.log` log, as well as a transcript of the troubleshooting session and a handy list of the commands, at `https://github.com/PacktPublishing/Mastering-Palo-Alto-Networks-Third-Edition`.

In this section, we learned how to collect and read logs from the data plane packet processing processes. In the next section, we'll learn how to debug other processes.

Debugging processes

Like all operating systems, both the firewall and Panorama systems have several processes that perform specific tasks. Each of these processes has a log file and a configurable logging level, also called a debug level. By default, most processes have a lowered debug level, so only the most important log entries are written to the log file, which conserves space and is better for retention. Debug levels can be increased, but this could lead to shorter retention and, in some cases, an increase in resource use.

You can verify the current debug level of each process by issuing the following command:

```
reaper@pa-440> debug <process name> show
```

For example:

```
reaper@PA-440> debug device-server show

debug level: info
  Features:
    config  : basic
```

You can change the debug level by issuing the following command:

```
reaper@pa-440> debug <process name> on <debug level>
```

For example:

```
reaper@PA-440> debug device-server on debug

debug level: debug
  Features:
    config  : basic
```

You can turn off debugging altogether by issuing the `off` attribute, but I would not recommend turning off logging. Instead, opt for the lowest level of debugging. The following debug levels can be set for most processes:

- `dump` writes everything to `log`.
- `debug` writes errors, warnings, and informational and debug logs.
- `info` writes errors, warnings, and informational logs. Some daemons use `normal` instead of `info`.
- `warn` writes warning and error logs.
- `error` only writes error messages to `log`.

When you change the debug level of a process, remember to return it to its original setting after you're done debugging.

Depending on the platform, some processes will run on the management plane, while others run on the data plane. Platforms that have multiple data planes have a copy of every data plane process on each data plane. Smaller chassis, such as the PA-800 and PA-220, and virtual systems, such as the PA-VM, only have a single plane, so all processes reside on the management plane. Panorama does not have a data plane and doesn't have data plane processes. Large platforms have an additional control plane that takes on some of the processes, such as `routed` and `mprelay`. The location of the processes also dictates where their respective logs are stored. To read logs, you can use the `grep`, `less`, or `tail` commands, followed by the log directory and the log file.

The management plane log directory is `mp-log`, the data plane log directory is `dp-log` (`dp0-log`, `dp1-log`, and so on for systems that have multiple data planes), and `cp-log` is used for systems that have an additional control plane.

For example, the authentication process (`authd`) runs on the management plane, so it can be accessed from `mp-log`:

```
reaper@pa-440> less mp-log authd.log
```

Patterns can be sought with the `grep` command to search a log file and return all relevant log entries:

```
reaper@pa-440> grep count yes mp-log authd.log pattern reaper
```

The `count` option will simply show how many lines contain the string in the pattern.

A "live" view of the logs can be enabled by using the `tail` command. The `tail` command will return the last 10 lines of a log file, and the `follow yes` parameter will start displaying the log in real time:

```
reaper@pa-440> tail follow yes mp-log ms.log
```

The following is a list of the most important management plane processes/daemons:

- `appweb3-sslvpn` is the GlobalProtect SSL web process.
- `authd` handles the authentication of users logging on to the device.
- `cryptod` takes care of encrypting and decrypting passwords and private keys for the system.
- `devsrvr` is responsible for communicating with the data plane and pushes the configuration to the data plane. It also handles URL filtering queries from the data plane.
- `ha-agent` verifies the HA status and synchronizes configuration to the HA peer.
- `ikemgr` is the ISAKMP daemon.
- `keymgr` is the IPSec key repository.
- `logrcvr` receives and writes logs forwarded by the data plane.
- `masterd` is the master process that ensures all the other processes are running. You can verify the status of all the processes with the following command, which polls `masterd`:

  ```
  reaper@pa-440> show system software status
  ```

- `management-server` is the management server, which takes care of reporting, configuration management, and distributing commits to all processes. Its log is called `ms.log`.

- `rasmgr` is the backend process for GlobalProtect.
- `routed` is the routing daemon and maintains the routing and forwarding tables and FQDN mapping. It also maintains communication with dynamic routing peers and updates the data plane network chip with routing changes.
- `satd` is the GlobalProtect process for connected satellite devices.
- `sslmgr` performs OCSP and CRL operations and maintains a repository.
- `sysd` manages communication between processes.
- `useridd` maintains communication with the user ID agents.
- `varrcvr` is used to receive PCAP files for threats from the data plane and for WildFire logs. It also processes log forwarding to Panorama and syslog.

The data plane processes are as follows:

- `brdagent` configures and monitors interfaces and networking chips.
- `pan_comm` is the data plane partner process to the device server. It receives `commit` jobs.
- `mprelay` communicates with `routed` (route daemon) to receive routing updates and perform tunnel monitoring. It maintains the forwarding table and brings tunnels up or down.
- `pan_dha` performs HA link/path monitoring.
- `pan_task_*` are the packet forwarding daemons. Each packet processing CPU core runs a `pan_task` process. `pan_task` is a pre-spun-up process, so it will show as using 100% CPU, even when the system is at minimum load.
- `Sysdagent` monitors the data plane and communicates with `sysd` on the management plane.

You can now troubleshoot sessions using global counters and you have learned how to increase the debug level and access log files for specific daemons. In the last section, we will go over some of the most useful CLI commands that can help troubleshoot issues to make your life a lot easier.

CLI troubleshooting commands cheat sheet

There is plenty of information that you can get from reading logs, but there are many commands that will simplify the search for information by providing the required information directly. In the tables in this section, I have tried to group some of the more interesting commands for you to manage your systems. Unless stated otherwise, all commands are in **operational mode**.

The first set of commands are generally useful commands:

Command	Function
`find command keyword <keyword>`	Lets you find any command as long as you know what you're looking for
`| match <value>`	Filters the output of a command and only returns the line that has a positive match

`| except <value>`	Filters the output of a command and returns everything except the lines that match the value
`tcpdump snaplen 0 filter "not port 22"`	Captures all sessions on the management interface except sessions on port 22
`view-pcap debug-pcap|filter-pcap|mgmt-pcap no-dns-lookup`	Shows packet captures taken on daemons via `packet-diag` or `tcpdump`
`show admins`	Shows currently logged-in admins
`delete admin-sessions username <user>`	Terminates an admin's session
`set system setting target-vsys <vsys>`	Changes operational commands to a `vsys` perspective
`show authentication allowlist`	Shows the allow list for all authentication profiles
`show system environmentals`	Shows system core temperatures and power levels
`scp|tftp export <thing> to user@destination:/path/`	Exports things from the system, such as log files, packet captures, or core files

Table 14.1: Generally useful commands

The next set provides basic information about the system:

Basic system information	**Function**
`show system info`	Returns basic device information such as serial, IP, installed content, and software versions
`show system software status`	Shows whether all processes are running properly
`show system logdb-quota`	Returns the LogDB usage
`show system disk-space`	Returns disk volume information
`show jobs all/id`	Returns the status of all commit, download, install, and GFDN jobs, and additional details on specific IDs

`show system files`	Shows whether any core dump files have been created due to a process crash
`request license fetch/info`	Retrieves and shows currently active licenses
`show netstat all yes`	Shows all listening and established connections on the management plane, per process
`show chassis-ready`	Shows whether the data plane is ready to process sessions
`show panorama-status`	Verifies connectivity with Panorama

Table 14.2: System information commands

With the following commands, you will be able to verify and control HA modes and make sure the cluster is operating optimally:

HA commands	Function
`show high-availability state`	Shows a quick rundown of the local peer's HA condition
`show high-availability all`	Shows a summary of all HA runtimes
`show high-availability state-synchronization`	Displays statistics about sent and received sync messages
`request high-availability sessions-reestablish force`	Re-establishes the HA1 link if the link was lost; use `force` if HA1 backup is not configured
`show high-availability session-reestablish-status`	Shows when HA1 and HA1-backup links were last re-established
`request high-availability sync-to-remote running-config`	Manually syncs running configuration to peer, in case automatic sync failed or if the status is out of sync
`request high-availability state functional|suspend`	Suspends or activates the local device

`request high-availability state peer functional|suspend`	Suspends or activates the peer device
`show high-availability transitions`	Indicates how many times a device has transitioned between HA states
`show high-availability flap statistics`	Details about pre-emption "flaps" (preemption activates device, error encountered again, device non-funct, recovers, preempt activates, error encountered again, etc.)
`show high-availability control-link statistics`	Detailed information about HA1 messages

Table 14.3: HA commands

The following commands will tell you more about how the system is performing:

Performance information	**Function**
`show system resources`	Shows management plane resource usage, similar to `top` in Linux
`show running resource-monitor`	Shows data plane CPU core utilization and buffer usage
`debug dataplane pool statistics`	Shows software buffer pool usage
`show session info`	Shows number of active sessions, packets per second, throughput, and other session-related parameters
`debug log-receiver statistics`	Information on log volume per second and any errors while writing or forwarding logs
`show system statistics application|session`	Shows live statistics about top applications or system throughput
`show report jobs`	Indicates whether reports are currently being generated (this could have an impact on management plane CPU usage)

Table 14.4: Performance-related commands

The DNS proxy is responsible for a couple of important functions within the system. These commands help you check whether the DNS resolution is working as expected:

DNS operations	Function
`show system setting ssl-decrypt dns-cache`	Shows the SSL decryption DNS cache
`show dns-proxy cache all`	Shows the DNS proxy cache
`show system setting ssl-decrypt memory`	Shows the SSL decryption memory usage
`show dns-proxy fqdn all`	Shows all FQDN objects with their resolved IP addresses
`request system fqdn refresh`	Refreshes all FQDN objects
`debug dataplane internal vif link`	Returns statistics on the internal hardware interfaces

Table 14.5: DNS proxy commands

The following commands will help you verify whether sessions are running into unexpected configurations or other issues:

Packet flow commands	Function
`show counter global filter delta yes`	Shows global counters
`show session all filter <filters>`	Shows active, discard, and predict sessions matching the filter (or `all` sessions)
`set session offload yes|no`	Enables and disables session offloading to hardware
`set session  tcp-reject-non-syn yes|no`	Disables dropping TCP ACK packets coming in without a proper handshake
`# set deviceconfig setting tcp asymmetric-path bypass|drop`	Disables dropping packets that arrive out of the window or out of sync

Table 14.6: Packet flow commands

The next set of commands lets you verify routing, routing protocol, and MAC and ARP information:

<table>
<tr><th>OSI layers 2 and 3 commands</th><th>Function</th></tr>
<tr><td><code>show routing route</code></td><td>Outputs the routing table (routing information base, or RIB)</td></tr>
<tr><td><code>show routing fib</code></td><td>Shows the forwarding table (forwarding information base)</td></tr>
<tr><td><code>show arp all</code></td><td>Shows the content of the ARP table (layer 3)</td></tr>
<tr><td><code>show mac all</code></td><td>Shows the content of the MAC table (layer 2)</td></tr>
<tr><td><code>show routing protocol ospf|bgp|rip summary</code></td><td>Returns a summary of the OSPF, BGP, or RIP status</td></tr>
<tr><td><code>show routing resource</code></td><td>Verifies that the number of routes is not reaching the system limits</td></tr>
<tr><td><code>debug routing pcap ospf|bgp|rip on|off</code></td><td>Enables/disables packet captures on the routing engine for the routing protocol; use for troubleshooting only</td></tr>
</table>

Table 14.7: Layer 2 and layer 3 information

NAT, QoS, and zone/DoS protection depend on memory pools. The following commands help you verify that the system isn't being oversubscribed:

Policy commands	Function
`show running nat-policy`	Shows all active NAT rules
`show running nat-rule-ippool rule <rulename>`	Shows memory usage, oversubscription ratio, and allocations per rule
`show running global-ippool`	Shows runtime statistics for global dynamic source NAT
`show running ippool`	Shows overall source NAT statistics
`show session all filter qos-class [1-8]`	Displays all sessions that match a specific QoS class
`show qos interface <interface> counter`	Shows general counter on QoS configured on an interface

`show qos interface <interface> throughput <Qid as seen in counters>`	Returns actual throughput for a QID on an interface
`show zone-protection zone <zone>`	Shows zone protection statistics for the zone
`show dos-protection rule <rulename> statistics`	Shows statistics for a DoS-protection rule
`show dos-protection zone <zone> blocked source`	Shows which IP addresses are currently being blocked due to DoS protection

Table 14.8: Memory pool used by rules

URL filtering uses a data plane cache to store the most popular and most recently visited URLs. The management plane holds a larger cache of the most popular URLs. Initially, the cloud seed file is used to populate the management plane cache with the most popular URLs per region and, over time, the cache will start to retain the URLs most commonly used within your organization. When a URL is accessed that is not known to the data plane cache, a lookup is performed on the management plane cache. If the management plane does not have an entry for the URL, a cloud lookup will be performed. The following commands help you manage and maintain these caches:

URL filtering	Function
`test url-info-cloud <url>`	Shows the category for a URL via cloud lookup
`test url-info-host <url>`	Shows the category for a URL in the management plane cache
`show running url`	Shows the category for a URL in the data plane cache
`request url-iltering update url <url>`	Refreshes the management plane cache entry for a URL with a cloud lookup
`show running url-cache all`	Outputs the URL cache to `mp-log dp_url_DB.log`
`show running url-cache statistics`	Shows memory usage of the URL cache
`show url-cloud status`	Returns connectivity information for URL lookup cloud connection
`clear url-cache all|url <url>`	Clears a single URL from the cache, or the entire cache from the data plane
`delete url-database all|url`	Clears a single URL from the cache, or the entire cache from the management plane

Table 14.9: URL filtering commands

Panorama has a few unique commands that can assist in troubleshooting log forwarding from firewalls:

Panorama commands	Function
`show logging-status device <serial>`	Returns log forwarding information for a device logging to Panorama
`debug log-collector log-collection-stats show incoming-logs`	Shows incoming log statistics, including the current log rate
`show system raid detail`	Shows information on the RAID array on an M appliance
`show system disk details`	Shows information on the disk status on a VM appliance
`replace old <serial> new <serial`	Replaces a managed device's serial with a new one after an RMA. This loads all the configuration previously associated with one device with a new one without needing to go in and assign a configuration to the new serial (it removes the old serial)
`request log-fwd-ctrl action latest|start-from-lastack device <serial>`	Starts log forwarding from a device from the last log/last acked log
`request log-fwd-ctrl start|stop latest device <serial>`	Starts or stops log forwarding from a device to Panorama with buffering
`request log-fwd-ctrl action live device <serial>`	Starts log forwarding without buffering (this could cause a large flood of inbound logs)

Table 14.10: Panorama commands

Here are a few commands that are useful when troubleshooting IPSec phase 1 and phase 2 issues:

Commands useful for IPSec	Function
`show running tunnel flow info`	Shows basic statistics about all VPN tunnels
`test vpn ike-sa gateway <gateway>`	Initiates an IKE negotiation with the designated gateway
`test vpn ipsec-sa tunnel <tunnel>`	Initiates an IPSec negotiation for the designated tunnel
`clear vpn ike-sa gateway <gateway>`	Clears the **IKE SA** (**IKE Security Association**) for a given gateway
`clear vpn ipsec-sa tunnel <tunnel>`	Clears the **IPSec SA** (**IPSec Security Association**) for a given tunnel
`show vpn ike-sa gateway <gateway>`	Shows the IKE SA for a given gateway
`show vpn ipsec-sa tunnel <tunnel>`	Shows the IPSec SA for a given tunnel

`show global-protect-gateway current-satellite`	Shows currently connected satellites to GlobalProtect
`show global-protect-gateway current-user`	Shows currently connected users to GlobalProtect

Table 14.11: IPSec troubleshooting commands

User identification has many facets, from user-to-IP mapping to group mapping. The following commands will help verify whether all the information is being collected properly:

User-ID commands	Function
`show user ip-user-mapping all|ip`	Shows all mapped users or the mapped user(s) for a specific IP on the data plane
`show user ip-user-mapping-mp all|ip`	Shows all mapped users or the mapped user(s) for a specific IP on the management plane
`debug user-id refresh group-mapping all`	Refreshes group mapping memberships
`show user group list`	Shows all groups used in group mapping
`show user group name <group>`	Shows all members of a group
`show user group-mapping state all`	Shows the state of all group mapping profiles
`show user group-mapping statistics`	Shows the last/next refresh of group mapping
`show user user-id-agent statistics|state all`	Shows agent state and statistics
`show user ts-agent statistics|state all`	Shows terminal server agent state and statistics
`show user server-monitor statistics|state all`	Shows the state of the agentless user ID agent
`show user ip-port-user-mapping all`	Shows user-to-port mapping for terminal server agents or a specific server IP

Table 14.12: User-ID troubleshooting commands

There are a few useful commands to verify whether WildFire is working as expected:

WildFire commands	Function
`show wildfire status`	Shows connection status to WildFire cloud
`show wildfire statistics`	Shows file transfer statistics
`test wildfire registration`	Tests connectivity to WildFire cloud

Table 14.13: WildFire

The following are a few useful commands to take control of DHCP on the firewall:

<table>
<tr><th>DHCP commands</th><th>Function</th></tr>
<tr><td>show dhcp server lease all</td><td>Shows all DHCP leases</td></tr>
<tr><td>clear dhcp lease interface <interface> ip|mac|expiredonly <value></td><td>Clears a lease for an IP or MAC address, or all the expired ones</td></tr>
<tr><td>debug dhcpd pcap on|off</td><td>Enables packet capture of DHCP transactions on the daemon</td></tr>
<tr><td>show dhcp client state <interface></td><td>Shows DHCP information for an interface, that is, the DHCP client</td></tr>
<tr><td>request dhcp client release|renew <interface></td><td>Releases or renews the DHCP client lease for a DHCP client interface</td></tr>
</table>

Table 14.14: DHCP commands

The following commands are extremely versatile and let you extract just about any details from the system. They help determine what limits the system has, the memory addresses, the temperatures, the fan speeds, all of the configuration elements, the interface states, and even what kind of fiber optic transceiver is installed:

<table>
<tr><th>Device state super commands</th><th>Function</th></tr>
<tr><td>show system state</td><td>Returns the state of the entire device</td></tr>
<tr><td>show system state filter env.*</td><td>Shows system core temperatures and power levels</td></tr>
<tr><td>show system state | match fan</td><td>Searches the system state for any line containing fan to find fan speeds</td></tr>
<tr><td>show system state | match cfg.general.max</td><td>Returns the maximum number of configurable objects the system supports</td></tr>
<tr><td>show system state filter-pretty sys.s1.*</td><td>Shows information about all the interfaces in slot 1</td></tr>
</table>

Table 14.15: Device state

All the preceding commands can also be accessed from `https://github.com/PacktPublishing/Mastering-Palo-Alto-Networks-Third-Edition`.

The CLI commands we learned about in this section will help troubleshoot and debug most issues you will encounter. Make sure you keep the cheat sheet close by and, when in doubt, remember to fall back on `find command keyword` as this has saved me numerous times.

Summary

In this chapter, you learned how to use global counters to find out what is happening to a session and how to interpret the output. You are now able to collect deep-dive logs for each process that touches a session and should be able to add additional logging to suit explicit scenarios. You should also be able to organize a troubleshooting session efficiently so that you can get to the root cause of an issue much more quickly than you would have done before. The cheat sheet of CLI commands provided here should come in handy to collect any additional information.

If you're preparing for the PCNSE, remember that each plane has its own processes, and logs can be found in the plane's log directory.

In the next chapter, we will look at how to deploy firewalls in a cloud environment, and some of the unique considerations that come up when setting them to protect resources.

15

Cloud-Based Firewall Deployment

In this chapter, we are going to take a look at how deploying a firewall in a cloud environment differs from deploying a firewall on-premises. We will look into some key differences and things to be mindful of when working with cloud firewalls.

We will focus mainly on Azure for its ease of use and availability. Other cloud vendors, like **Amazon Web Services** (**AWS**) and **Google Cloud Platform** (**GCP**), will have similar procedures, but the details will vary.

In this chapter, we're going to cover the following main topics:

- Licensing a cloud firewall
- Deploying a firewall in Azure
- Bootstrapping a firewall
- Putting the firewall in line
- Setting up a load balancer

By the end of this chapter, you'll be able to set up a firewall in a cloud environment to protect virtual assets.

Technical requirements

This chapter will demonstrate several ways to deploy a firewall in a cloud environment. It is recommended to have a cloud subscription available with your preferred provider, but most providers offer a free trial or free credits for a limited time for you to test their environment.

> Remember that spinning up resources in a cloud environment will start the clock on billing. If you set up a lab environment, don't forget to break it down (shut down and then delete the resource group, storage account, etc.) afterwards to prevent incurring additional costs.

We recommend you set up a subscription or create a trial account with Microsoft Azure, as we will be focusing on Azure mostly in this chapter: `https://portal.azure.com/`. This will allow you to follow along with all the topics covered in the chapter.

Licensing a cloud firewall

Firewalls deployed in a datacenter typically get booted up and will run for years, but in a cloud setting, a firewall may serve only a very temporary purpose. Resources, like additional web servers, may only be spun up during certain times of the day to handle additional capacity, and then get spun down as the load diminishes. So, too, can cloud firewalls be spun up to provide more capacity and shut down once the need dies down.

Typical licensing is charged on a yearly basis and doesn't consider how many hours or processing cycles are consumed by the firewall. In cloud environments, the licensing model can be tailored to better suit the actual usage of the firewall.

At the time of writing, there are 2 different flavors available from the marketplace:

- **VM-Series Next Generation Firewall**: This is the traditional virtual machine firewall. It is completely managed like a hardware firewall and is licensed using Flex credits or "Pay as You Go" ("PayGo"). For flex credits, there is a base cost of credits associated with the firewall system itself that depends on the number of vCPUs active on the firewall that determine the capacity of the platform, and additional subscription licenses. For example, a single VM firewall with 2 vCPUs, threat prevention, URL filtering, WildFire, and DNS security requires 16,19 credits right now, which is rounded up to 17. Palo Alto provides a credit estimator here: `https://www.paloaltonetworks.com/resources/tools/ngfw-credits-estimator`.
- **Cloud Next-Generation Firewall by Palo Alto Networks**: This is a SaaS ("Software as a Service") that is a managed service with far less input, but also less overhead, for customers. This model uses "PayGo" licensing, which does not require credits but is instead calculated based on actual compute cycles, i.e., pricing is determined by how much compute time is used rather than a flat rate for features over a yearly period.

For this chapter, we'll focus mostly on the VM flavor, as that still requires the same level of attention from the administrator as regular firewalls.

In the following screenshot, you'll see the options available when creating a new instance of this firewall from the Azure Marketplace:

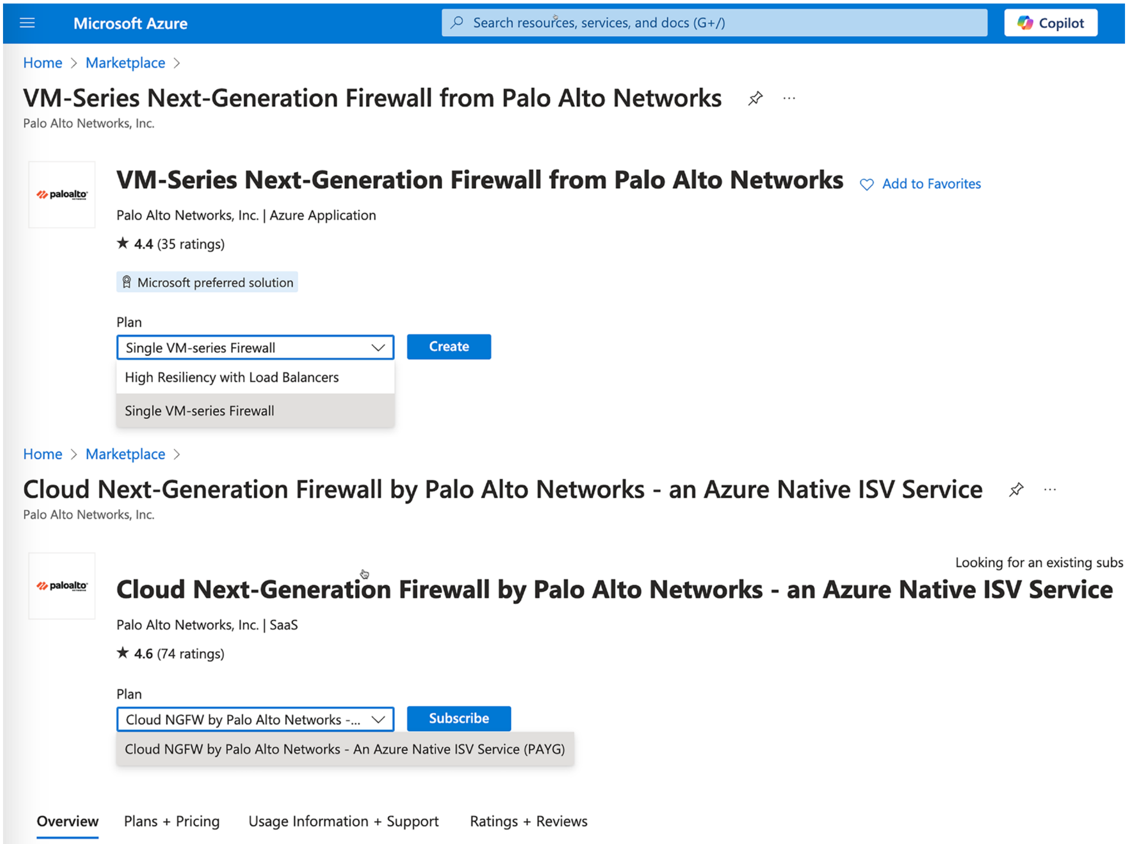


Figure 15.1: Microsoft Azure VM series licensing options

We will take a look at the VM-Series Next Generation Firewall (NGFW) plan options in the next section.

If you search the Marketplace in AWS, you will notice the experience is very similar to Azure, but there is no **Bring Your Own License** (**BYOL**) option available at this time. You can instead deploy one of the two available PAYG (Pay as You Go) bundles, as you can see in the next screenshot. The first only has the Advanced Threat Prevention subscription; the second one has all the subscriptions:

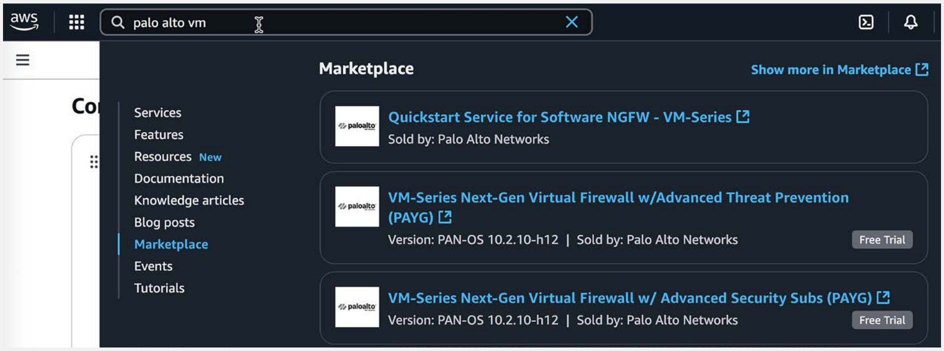


Figure 15.2: AWS VM series licensing options

If you have purchased flex credits, you can create an instance using the VM flavor in Azure, and once the setup is complete, register the firewall just like any other VM using the **CPUID** and **universally unique identifier** (**UUID**).

We'll take a look at the PayGo bundle and installation process in Azure in the next section.

Deploying a firewall in Azure

Very differently from a locally deployed VM firewall, you do not need to find an appropriate installation package from the Palo Alto support portal download page, but instead need to go through the Azure Marketplace.

From the home screen, type `marketplace` in the search bar and click **Marketplace** when it appears in the search results.

Once in the Marketplace, search for `palo alto vm` and select **VM-Series Next-Generation Firewall from Palo Alto Networks** from the available options, as illustrated in the next screenshot:

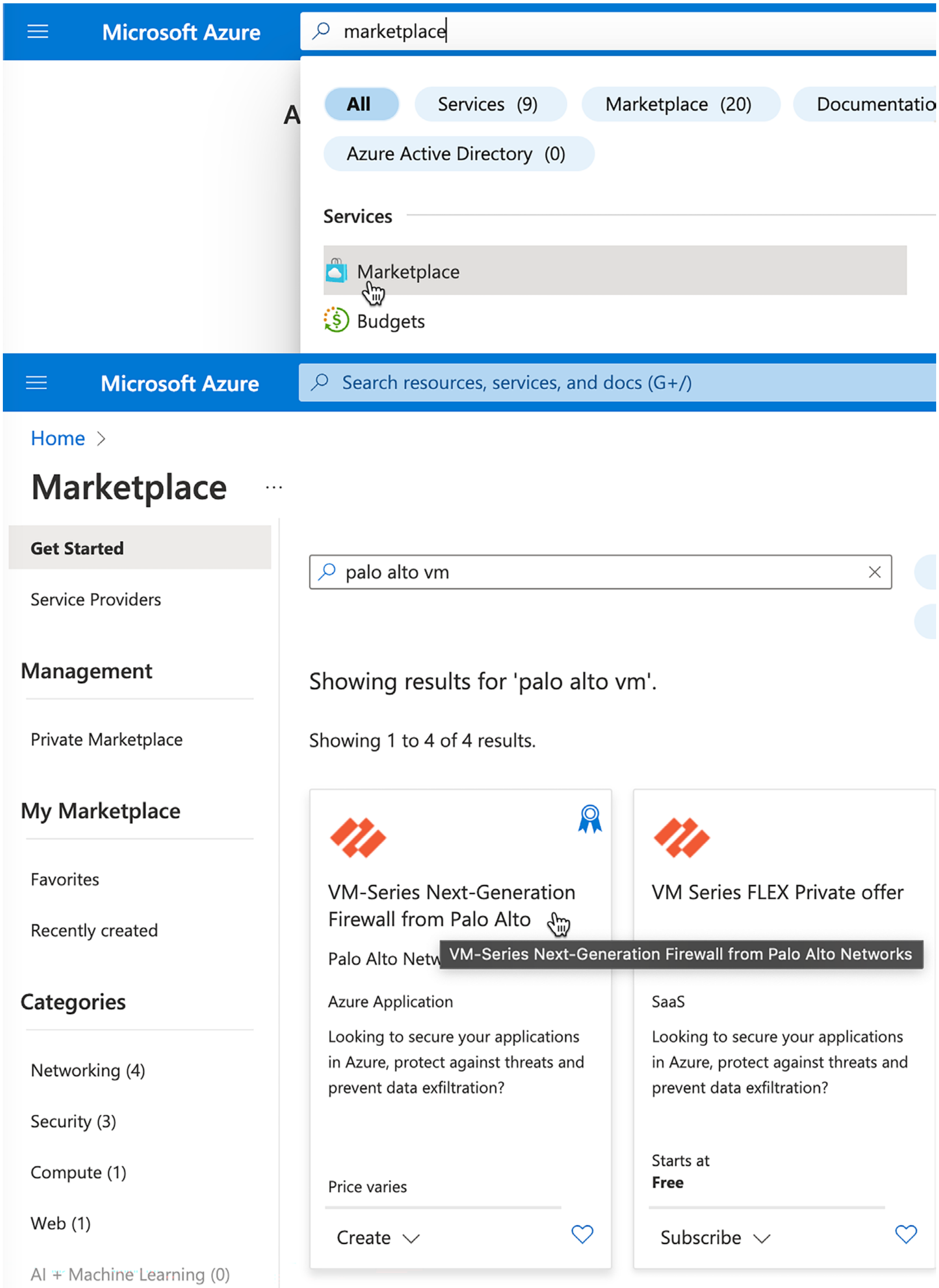


Figure 15.3: Selecting the VM series NGFW

On the next screen, you will be presented with the plan options:

- **Single VM-series Firewall** is just a standalone firewall
- **High Resiliency with Load Balancers** is a set of 2 firewalls that get deployed with load balancers in one of the following constellations:
  - **Common Firewall set with Load Balancers** is a "load balancer sandwich" with the firewalls in between an inbound and outbound load balancer
  - **Dedicated Inbound Firewall set with ILB** places the firewalls behind an **inbound load balancer** (**ILB**)
  - **Dedicated Outbound and East-West Firewall set with ILB** places both firewalls in front of an outbound load balancer
  - **Dedicated Inbound, Outbound and East-West Firewall set with Public LB and ILB** deploys a set of 4 firewalls with 2 fronted by a public load balancer for inbound traffic and 2 fronted by an internal load balancer

For the following examples, I will select **Single VM-series Firewall**. Regardless of the bundle selection, the next steps will all be identical. Select the appropriate plan and click **Create**:

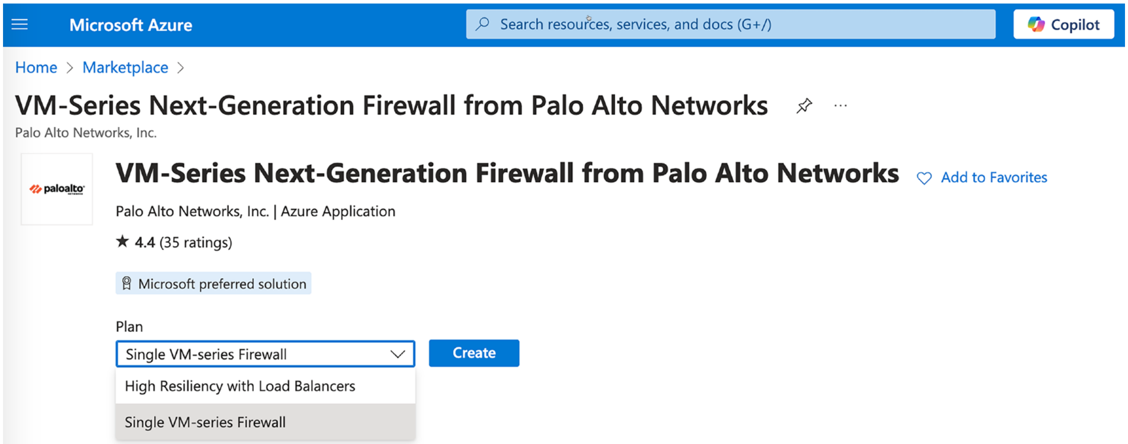


Figure 15.4: Microsoft Azure VM series licensing options

After selecting the appropriate plan and clicking **Create**, we are taken to the next page, where we need to select a subscription. This is where the cost of running the VM will be charged; some organizations may have multiple subscriptions to differentiate between cost centers, or production from lab costs.

You will also need to select or create a resource group. A resource group is a logical grouping of different resources that may include hosts and network segments. You can only deploy one firewall (from the Marketplace) per resource group.

I'm creating a new resource group as this is a fresh deployment, but you could add a firewall to an existing resource group if there are already some servers running. If you select an existing resource group, the **Region** option will disappear, but if you create a new one, you must select a region for the resource group to be created in. This will determine in which compute location your virtual resources will be created and may have an impact on latency.

Next, we create a username and password for the first administrator to be configured on the NGFW. We also need to select a license type. This is where you choose to deploy the firewall so that you will provide the licenses through flex credits (BYOL) or use one of the predefined PayGo bundles (no additional licenses required, but slightly more expensive). You can also add a tag to all the resources in the deployment.

Your screen should look similar to the following screenshot:

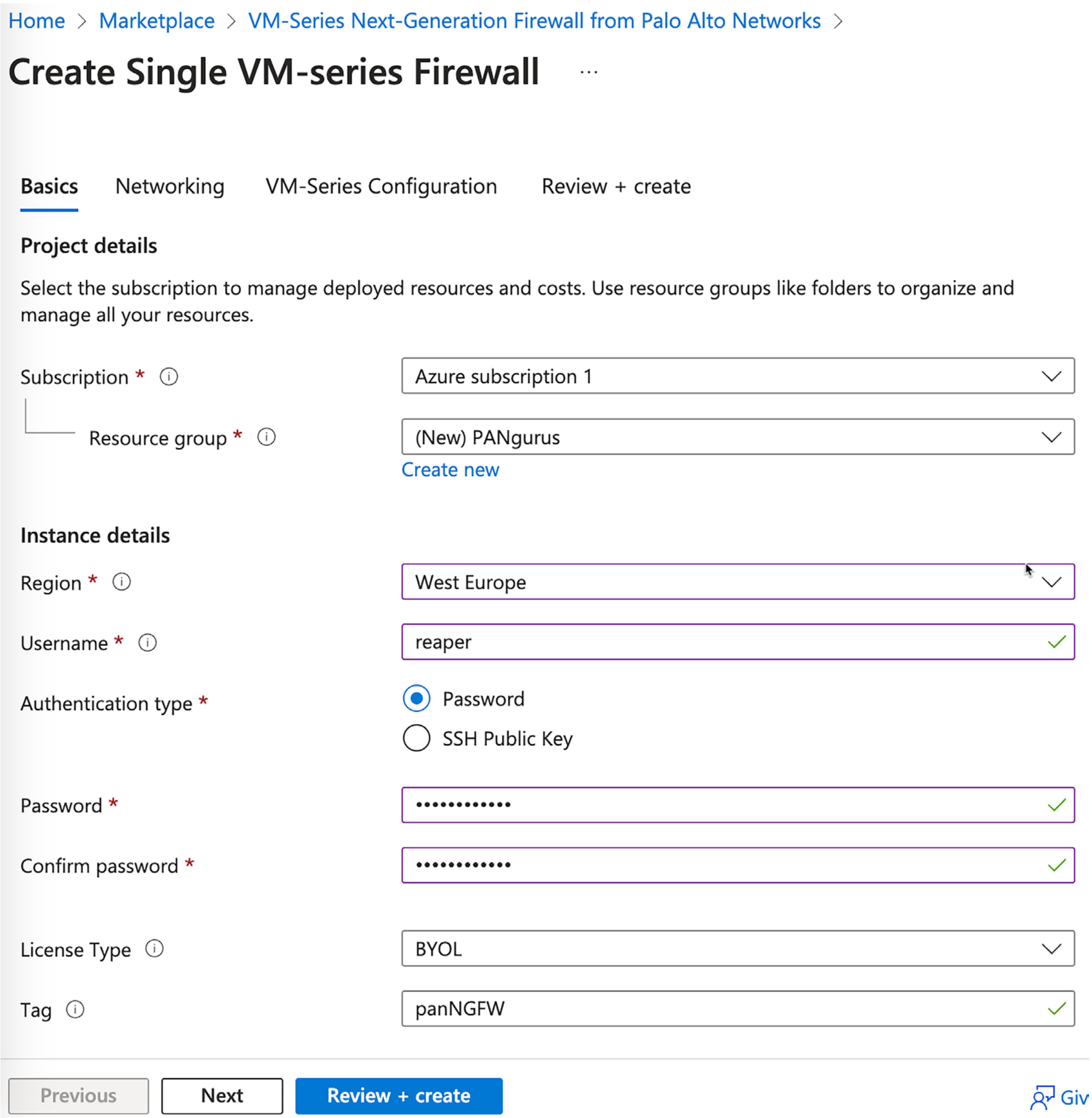


Figure 15.5: Basics configuration tab

Selecting **Review + create** at this point would fast-track you to the final stage, accepting all default settings, but will likely error out as you still need to assign IP addresses. Instead, select **Next** to get to the **Networking** tab when everything is filled out appropriately so that we can review what those default settings are and make changes where appropriate.

The next step is to determine the subnets associated with the new resource group (if you select an existing resource group, the subnets associated with that resource group will show up, and you need to select the appropriate one to match each interface).

There are two **Deployment Architecture** options available:

- **Two Arm - Management + Private + Public** provides a public interface and a private one
- **Single Arm - Management + Private** only provides a private interface

The default setting (**Two Arm**) creates three 172.16.x.0/24 subnets and assigns them in order to the **Management**, **Public**, and **Private** subnets. You can switch them around if you like, but you can't put two interfaces in the same subnet.

While the **Two Arm** architecture presents a traditional "pass through" design, the **Single Arm** architecture can be used to segregate internal hosts from each other by forcing the routing (by means of a **User Defined Route** (**UDN**) on the subnet) through the firewall.

By default, the **Network Security Group** (**NSG**), which is a built-in mini-firewall for the resource group that protects your management interface, allows 0.0.0.0/0 inbound access. If you are on a static IP or subnet, this is a good time to add that subnet to the NSG inbound source IP, as this will limit who will have access to the management interface of your firewall from the internet.

> You can use a service like `https://whatismyip.com` to find your public IP, so you can add that to the NSG.

The **Networking** settings will look similar to the following screenshot:

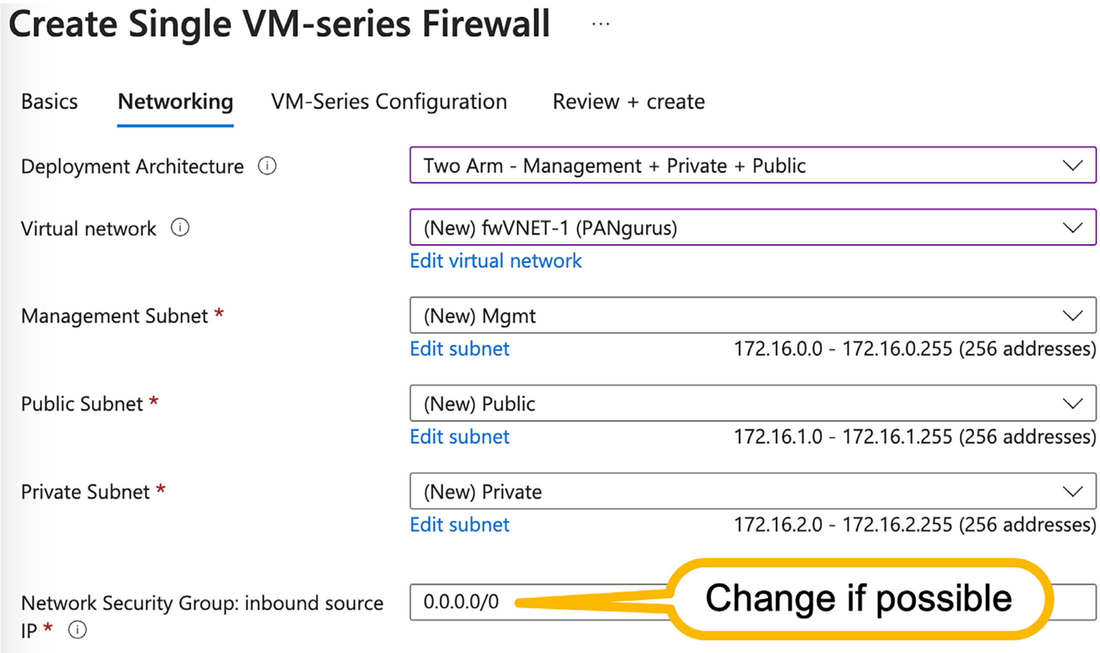


Figure 15.6: Networking tab

Click **Next** to get to the next tab.

In the **VM-Series Configuration** tab, you will be given a public IP, which will be assigned to the management interface of your firewall through inbound NAT. If your resource group has public IP addresses assigned to it, you can also select one of those. You can also create a friendly **fully qualified domain name** (**FQDN**) associated with the compute location of your resource group to access the management interface.

Next, pick a name for the firewall. All resources associated with the firewall will be prefixed by this name, so make sure you choose something that can easily be identified in case you later need to make changes. You can also choose which PAN-OS version the firewall will first be deployed in. At the time of writing, the available options are **11.2.5**, **11.1.203**, **11.0.401**, and **10.2.901**.

You can choose to bootstrap the firewall, which we will cover in the next section, so select **No bootstrap** for now.

Lastly, you can determine the size of the firewall, which will influence the capacity, throughput, and price of the VM. For reference, you can find all the VM series models and their capacity details here: `https://docs.paloaltonetworks.com/vm-series/10-2/vm-series-performance-capacity/vm-series-performance-capacity/vm-series-on-azure-models-and-vms`.

Your **VM-Series Configuration** tab should look similar to the following screenshot:

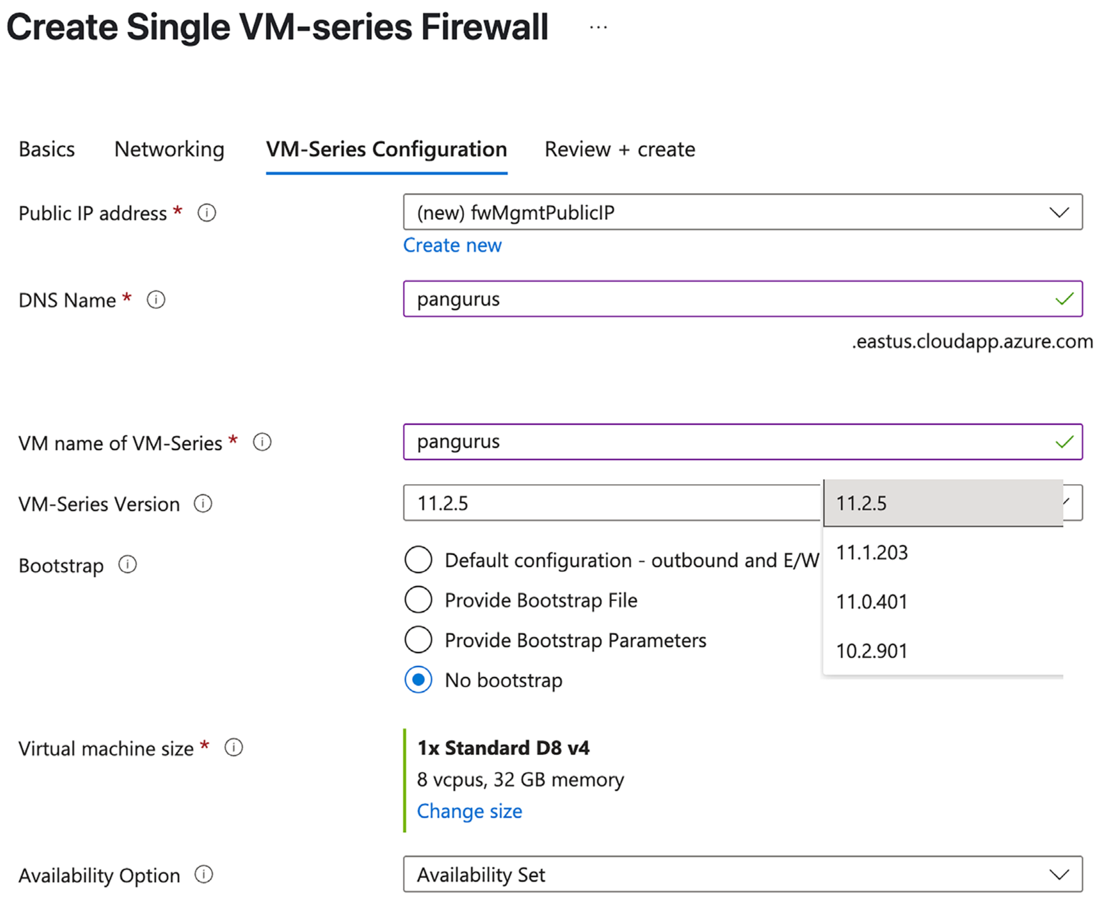


Figure 15.7: VM-Series Configuration

When you click **Review + create**, your configuration will be validated, and you will be presented with an overview of what you are about to deploy. The output will look like the following screenshot:

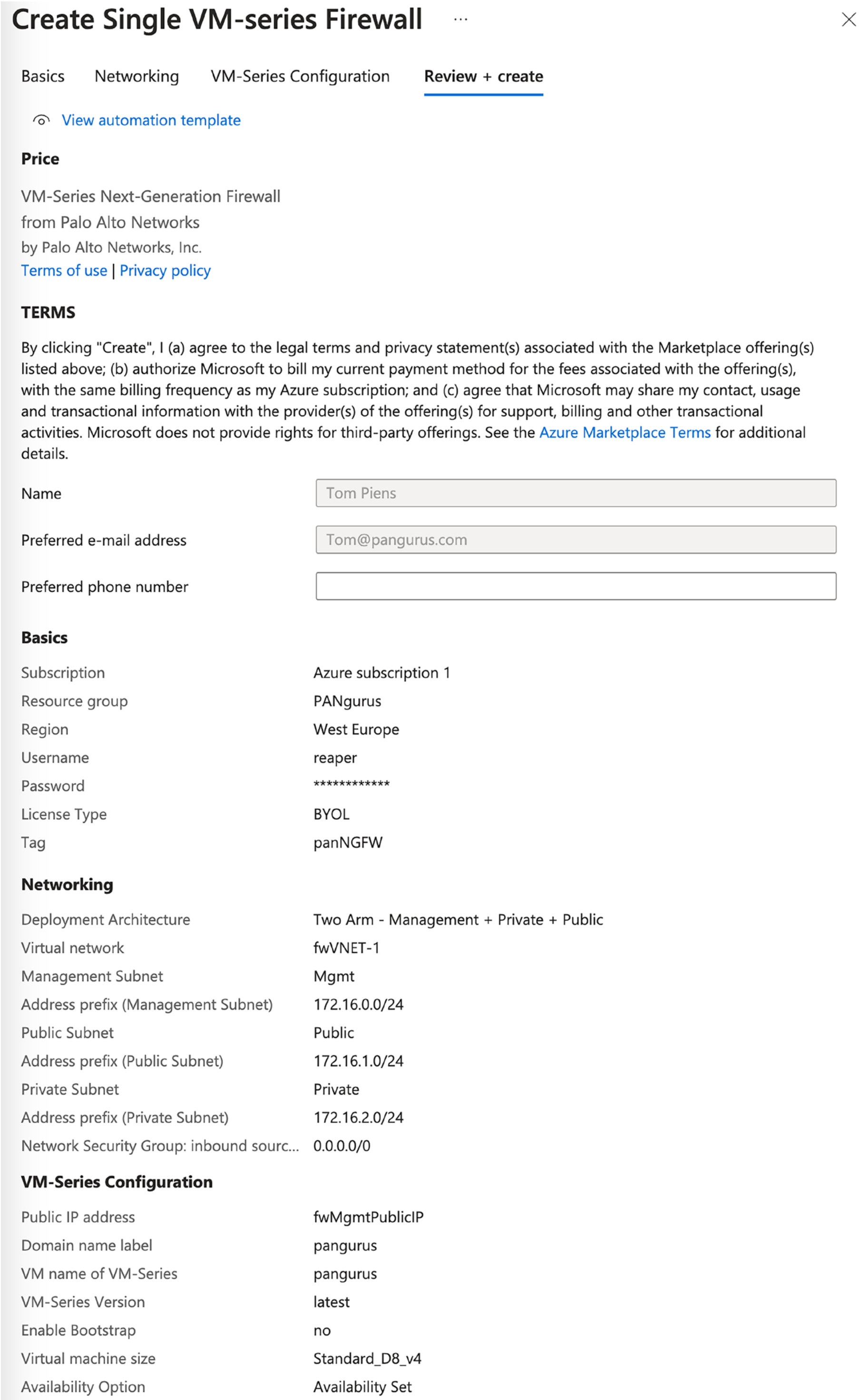


Figure 15.8: Review + create tab

Once you're ready to move ahead, click **Create** and the firewall will be deployed.

You'll be taken to a progress screen that shows you which components are being created. Wait for the **Deployment is in progress** page to change into **Your deployment is complete**; this process will take several minutes, so sit by patiently or go get a refreshment while waiting. The whole process will look similar to the following screenshots. Once completed, click the **Go to resource group** button:

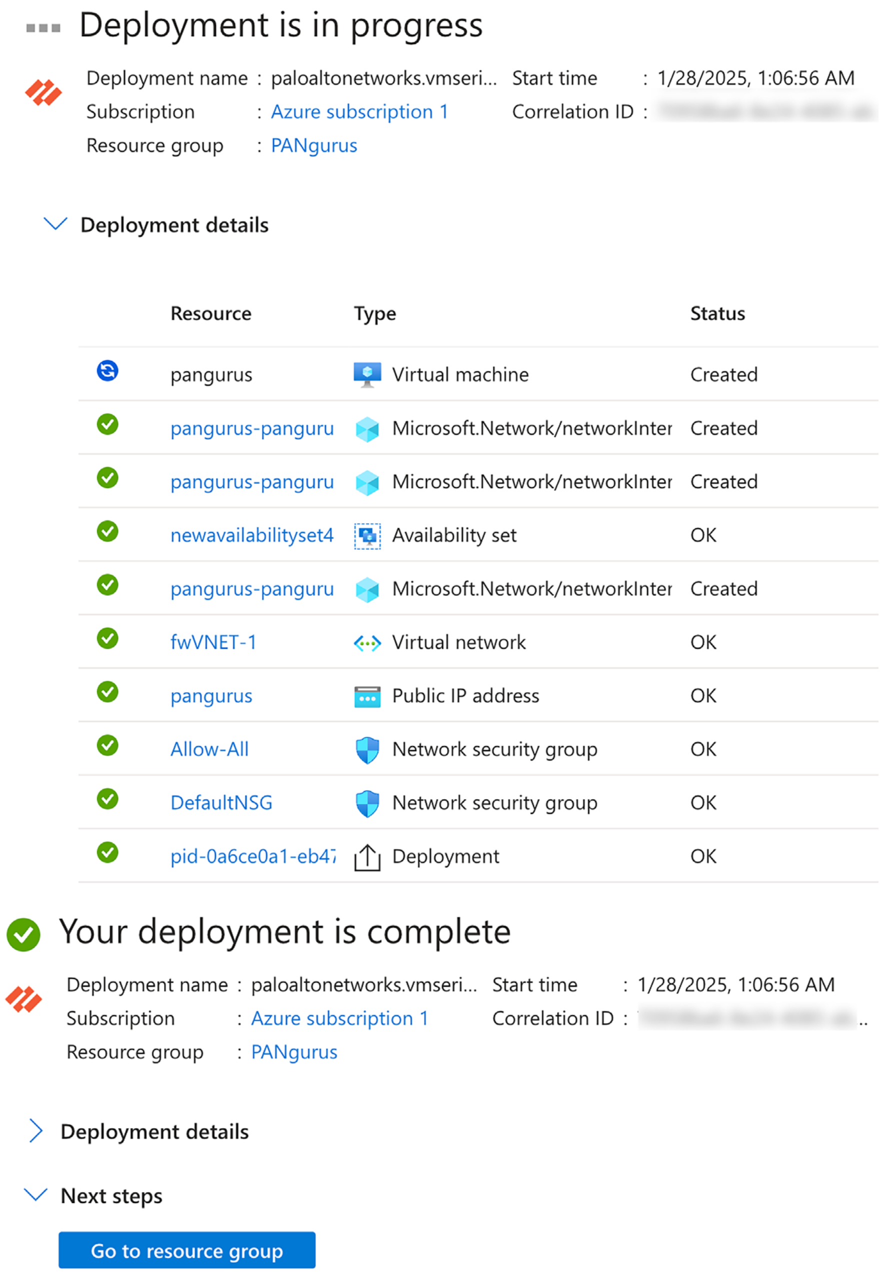


Figure 15.9: Deployment process

Once you're on the resource page, you will notice there are multiple resources listed, which can each individually be edited but may be dependent on other resources. The following screenshot illustrates a resource group with only a firewall in it. Once more resources are added, the list will grow significantly:

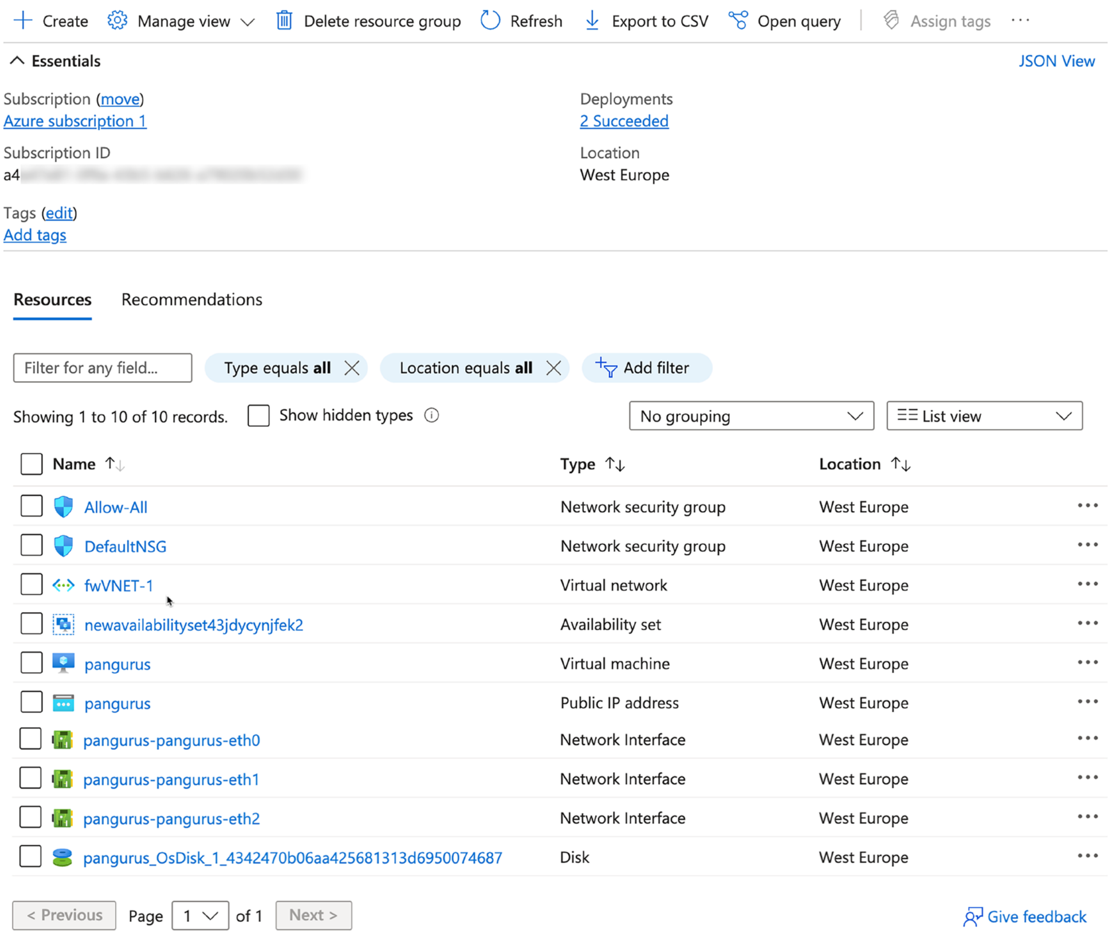


Figure 15.10: Resources associated with resource group and firewall

The following resources have now sprung to life:

- **DefaultNSG** is the "mini-firewall" component that controls which sources are allowed to connect to the management interface, and could also be used to control what is allowed to leave the VNet.
- **fwVNET-1** is the VNET that was created when the resource group was created. A VNET represents the network inside the resource group and can contain multiple subnets, but it is not able to communicate with other VNETs unless an explicit peering is established. A VNET can be compared to a physical network switch where multiple subnets live next to each other in different VLANs, and with the right configuration, they can be made to communicate with one another.
- The **Virtual Machine** is a resource object.

- The **Public IP address** assigned to the management interface is represented by a resource.
- Each **Network Interface** connected to the firewall has its own network interface resource.
- **Disk** storage is represented by a separate resource, as this also has consequences on pricing.

If you click on the public IP address resources, you can view their details, as you can see here:

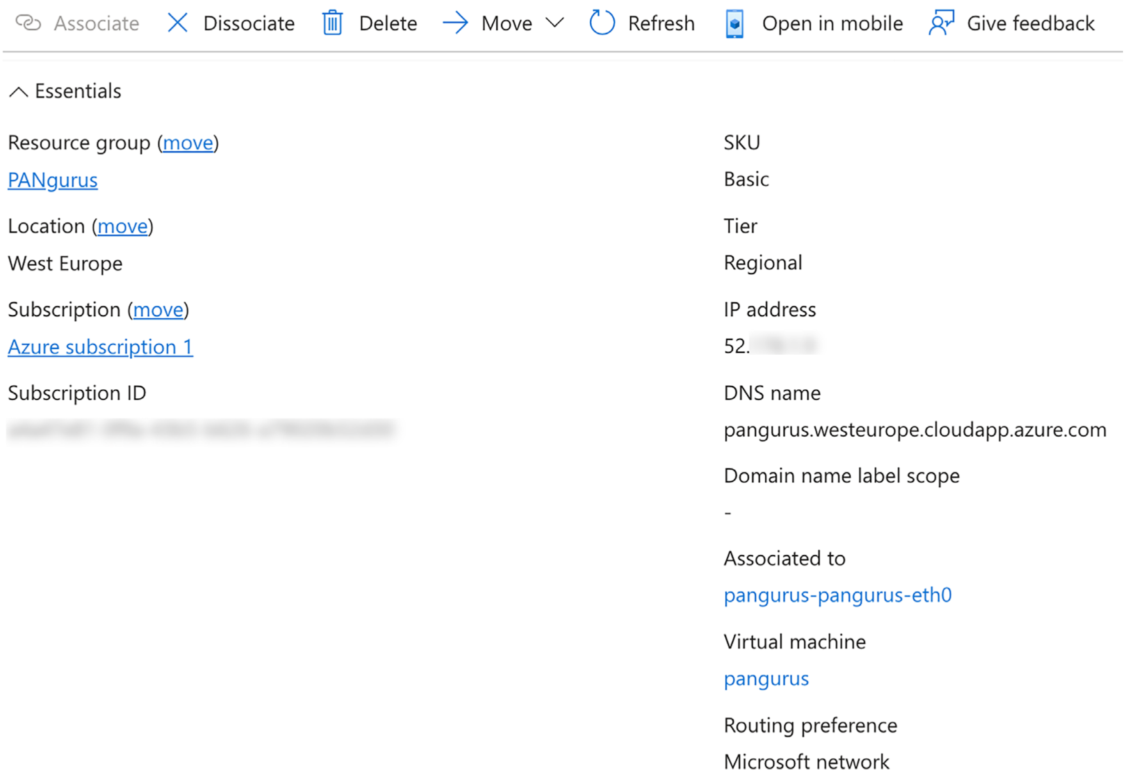


Figure 15.11: Public IP address details

You are now able to log on to the management interface using the FQDN or public IP and the username and password you created in the **Basics** tab. On the **Dashboard** page, you'll notice a few things:

- The **MGT IP Address** is set to the x.x.x.4 address of the subnet assigned to the management interface. This is because Azure reserves .1 through .3 by default, setting .1 as the default gateway for all hosts in the subnet.
- A serial number already exists if the **Bundle 1** or **Bundle 2 PayGo** option was selected, while the serial will be empty for BYOL deployments.
- The **VM License** will automatically be adjusted to the "size" of the VM you selected in the **VM-Series Configuration** tab. The default is a VM-300-sized firewall, but larger virtual machine sizes will automatically come with a larger VM license (DS4V2 equals the VM500, for example), and associated costs.

- The PAN-OS version of the firewall may be more recent than that selected during the deployment phase, which should not prevent you from double-checking if a more recent version is available.

The firewall dashboard will look similar to the following screenshot:

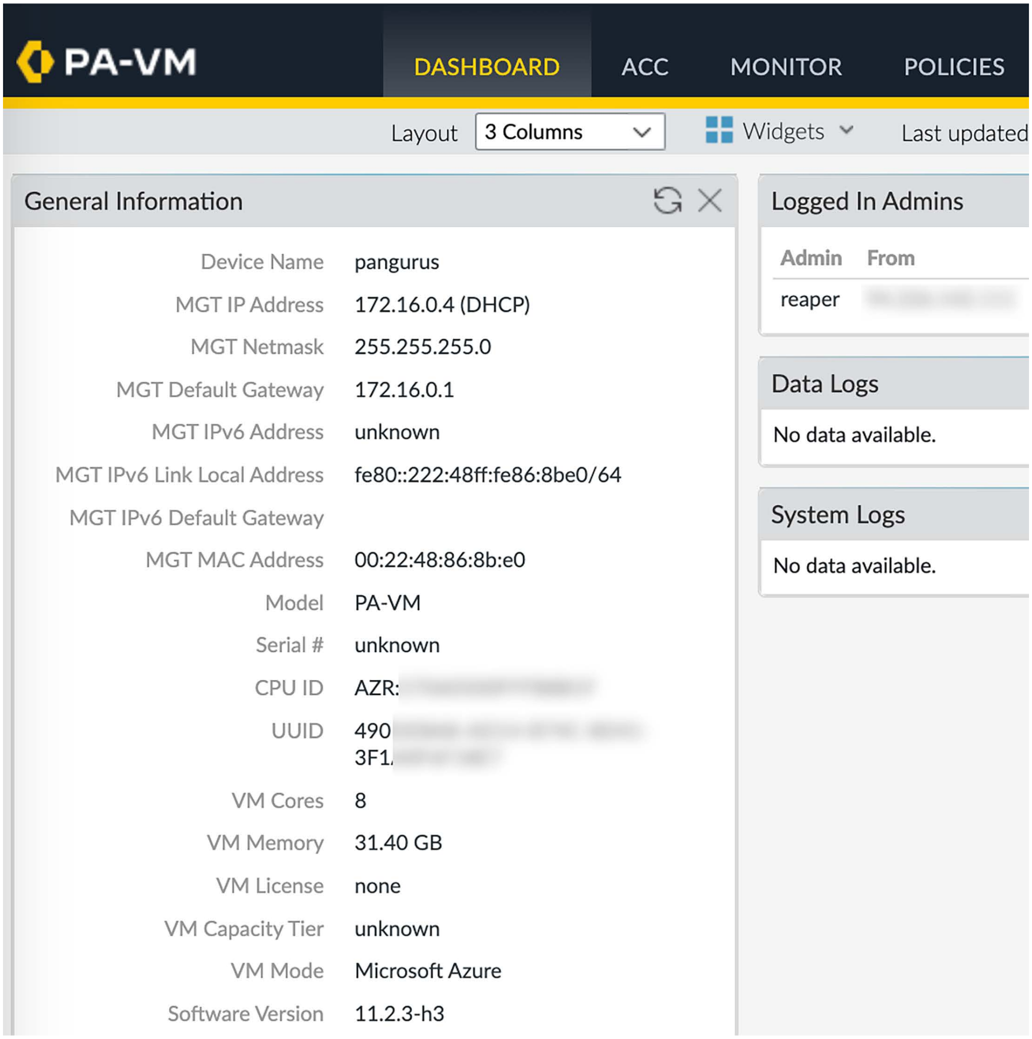


Figure 15.12: Firewall dashboard page

Your firewall is now up and running, and you can start registering it, if you chose the BYOL option, or immediately start updating it if you went with one of the PayGo options.

Because this is a fresh firewall installation, unless you have a Panorama instance set up with the appropriate configuration already, you will need to configure the firewall from scratch. To save time, you can also use bootstrapping to load the firewall with a base configuration. This can save time when setting up a fresh firewall, but can also be useful when you need to be able to scale your deployment (spin up additional firewalls during high load) or recover from a failure.

Bootstrapping a firewall

Bootstrapping the firewall pushes a pre-prepared configuration into a newly deployed firewall while it is being installed, so it is immediately operational.

This can shave off valuable time when recovering from a failure or ramping up a deployment to deal with an increased load on the infrastructure.

To enable bootstrapping, you first need to create storage that can be accessed by the firewall while it is being deployed. For an ESXi deployment, for example, you can create a root folder on your workstation that contains the required subfolders (see *Creating a bootstrap file share* for the correct folder structure), put the folder into an ISO file (using your preferred ISO burner tool), and upload the ISO to an accessible VMFS/NFS file share so it can be loaded as a disk image when the firewall is deployed.

Cloud deployments require a storage account so you can store the few files that will be used during the bootstrapping phase. From the Azure home page, search for `storage` and select the storage accounts icon. If a storage account already exists within your organization, select the appropriate one and move on to *Creating a bootstrap file share*. Otherwise, create a new account.

Creating a new storage account

When you create a new storage account, you need to select the appropriate subscription and resource group. It is important to add the storage account in the right resource group, as it needs to be the same group that the firewall(s) will be created in. The storage account will require a unique name and should be placed in a geolocation close to where the firewalls will be deployed.

Pro tip

For bootstraps, I usually pick the cheapest redundancy option, as we're only storing a few files that can be quickly recovered from local storage if needed. In a high availability environment, pick a more expensive option that offers a more robust redundancy option.

Your **Basics** tab should look similar to the following screenshot:

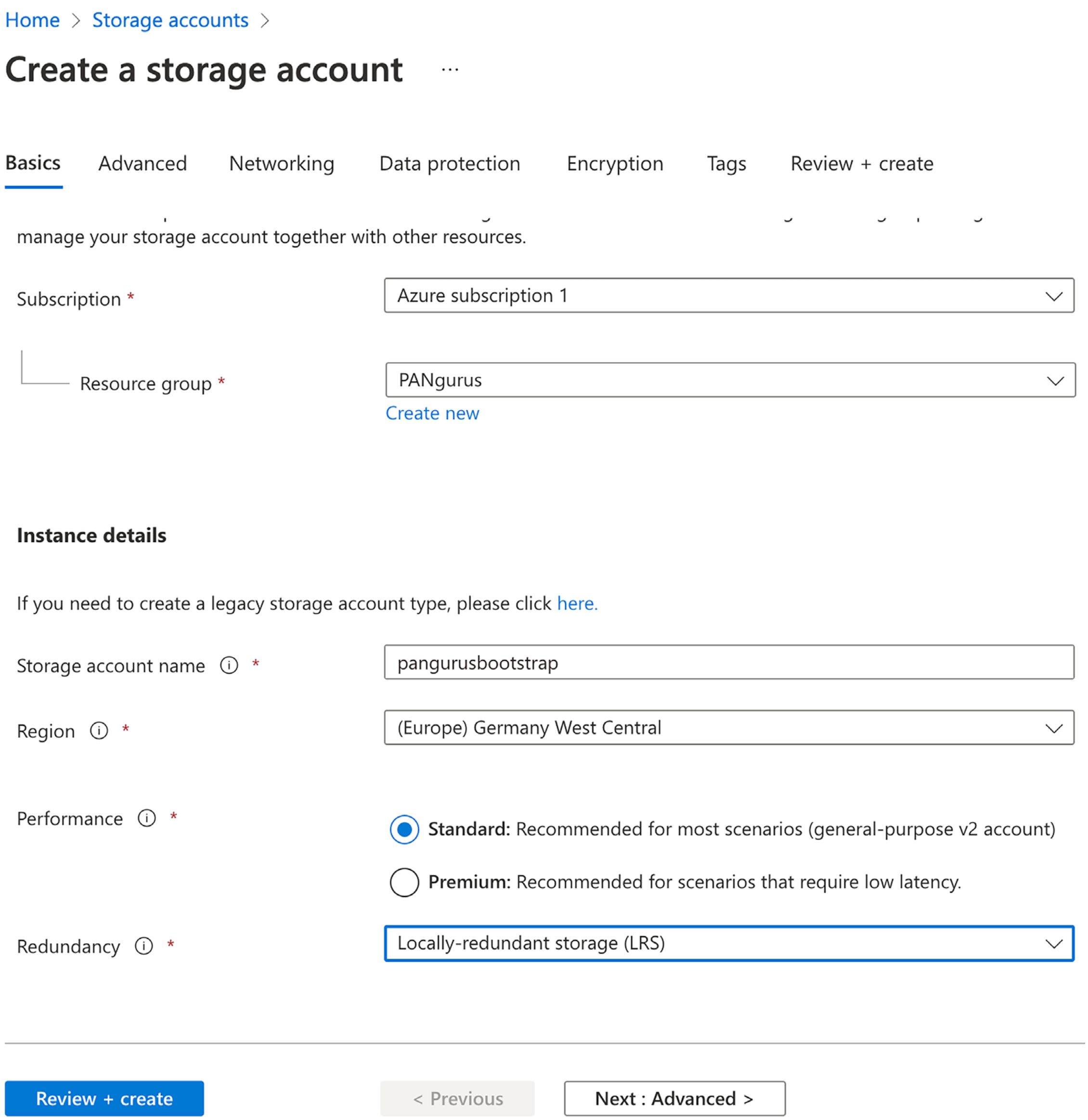


Figure 15.13: Bootstrap storage account basics

For now, select **Review + create** and then **Create** to make a new storage account. You will be taken to another **Deployment is in progress** screen, so wait a while for the process to complete. Once the process ends, click **Go to resource** to access the storage account.

The storage account will not be accessible without the access key, so open the **Access keys** menu, click **Show** next to the key, and copy the access key for use during the bootstrap phase of a new firewall deployment.

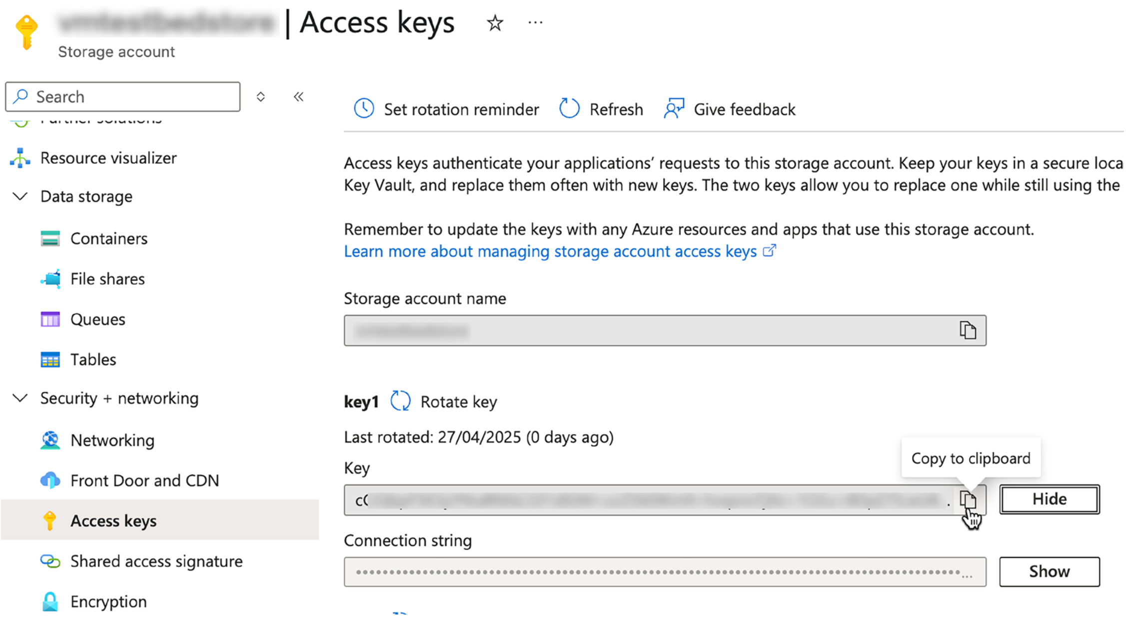


Figure 15.14: Creating a new file share

Next, you will need to create a file share that will contain a set of directories and files that will allow for a successful bootstrap.

Creating a bootstrap file share

In the storage account, navigate to **File shares** in the left-hand navigation and create a new file share, as illustrated in the following screenshot:

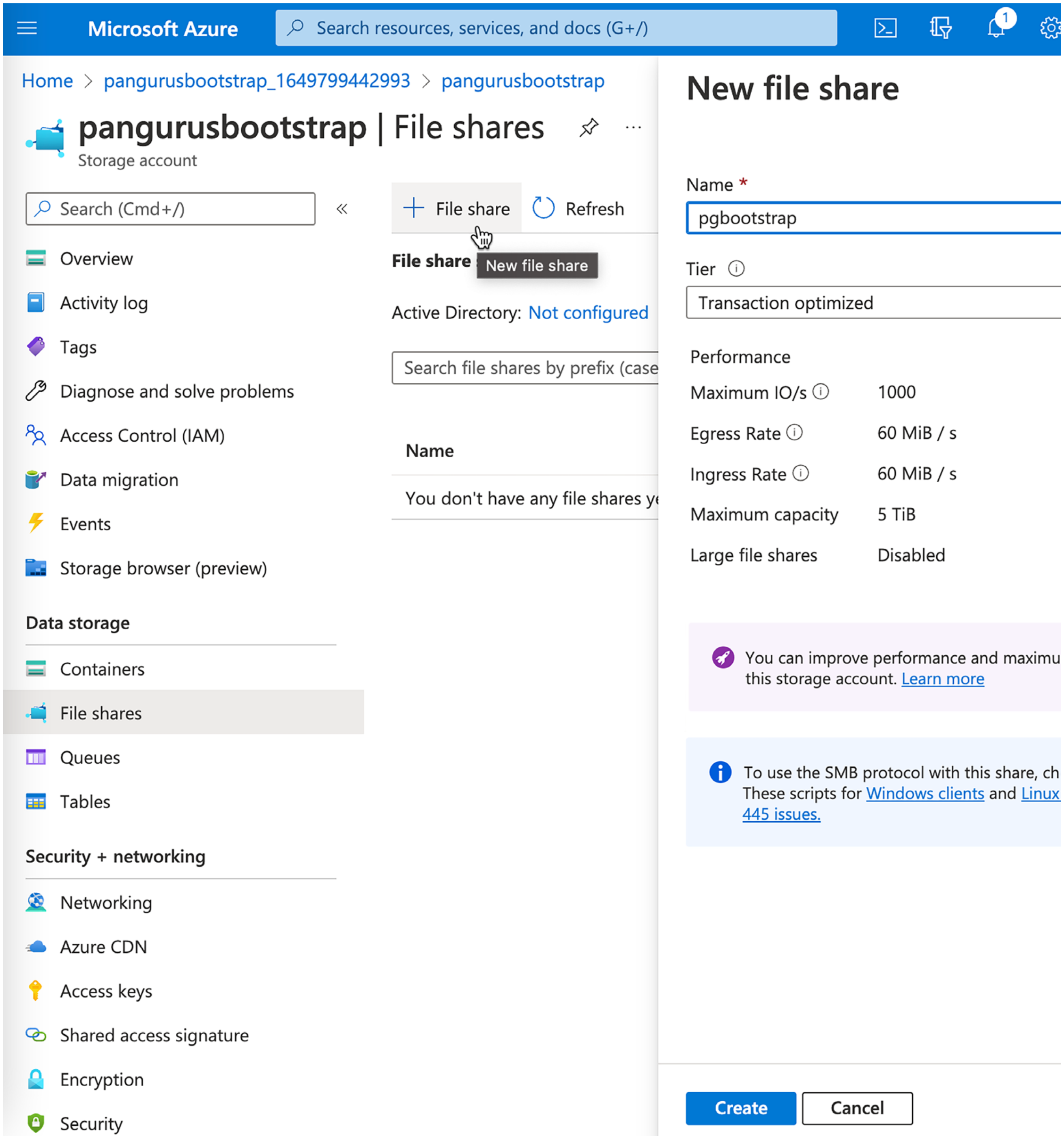


Figure 15.15: Creating a new file share

Once the file share is created, you will need to create a container folder in the root. If you need to have multiple different bootstrap versions available, you can create multiple folders in the root that represent each version. Inside each bootstrap version folder, four subfolders need to be created – **config**, **content**, **license**, and **software**, as illustrated in the following screenshot:

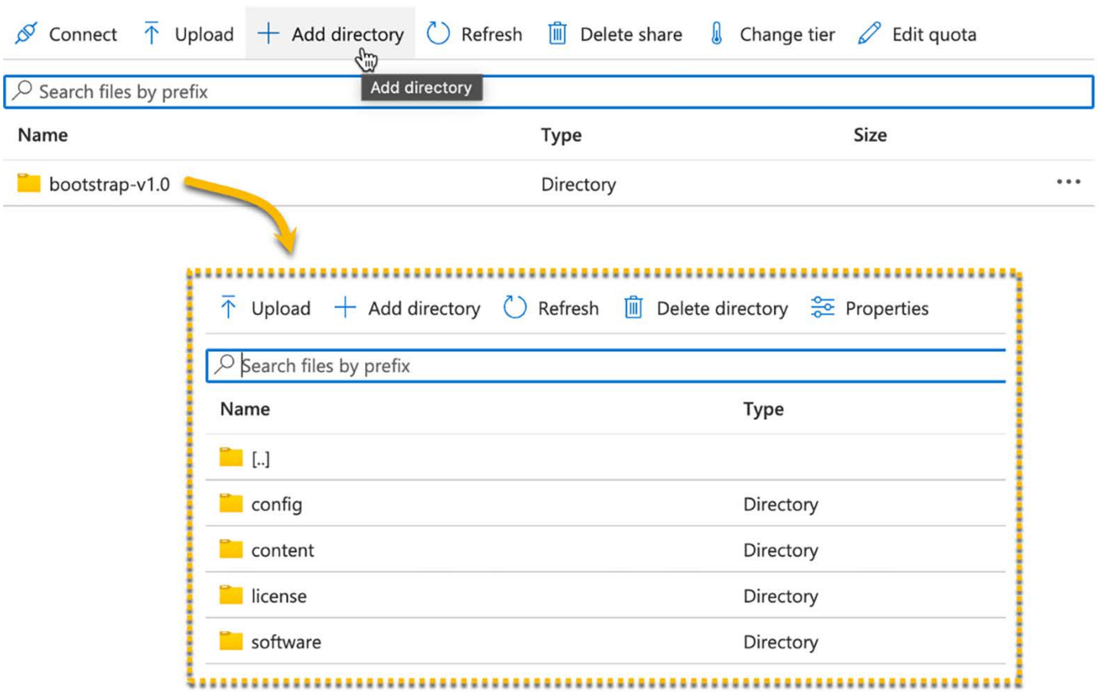


Figure 15.16: Bootstrap storage folder structure

All four subfolders need to be present for the bootstrap to function properly, even if some are left empty.

- In the **config** subfolder, an **init-cfg.txt** and **bootstrap.xml** file should be uploaded. More about these files in the next subsections.
- In the **content** folder, you can upload content packages so the firewall is immediately updated to this/these content versions.
- License files downloaded from the support portal as .key files can be uploaded to the **license** folder. The filename can be changed to suit your needs, but the .key extension needs to be maintained. A bundle license file will work best.
- If a specific PAN-OS version is desired for the bootstrapped firewall, it can be added to the **software** folder. All intermediate software packages need to be uploaded to get the firewall to the desired version from the base version provided by the hypervisor, i.e., if the base image is 10.0 and you want to upgrade to 10.2, you will need to upload the 10.1.0 base image, and 10.2 base and maintenance images.

- An optional folder called **plugins** can be created if a specific **vm_series plugin** (installed in **Device | plugins**) version is required for the bootstrapped firewall. Do not upload more than one plugin file, only the **vm_series plugin** you might need.

Don't add any additional folders other than the five folders listed above.

As mentioned above, the **config** subfolder should be prepared by adding two configuration files. Let's look at these, starting with the **init-cfg.txt** file.

The init-cfg.txt file

The **init-cfg.txt** file contains the basic configuration for the management interface. It will set a static IP address, subnet, and default gateway, or turn the interface to DHCP client mode. It will provide DNS configuration, Panorama IP addresses, and a basic template or device group configuration as needed.

You can use a single **init-cfg.txt** file, or create multiple versions and set a prefix in the filename to indicate which device(s) it is intended for by matching the prefix to the target's serial number or UUID (e.g., **0008C000001-init-cfg.txt**). When the newly created firewall boots, it will first look for a file containing its serial or UUID, then default to the regular **init-cfg.txt** file for its management configuration.

The **init-cfg.txt** file should contain a selection of the following parameters. Any parameters that are not used (i.e., panorama-server for unmanaged systems) can be left blank.

	Parameter	Value
*	`type=`	`static` or `dhcp-client`
**	`ip-address`	Sets the static IP of the mgmt. interface; can be left blank if the interface type is set to `dhcp-client`
**	`default-gateway=`	Sets the static gateway of the mgmt. interface; can be left blank if the interface type is set to `dhcp-client`
**	`netmask=`	Sets the static netmask of the mgmt. interface; can be left blank if the interface type is set to `dhcp-client`
	`ipv6-address=`	Optionally sets an IPv6 address on the `mgmt. interface`
***	`ipv6-default-gateway=`	Optionally adds the IPv6 gateway
	`hostname=`	Sets the firewall hostname

	Parameter	Value
	`panorama-server=`	Sets the primary Panorama IP or FQDN
	`panorama-server-2=`	Sets the secondary Panorama IP or FQDN
	`tplname=`	The template stack the firewall will be assigned to in Panorama
	`dgname=`	The device group the firewall will be assigned to in Panorama
	`cgname=`	The collector group the firewall will be forwarding logs to
	`dns-primary=`	Primary DNS IP
	`dns-secondary=`	Secondary DNS IP
	`vm-auth-key=`	An auth key can be pre-generated on Panorama to authenticate VM devices; this attribute does not apply to hardware firewalls
	`op-command-modes=`	Operational commands allow some special operational modes to be preset: 1. `multi-vsys` enables multiple virtual systems on hardware firewalls 2. `jumbo-frame` enables support for jumbo frames 3. `mgmt.-interface-swap` allows you to switch a predefined data plane interface with the management interface on a VM system. This is only supported on AWS, GCP, ESXi, and KVM (so not supported on Azure)
	`op-cmd-dpdk-pkt-io=`	On any platforms that support DPDK (Data Plane Development Kit), the VM equivalent of hardware acceleration, you can enable (`on`) or disable (`off`) DPDK

	Parameter	Value
	plugin-op-commands=	Operational commands for the VM plugin can be added as a single line separated by commas: 1. sriov-access-mode-on ESXi and KVM only 2. aws-gwlb-inspect:enable enables AWS integration with load balancer 3. aws-gwlb-associate-vpce:<vpce-id>@ ethernet<subinterface> associates a VPC endpoint with a VM series interface 4. aws-gwlb-overlay-routing:enable enables overlay routing for the VM 5. set-dp-cores:<#-cores> customizes the number of DP CPUs (PAN-OS 10.1 or later) 6. plugin-op-commands=set-cores:<#-cores> specifies the number of vCPUs when using NGFW credits 7. numa-perf-optimize:enable enables NUMA performance optimization for VMs that have VM-plugin 2.1.2 or later installed
	dhcp-send-hostname=	Sends the hostname to the DHCP server
	dhcp-send-client-id=	Sends the client ID to the DHCP server
	dhcp-accept-server-hostname=	Accepts the hostname assigned by the DHCP server
	dhcp-accept-server-domain=	Accepts the domain assigned by the DHCP server
	vm-series-auto-registration-pin-id= vm-series-auto-registration-pin-value=	pin-id and value are used to automatically register a VM to AutoFocus and Strata Logging Service. Both values can be configured from the Customer Support Portal: https://support.paloaltonetworks.com

Table 15.1: Parameters for the init-cfg.txt file

*If a required parameter is missing from the **init-cfg.txt** file, the bootstrap process is aborted, and the firewall is booted in the default configuration. The parameters marked by an asterisk in the table above are required.

** IP information is only required if the `type=parameter` is set to static.

*** `ipv6-default-gateway=` is only required if an `ipv6-address=attribute` is present.

Examples for a static and a DHCP-enabled management interface **init-cfg.txt** file can be found at `https://github.com/PacktPublishing/Mastering-Palo-Alto-Networks-Third-Edition` by Packt.

> **Note**
>
> There are no spaces between the attribute, = symbol, and the value.
>
> E.g., `type=static`, not `type = static`

The bootstrap.xml file

If the firewall will not be managed by Panorama, a full config file can also be provided to configure the firewall upon deployment.

This file can be created from scratch, but it is easier and more convenient to export a configuration file from an existing firewall and customize it to suit the bootstrap requirements. On an existing firewall, go to **Device** | **Setup** | **Operations** and select **Save named configuration snapshot** to create a fresh config file from your current configuration, and then click **Export named configuration snapshot** and select the file you just created.

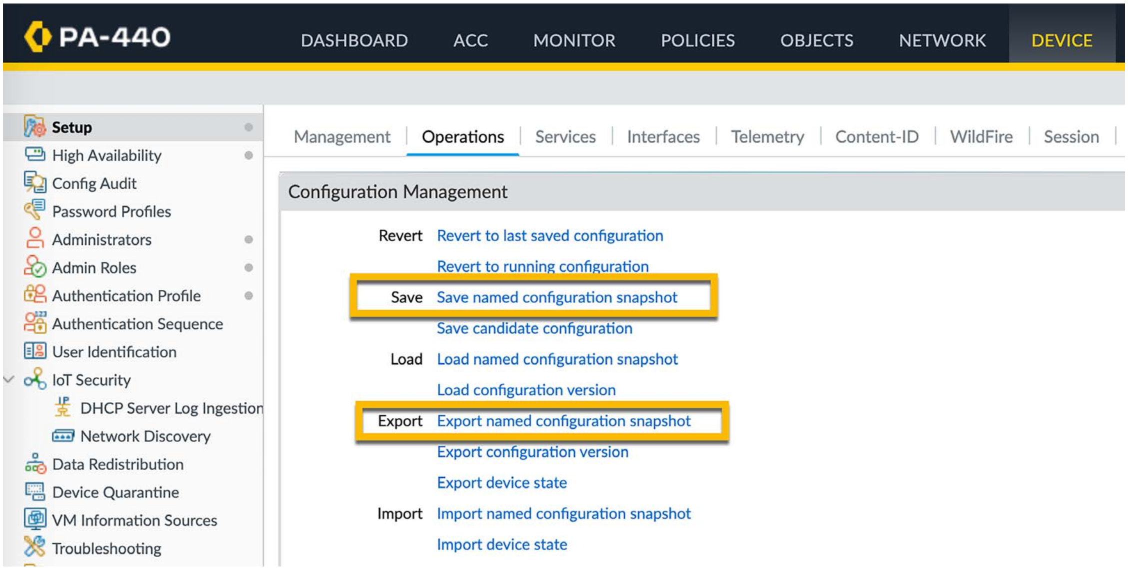


Figure 15.17: Save and Export config file

Edit the file as needed, rename it to **bootstrap.xml**, and upload it to the **config** folder in the storage account.

Bootstrapping a firewall on Azure

To bootstrap a firewall, we start off from the same spot as we do for a simple Marketplace deployment: from the Azure dashboard, access the **Marketplace**, search for **Palo Alto Networks firewall**, and start a PayGo or BYOL deployment.

In the **Basics** and **Networking** tabs, provide the same information, subscription and resource group, region, VNET, and inbound security group restrictions for your management IP:

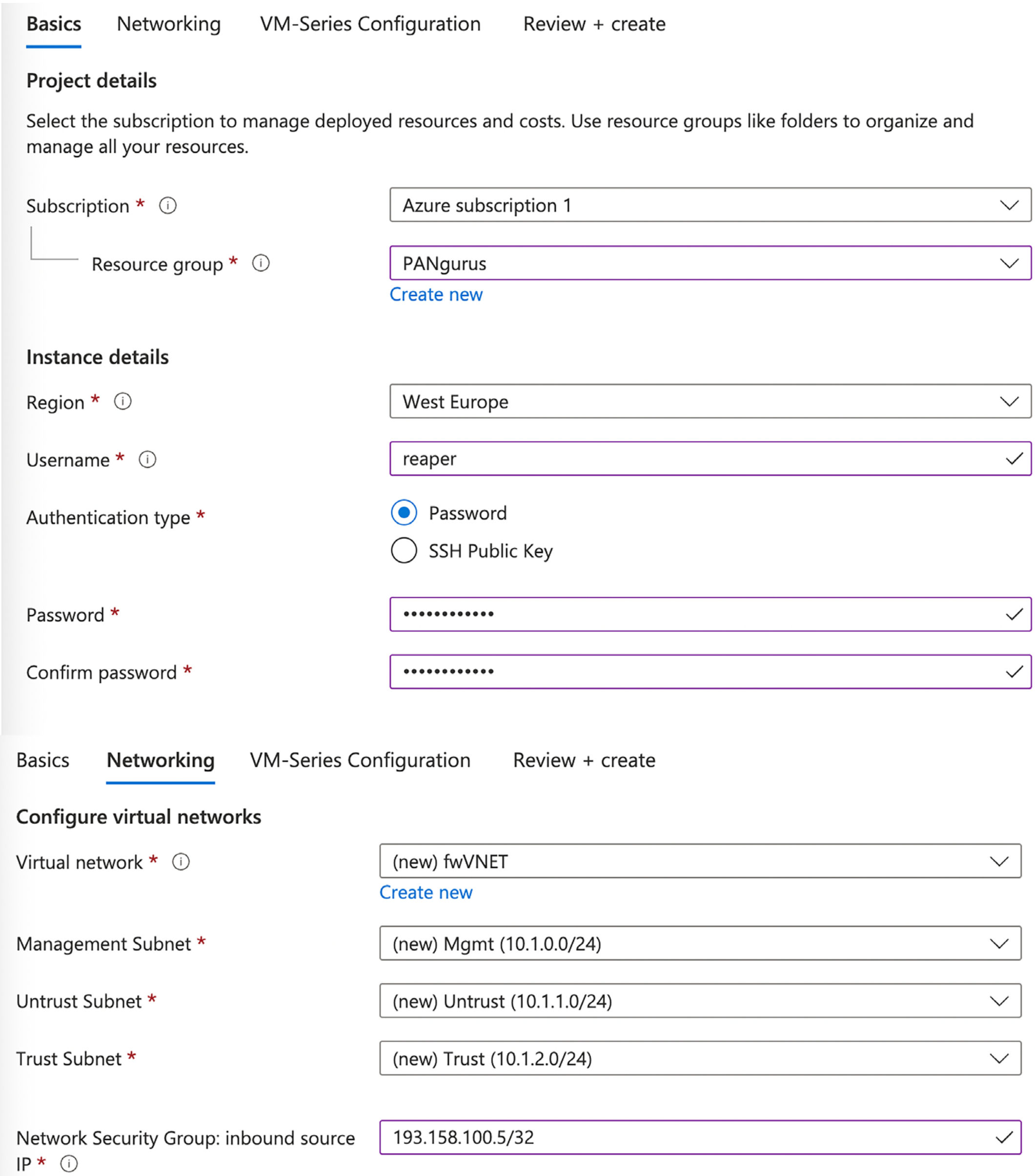


Figure 15.18: Basics and networking configuration for the bootstrapped firewall

The **VM-Series Configuration** tab is where we can enable bootstrapping. You will need to provide the name of the file share we created, the access key we copied earlier, and the storage account name. Optionally, you can add a subdirectory name if more than one bootstrap folder exists in the file share.

You should also provide a DNS name to easily access the management interface of the firewall. This DNS name is mapped to the public IP address that is automatically created with this deployment. If you have a pool of static IP addresses available in your subscription, you can select an available IP from the dropdown.

If you do not want the management IP to be reachable from the internet, we can delete the public IP address after the deployment. We want to keep it for now so we can verify that all the settings we provided in the bootstrap have been properly applied.

Provide all the information and click **Review + create** and then **Create** to start the deployment:

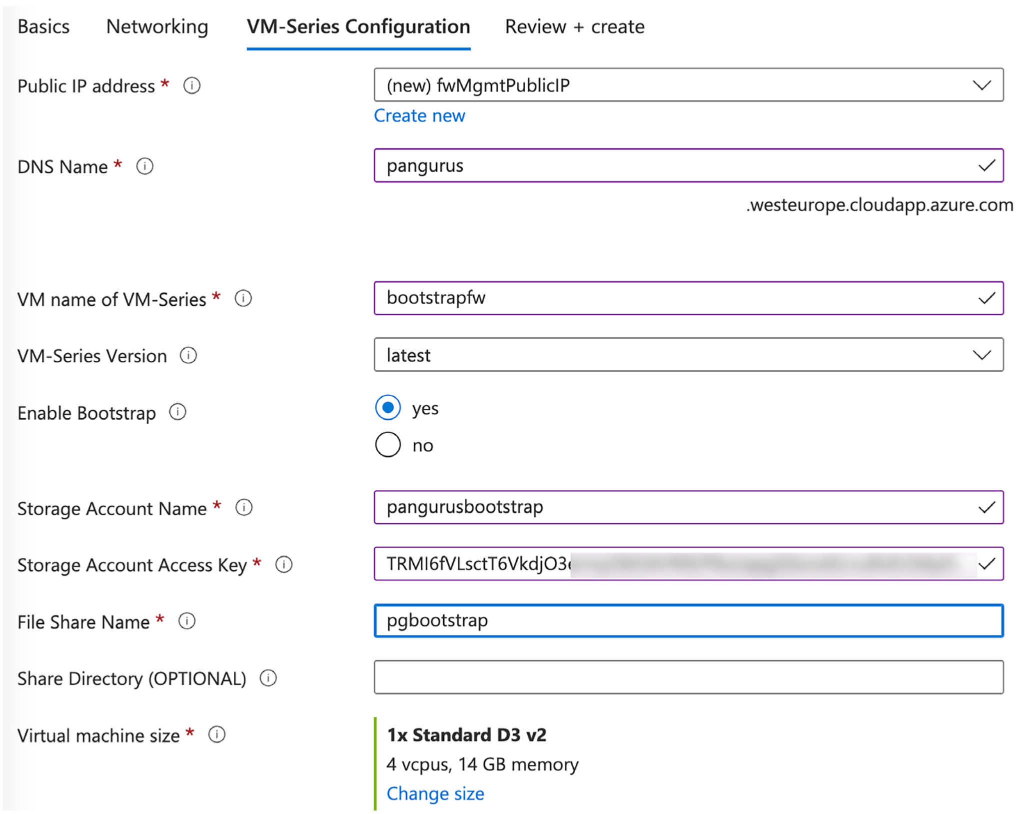


Figure 15.19: VM-Series Configuration for the bootstrapped firewall

Wait a few moments for the deployment to complete. You should see all the individual tasks being completed, as in the following screenshot:

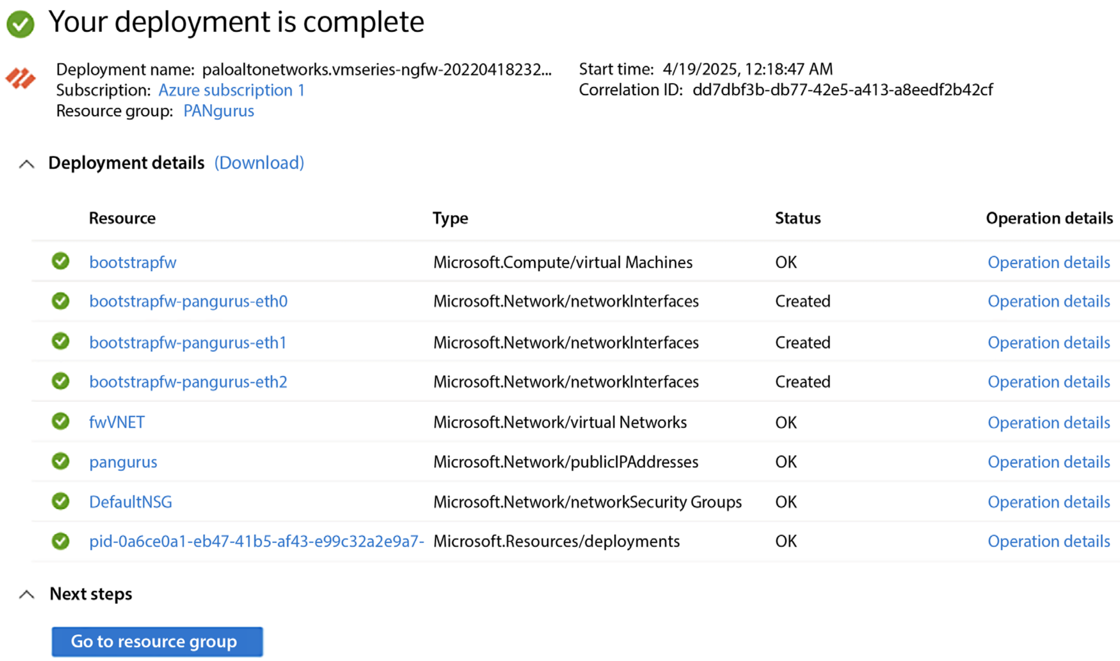


Figure 15.20: Completion of the bootstrap deployment

You can now access the management interface through the DNS name you provided earlier, and the username/password provided in the configuration, or an admin account that may have been included in **bootstrap.xml**:

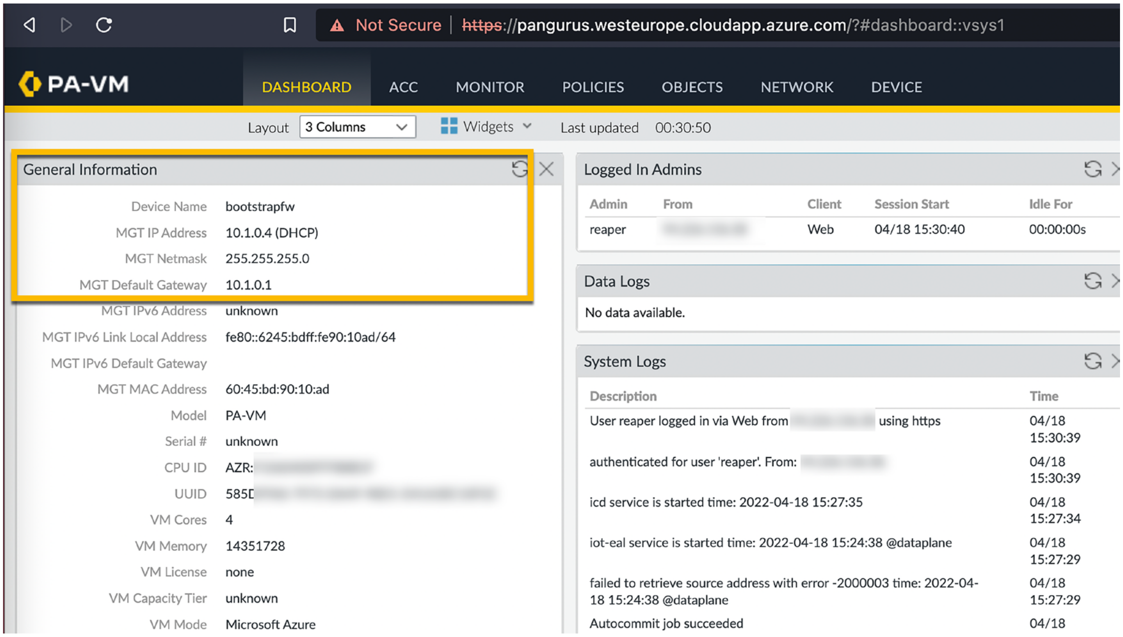


Figure 15.21: Verify the bootstrap config was applied

You now have a running firewall on Azure, but that's not the end of it: you still need to configure the rest of the resource group and the VNET.

Putting the firewall in line

Simply configuring the firewall is not enough: a cloud environment behaves quite differently from a traditional network. If you haven't deployed many firewalls in a cloud setting before, the most important considerations will be covered in this section. We will focus on Azure in this section to align with the previous sections. Other cloud vendors have similar processes.

When the firewall is created, one of the additional objects that gets created in Azure is the **DefaultNSG**. An NSG, or Network Security Group, is the firewall component in Azure networking that creates an inbound bridge from the internet. This means any subnet in the VNET that is not added to the NSG will not be able to receive connections from the internet.

In addition to being a member of the NSG, a public IP address object is required to receive incoming connections, which are mapped to either an interface or a load balancer.

The default deployment only has the management subnet in the NSG and a single public IP assigned to the eth0 (**Mgmt**) of the VM, as you can see in the following screenshot:

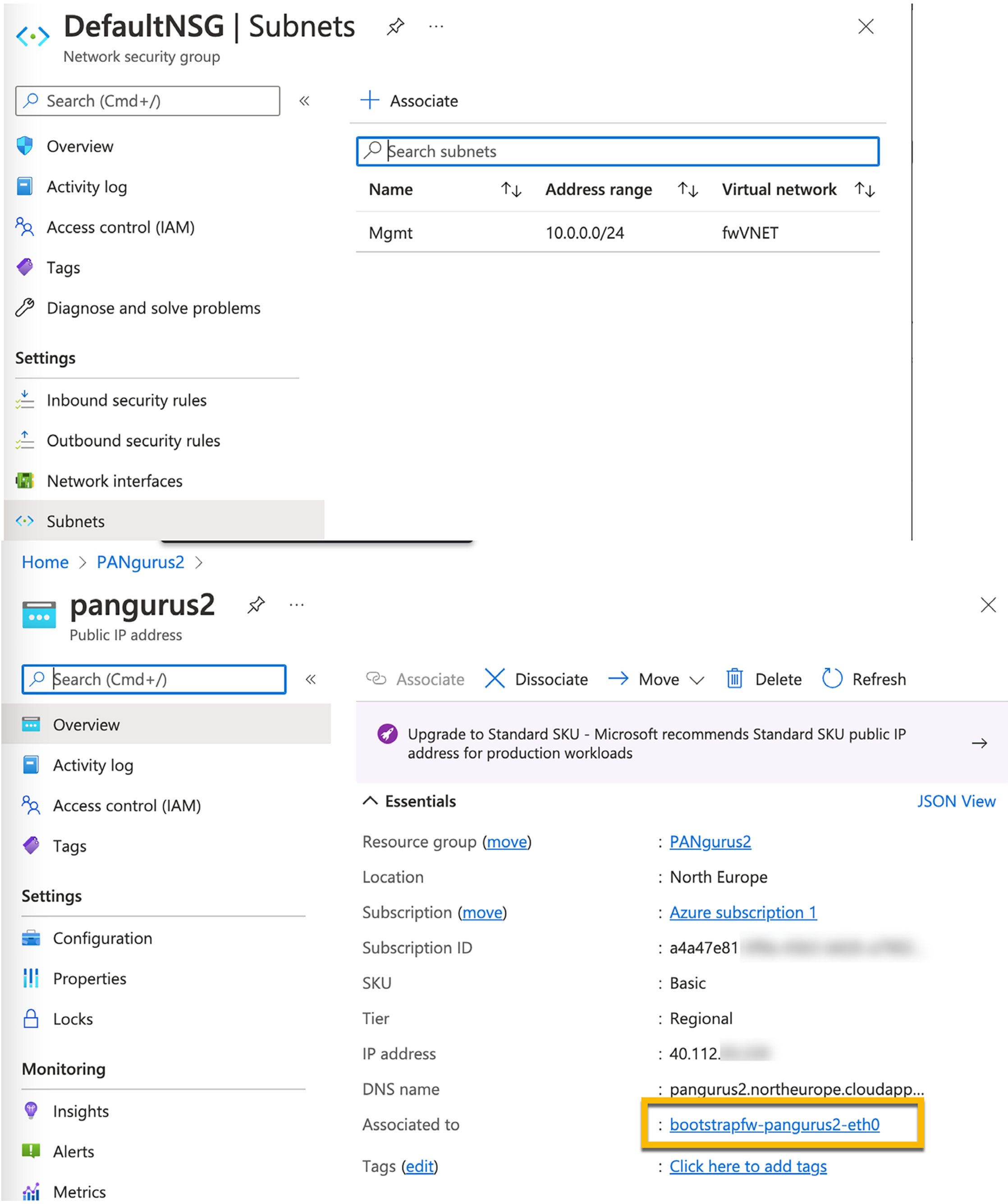


Figure 15.22: DefaultNSG and public IP

In this configuration, the management interface is able to accept incoming connections from the internet, but the **Untrust** interface on the firewall is only able to send packets out to the internet. If the firewall also needs to be able to receive inbound connections, an additional public IP address needs to be assigned to the firewall **eth1** (**Untrust**) interface, and the **Untrust** subnet added to an NSG.

Adding a new public IP address

In the search bar, search for `Public IP`, and open the **Public IP addresses** service. Click **Create** to add a new public IP. The **Standard** SKU is a static IP, while the **Basic** SKU can be dynamic or static. Both can be set with a regional dynamic DNS record, `<yourname>.<region>.cloudapp.azure.com`, which can come in handy as a CNAME for a proper subdomain or as an FQDN for any IPsec tunnels. Add it to the proper resource group and click **Create**.

Once it is deployed, it can be assigned to the **eth1** (**Untrust**) interface of the firewall, as illustrated in the following screenshot:

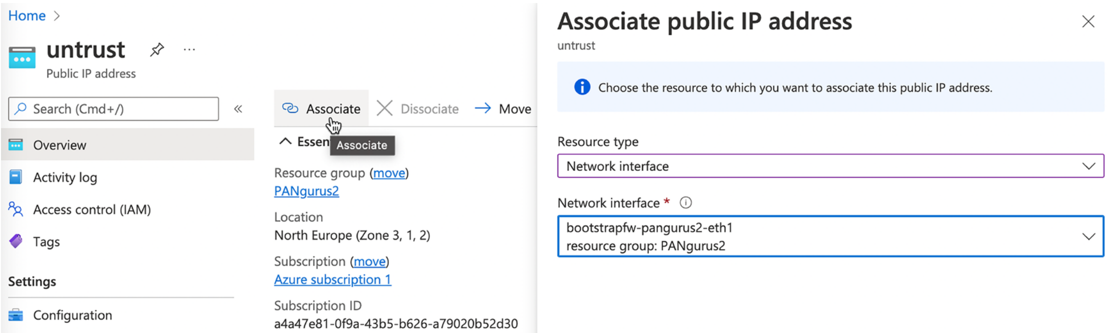


Figure 15.23: Assigning a public IP to eth1

The **Resource type** option will allow you to bind directly to an interface or to a load balancer. If you intend to deploy two or more firewalls, you could choose to use a load balancer to distribute incoming connections over multiple firewalls. (Make sure that if you deploy multiple firewalls, all have the exact same subscription and support licenses.)

Adding the Untrust subnet to an NSG

Because the **DefaultNSG** object is used to limit access to and from the management interface, it's a good practice to create a fresh NSG.

To create a new NSG, type `NSG` in the search bar at the top, click on **Network Security Groups**, and then click **Create**. Assign the new NSG to the appropriate resource group and click **Create**.

Once the deployment is complete, click on **Go To Resource** and access the subnets. Click **Associate**, select the appropriate VNET, and select the **Untrust** subnet, as illustrated in the following screenshot:

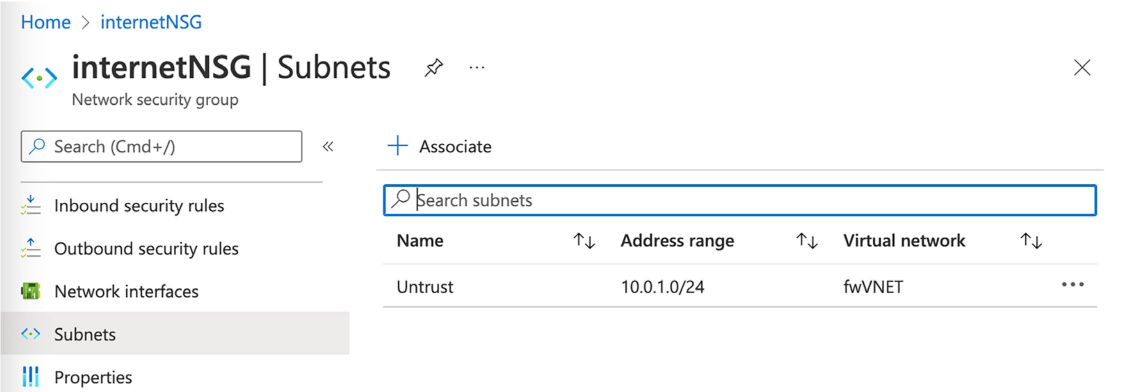


Figure 15.24: Adding Untrust subnet to an NSG

Next, access **Inbound security rules** and add a new rule that allows everything, as by default the NSG limits inbound access. The outbound security rules should already have a rule that allows all outbound traffic access to the internet.

When you add new rules to the NSG, they are assigned a priority that is matched lowest to highest. If you need to add multiple rules, make sure the priority is set so that they are stacked in the appropriate order. The new inbound security rule should look similar to the following screenshot:

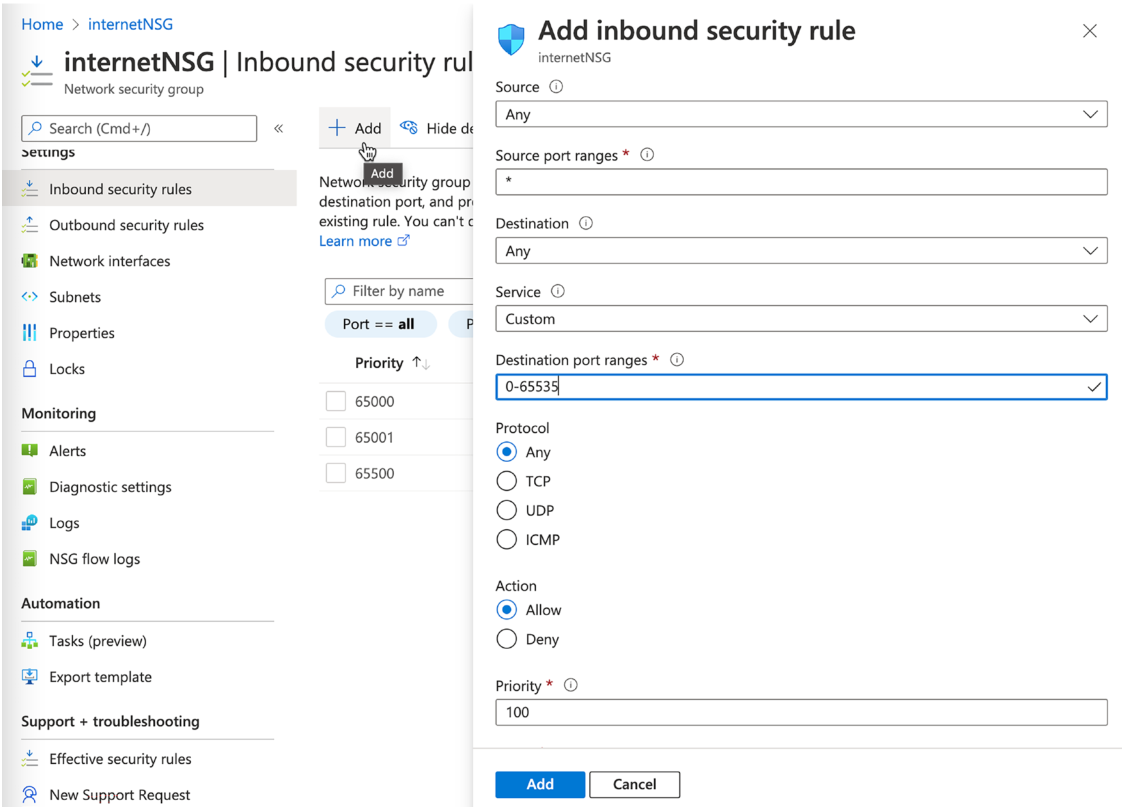


Figure 15.25: Inbound security rule for the Untrust subnet

You will have noticed that all subnets in the VNET are private (RFC1918), yet the firewall is able to communicate with the internet, and incoming connections on the public IP address are arriving on the firewall with its private IP.

Creating a server subnet

If additional subnets are needed, for example, to host your servers, you can easily add them by navigating to your resource group and clicking on the **VNET** object. Next, select the subnets and click **+ Subnet** to create a new subnet.

You can give it a name, and Azure will automatically populate the next available **/24** subnet in your VNET range. Change this if needed. You can also choose to associate the new subnet with an NSG or route table immediately:

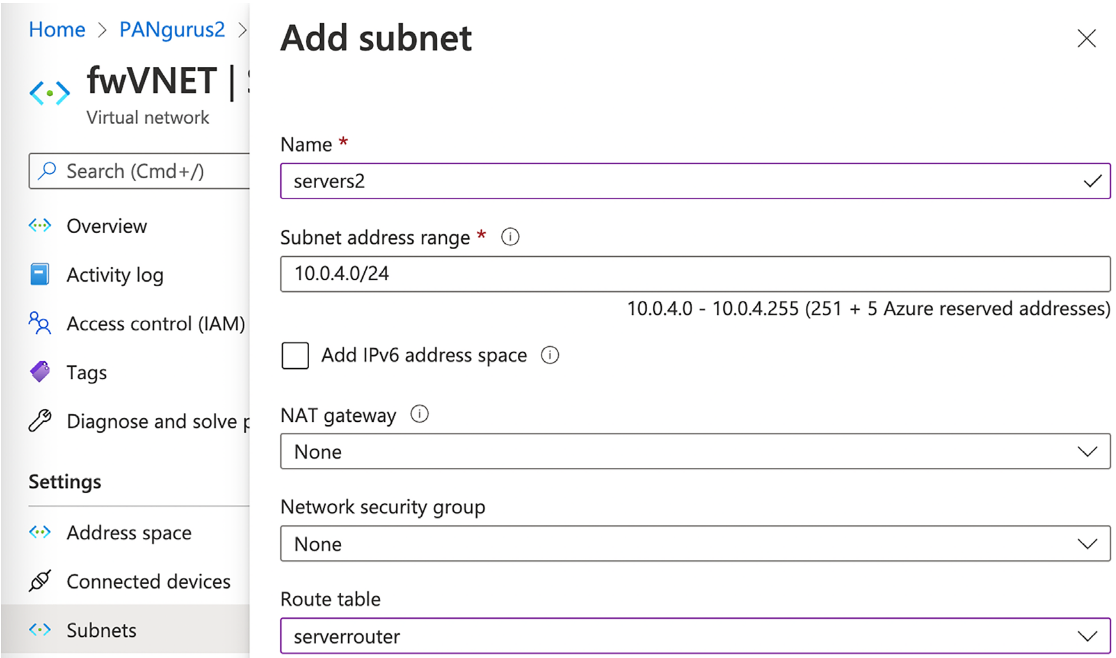


Figure 15.26: Adding new subnets

It is advisable to host your servers in a different subnet than the ones used by the firewall, as this allows more control over how sessions are routed.

Setting up routing

In each subnet, the x.x.x.1 through x.x.x.3 addresses are reserved for Azure, with x.x.x.1 being the default route for hosts in the subnet. This default route allows hosts in the subnet access to the internet, which, since we are setting up a firewall, is not something we want for all of our hosts.

We first need to ensure the firewall behaves as expected, so we need to set the interfaces up properly and set routing in the virtual router.

By default, `ethernet1/1` is the Untrust interface, `ethernet1/2` is trusted, and both will be associated with the corresponding subnet in the Azure VNET.

In other words, the VNET will consist of the following subnets that are linked to three Azure network interfaces that are associated directly with a Palo Alto VM interface:

```
fwVNET Mgmt 10.0.0.0/24 -> vmname-resourcegroup-eth0 -> palo alto VM management
interface
fwVNET Trust 10.0.1.0/24 -> vmname-resourcegroup-eth1 -> palo also VM
ethernet1/1
fwVNET  Untrust 10.0.2.0/24 -> vmname-resourcegroup-eth2 -> palo alto VM
ethernet1/2
```

All of the Palo Alto interfaces should be configured as DHCP clients and will receive the x.x.x.4 IP inside their respective subnet address. For the data plane interfaces, disable **Automatically create default route pointing to default gateway provided by server**, as illustrated:

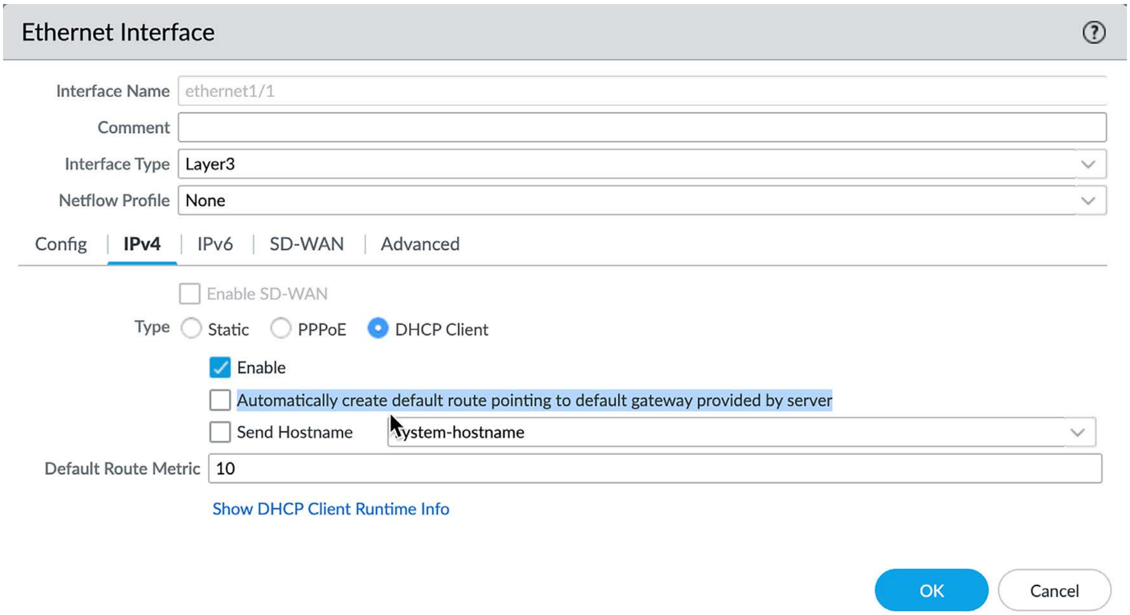


Figure 15.27: Data plane DHCP configuration

Instead of a dynamic default route, add routes in the virtual router manually, pointing to the respective x.x.x.1 IP of each subnet. In the following example, my 0.0.0.0/0 **default route** points to 10.0.1.1, which is the internet-facing Untrust subnet, while my server subnet 10.0.3.0/24 is directed at 10.0.2.1, which is the internal Trust interface. If you deployed multiple virtual routers, don't forget to add routes to **Next VR** on each VR:

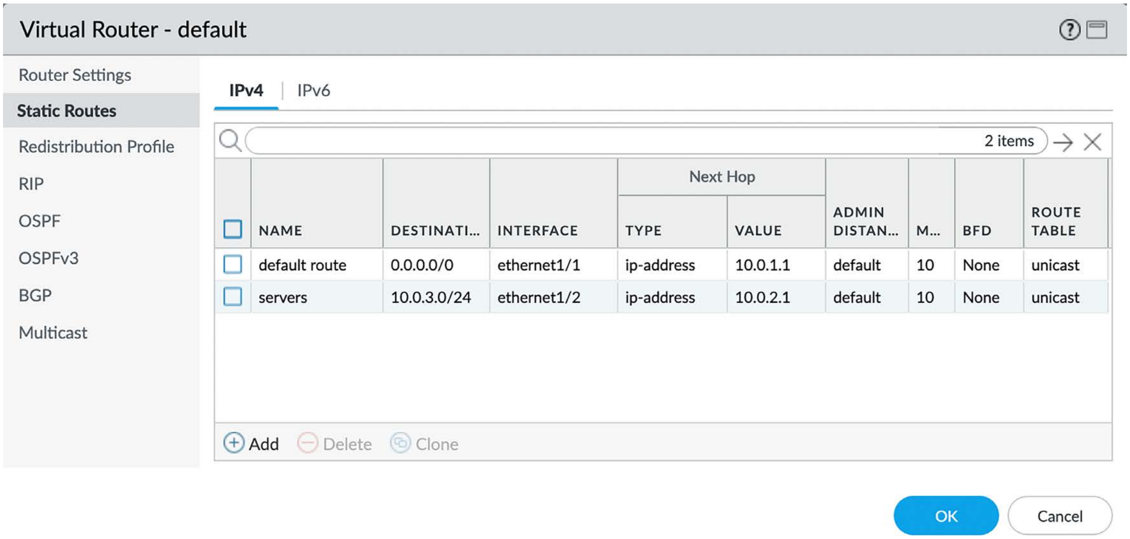


Figure 15.28: Virtual Router routes

The firewall is now set up to pass traffic, but the servers still need to be made to talk to the firewall.

> It is recommended to place all servers in one or more additional subnets to facilitate routing and to allow for segmentation.

Forcing internal hosts to route over the firewall

By default, any "internal" servers will send outbound packets to the x.x.x.1 IP in their subnet, which will allow them access to the internet.

To force these outbound connections to be pointed at the firewall rather than the internet, an Azure route table needs to be created to change the default behavior of Azure routing. The server should be in a different subnet than the firewall interface to prevent routing loops.

In the search bar, type `Route` and select **Route Tables** from the search results.

Click **Create** to make a new route table and place it in the right resource group and region. Provide a clear name to easily identify it later. Select **No** for the route propagation unless you intend to set up BGP with another location in the future. Click **Review + Create** and then **Create**.

Once the deployment is complete, click **Go to Resource**.

In the route table, two things need to be performed. We need to add the appropriate client or server subnets, and we need to set up routing (which is actually more like traditional forwarding instead of actual routing).

In the **Route Table** resource, navigate to **Subnets** and click **Associate**. Select an appropriate server subnet and repeat if multiple subnets need to be added. Only add subnets that need to route their outbound connections through the trust interface of the firewall; do not select any of the data plane subnets of the firewall itself, as that may cause routing loop issues.

You can also add the **Mgmt** subnet, so outbound connections (dynamic updates, etc.) are routed through the firewall, so you retain full control of outbound connections. The **Subnets** sections will look similar to the following screenshot:

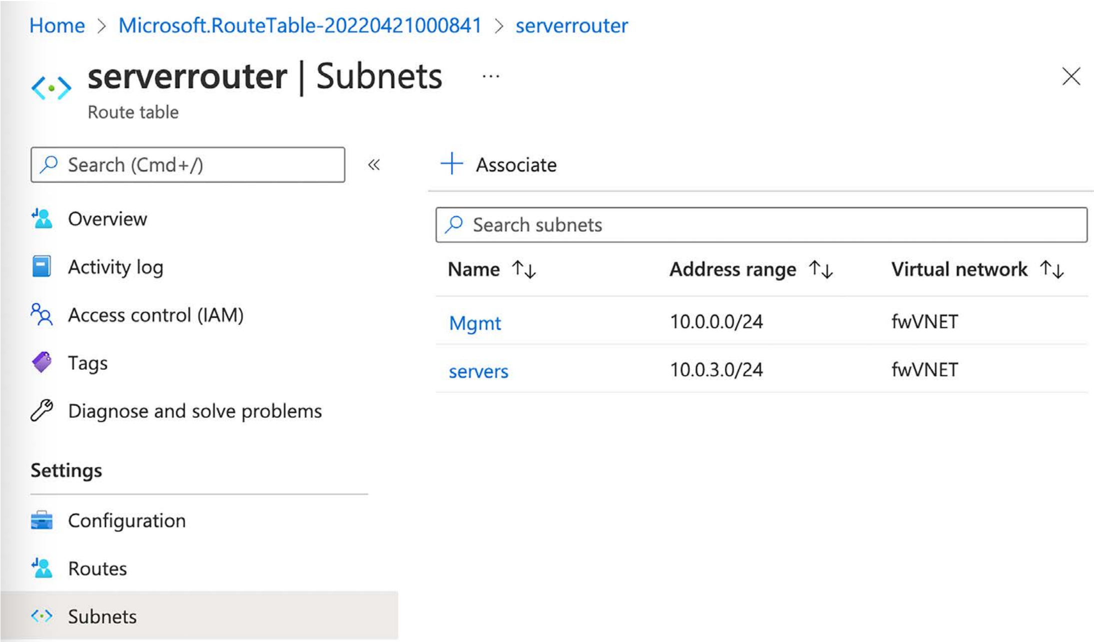


Figure 15.29: Adding subnets to a routing table

Next, navigate to **Routes** on the left menu and click **Add**. You will need to provide a route name and an address prefix source.

Address prefix source lets you use tags or IP addresses as the source; select **IP addresses.**

Next, you need to provide a **source IP address** or **CIDR block**. You can provide the individual host /32 IPs, the subnet IP range, or set `0.0.0.0/0` to include all hosts in all subnets that are added to the route table. This will only apply to subnets that were associated with the route table, so in most cases, using `0.0.0.0/0` is easiest (and allows adding more subnets later without needing to add additional route rules).

Next hop type has a few options:

- **Virtual network gateway** is the Azure version of an outbound gateway
- **Virtual network** allows you to forward to another VNET if peering was established between the two
- **Internet** applies the default action of sending outbound packets to the internet

- **Virtual appliance** forwards packets to a VM, like a Palo Alto firewall
- **None** creates a black hole route

Select the **Virtual appliance** option and then set the **Trust** interface dynamic IP address as **Next hop address**. In my case, this is the `10.0.2.4` IP, as you can see here:

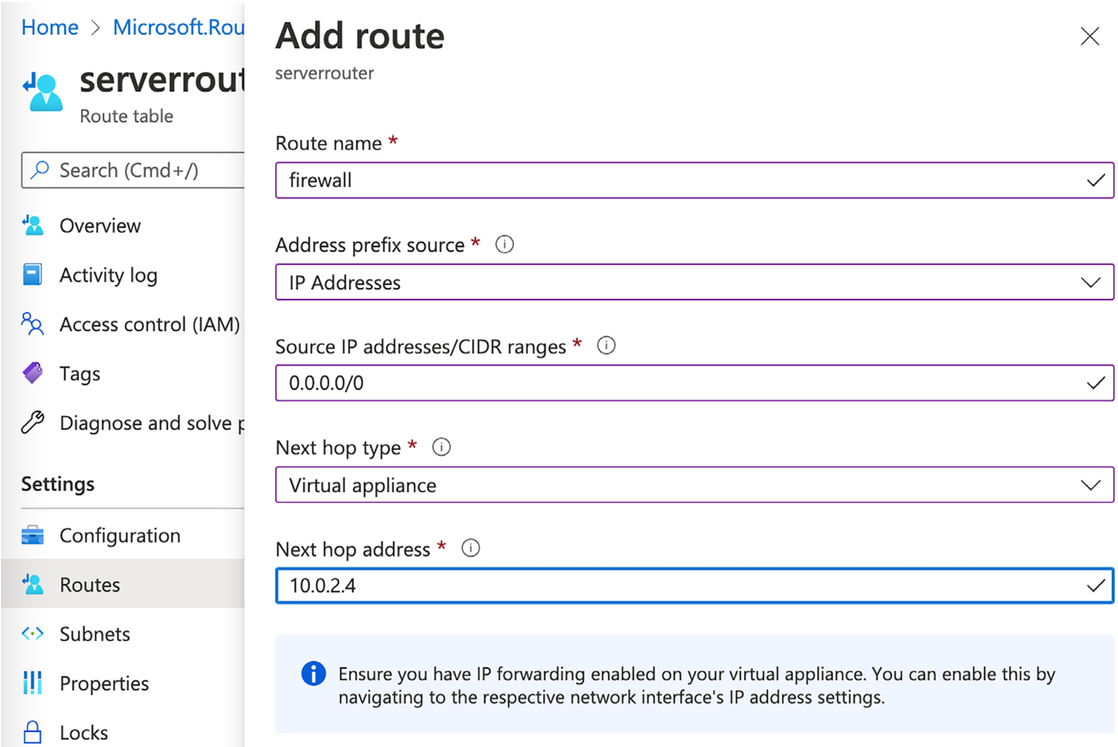


Figure 15.30: Adding a route to the route table

Click **Add** to complete the route, which will now cause all your servers to start sending their outbound packets into the firewall **Trust** interface. You can start building security rules!

Don't forget that NAT rules are applied via the **Untrust** interface's private IP. In my case, that would be `10.0.1.4`.

To spread inbound connections over multiple firewalls, we can also use load balancers.

Setting up a load balancer

Azure load balancers allow you to spread inbound and outbound connections over multiple VMs so you can more easily scale your environment and prevent one VM from being overloaded. To create a new load balancer, type `load balancers` in the search box and click the **Load balancers** service. Click **Create** to start a new deployment.

In the **Basics** tab, you should select the appropriate resource group and provide a name and region for the load balancer.

The **SKU** (**Standard** and **Basic**) determines the capabilities in terms of capacity for the load balancer. **Basic** is sufficient for a small environment for testing. Larger, more critical deployments should probably be set up with the **Standard** SKU to ensure sufficient capacity.

Type determines if the load balancer will be used for inbound connections from the internet or outbound connections from private subnets. A load balancer can't be used for both purposes at the same time, so an additional load balancer needs to be created if both types are needed. We will select **Public** for this example, as you can see in the following screenshot:

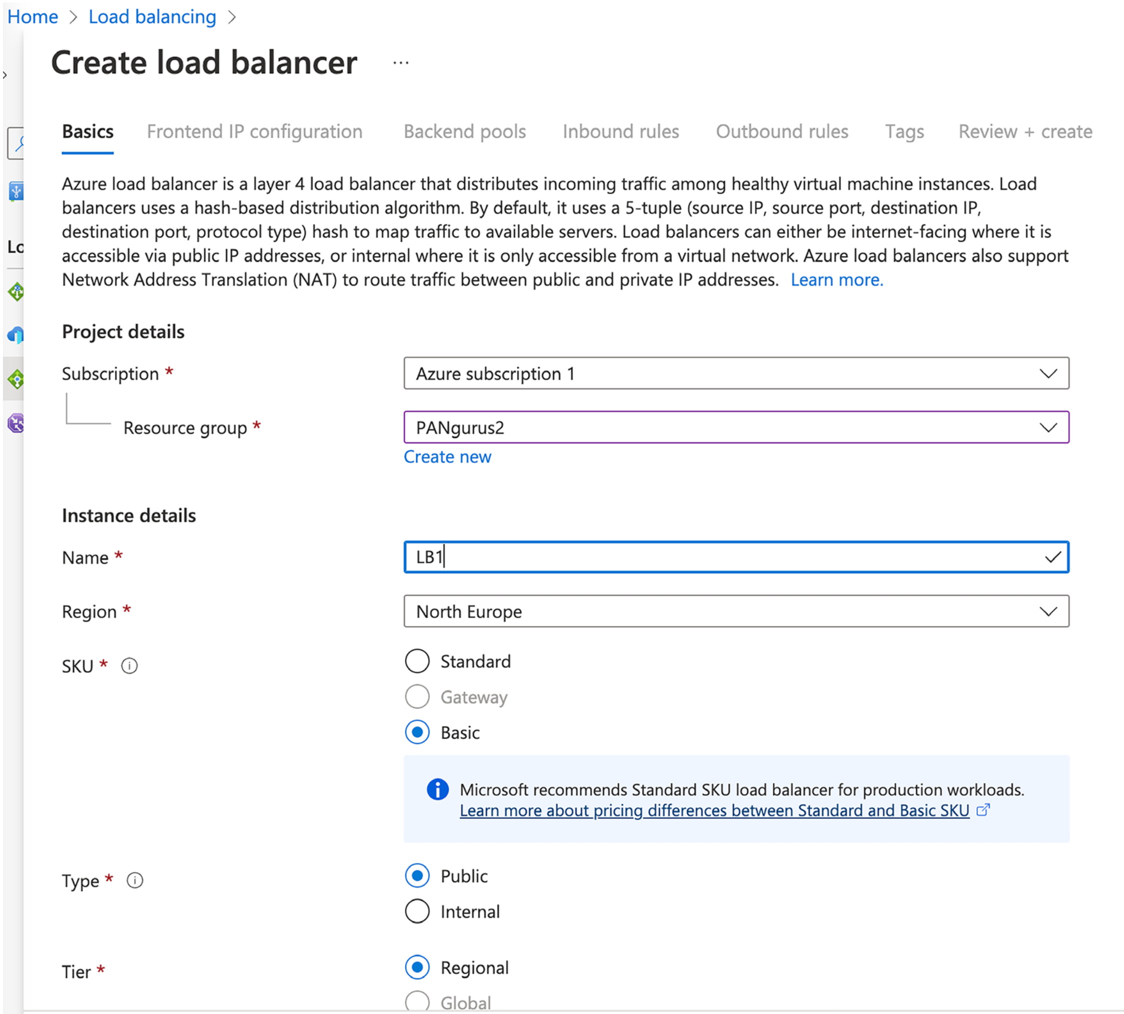


Figure 15.31: Load balancer basics

In the next tab, we can add a public IP to the load balancer. A public IP object may not already be associated with multiple resources, so dissociate your current public IP from the **Untrust eth1** interface so we can reuse it, or create a new public IP for the load balancer. Provide a name and select the desired public IP as illustrated:

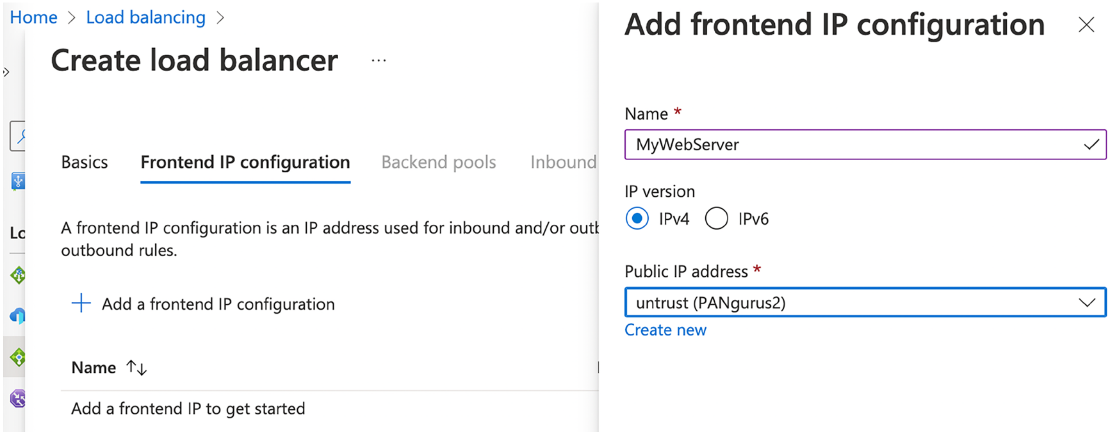


Figure 15.32: Adding a public IP

In the next tab, the backend pools are defined. These represent the destinations for the load balancer to distribute incoming packets. You need to set a name and attach the backend pool to a VNET.

Select to associate the backend pool with **Virtual machines**.

Once you click **Add**, you will be presented with a list of all the available virtual machines and all the IP addresses associated with them. Select the appropriate (**Untrust**) IP addresses for each firewall.

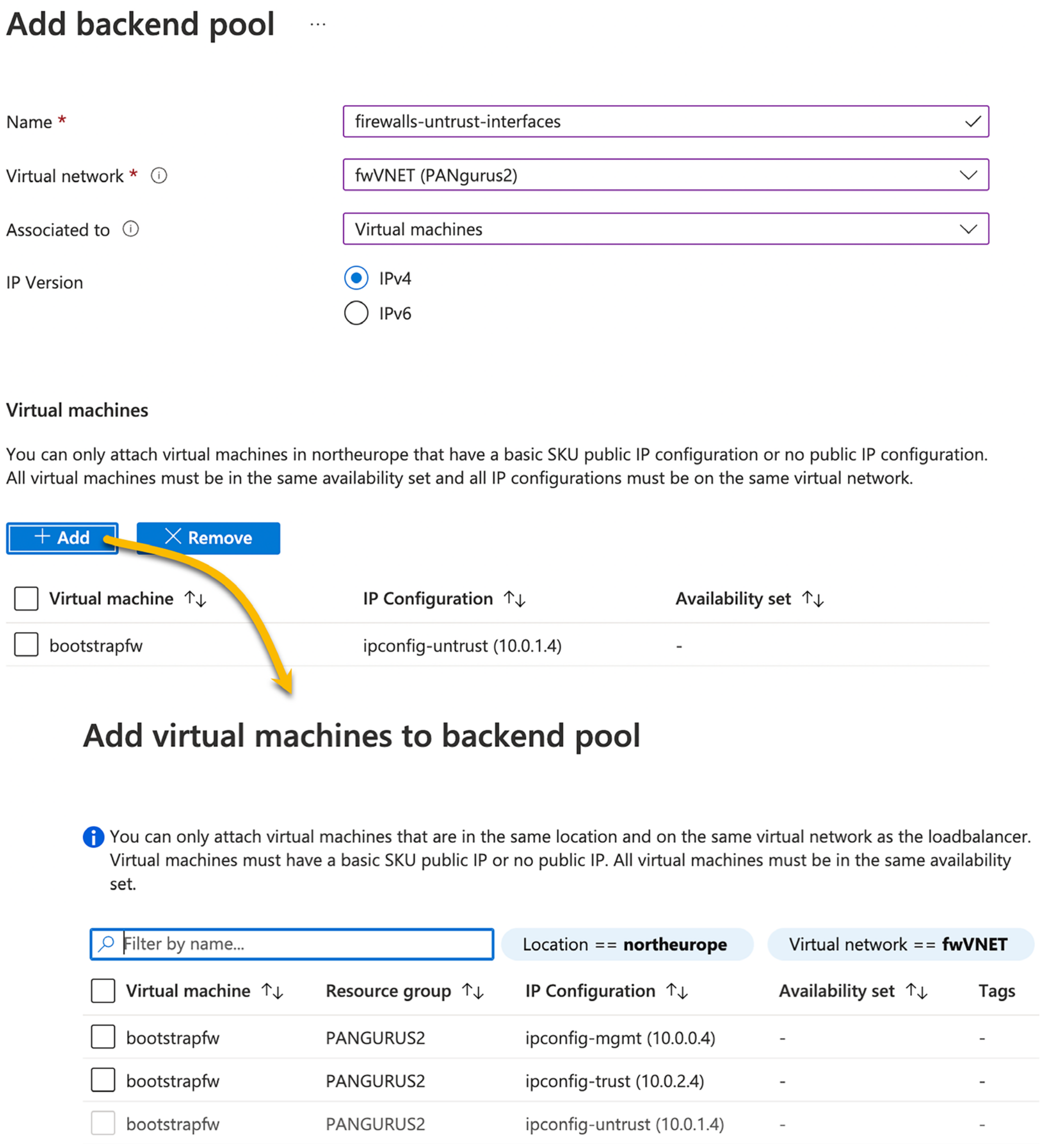


Figure 15.33: Adding backend pool IP addresses

Click **Add** and move to the next tab, **Inbound rules**. In the **Inbound rules** tab, you can define load balancing rules and inbound NAT rules.

1. Create a new inbound load balancing rule and provide a friendly name.
2. You must select a **Frontend IP address**, which is one of the public IP addresses that was associated with the load balancer, and select which **Backend pool** addresses will be included in this rule.
3. Next, you need to select the protocol to be used by this rule, which can only be **UDP** or **TCP**.
4. Set a **Port**: you can only set one destination port per load balancing rule, so if you need to distribute multiple destination ports, multiple rules will need to be created.
5. Set the **Backend port**. This can be a different port than the original destination port, or the same one.
6. A **Health probe** should be set. This is a monitor connection to all the participating backend pool VMs to ensure they are online and responding. If a probe fails, the VM is taken out of the backend pool, so sessions are not lost due to the resource not being available.
7. If no probe has been created yet, create a new one. Set a name and select which protocol will be used to probe (**TCP** or **HTTP**), and the port the probe will run on. Tweak the interval and threshold if you want.
8. Session persistence can be set to the following three settings:
   - **None**: No session persistence is used – a single session may be load-balanced over multiple backend pool hosts
   - **Client IP**: Each unique source IP's sessions will be forwarded to a single host in the backend pool
   - **Client IP and protocol**: Stickiness is determined by the source IP and protocol used for the session
9. Leave the **floating IP** disabled, as this is not needed.

The load balancing rule should look similar to the following screenshot if SSL on port 443 needs to be forwarded to the backend pool:

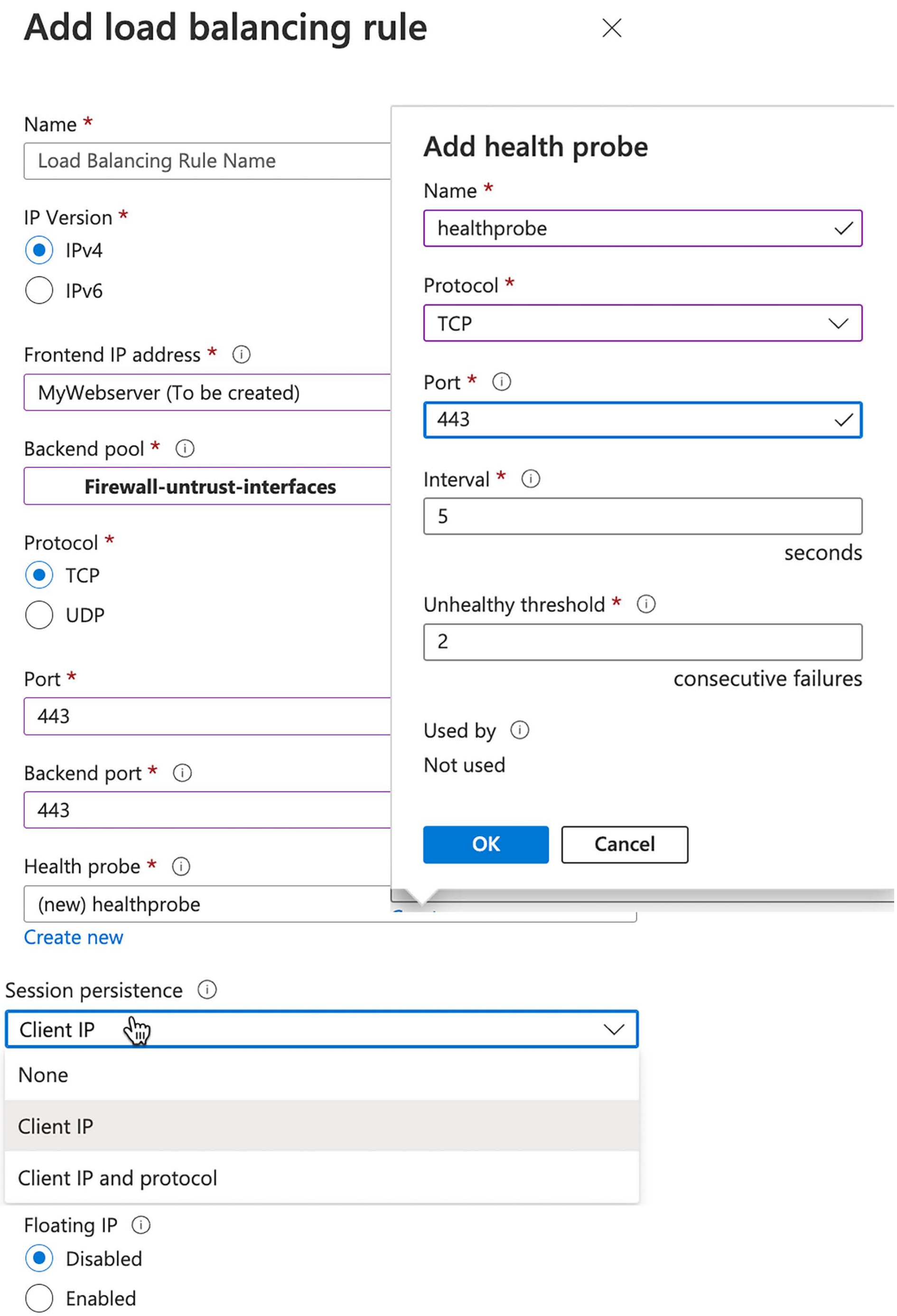


Figure 15.34: Load balancing rule

Inbound NAT rules can also be added that redirect certain frontend ports to a specific VM instead of a backend pool. This allows you to reuse the public IP for specific connections, like, for example, management access. In the following **inbound NAT rule**, external port **2222** is redirected to internal port **22** on IP `10.0.1.4` of the **bootstrapfw** firewall:

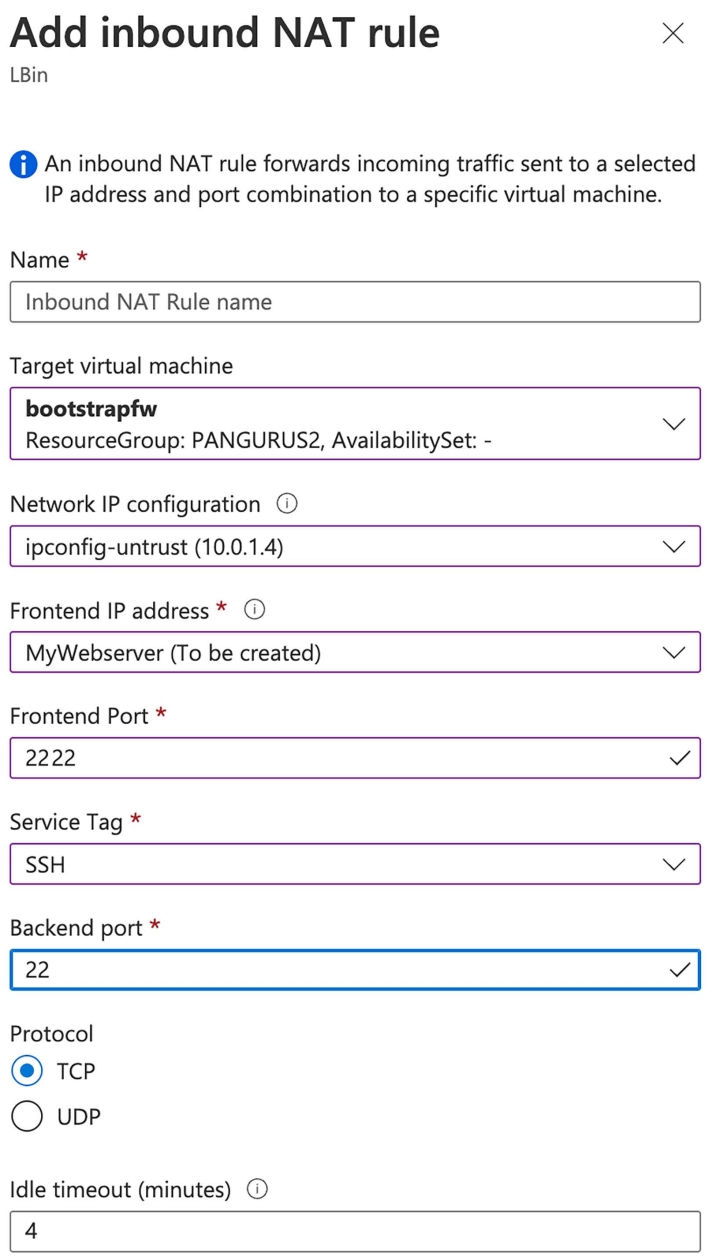


Figure 15.35: Inbound NAT rule

The next tab is the outbound rules. Since this is a public load balancer, no outbound rules can be created. On an internal load balancer, the inbound rules will be disabled, and similar rules can be created for the outbound rules. Click **Review + create** and then **Create** to deploy the load balancer.

Inbound connections on the public IP will now start to get distributed across the backend pool firewalls. Consider setting up Panorama so all firewalls can be configured with an identical configuration set. You can deploy one of the hardware appliances, install a VM on a datacenter server, or deploy one from the Marketplace in exactly the same way you would a firewall. The only flavor available, however, is BYOL. There is no PayGo option available, as you can see in the following screenshot:

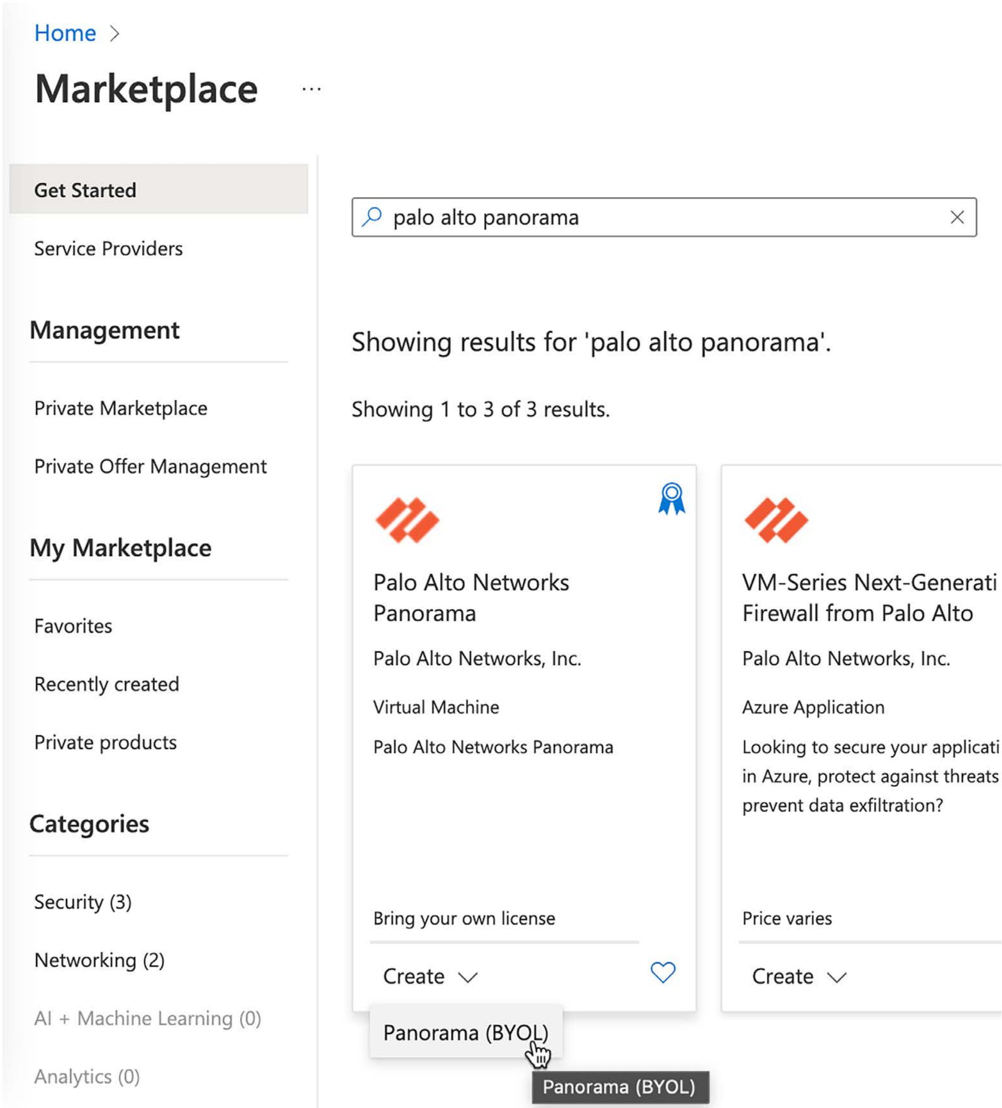


Figure 15.36: Panorama in the Marketplace

Once the deployment completes, you can reach the management interface from the newly created public IP, just like the firewall. See *Chapter 7, Managing Firewalls Through Panorama*, for further instructions on how you can add managed firewalls.

Summary

In this chapter, we covered the most important aspects of setting up a firewall in a cloud environment. We learned that there are different licensing models and how to standardize and streamline deployments by preparing bootstraps. We also looked at some caveats that will help you properly plan for routing and load distribution.

If you're preparing for the PCNSE, take note of the different licensing schemes.

Appendix

In this appendix, we are going to take a look at some “standalone” advanced features.

We’re going to cover the following main topics:

- Enabling the Advanced Routing Engine
- Activating cloud logging without centralized management

Enabling the Advanced Routing Engine

Starting from PAN-OS 10.2, an alternative to the default virtual router has been added that uses an industry-standard way of configuring methodology that can help reduce your learning curve, depending on your past experiences. This is the **Advanced Routing Engine (ARE).**

To enable the ARE, follow these steps:

1. Navigate to the **Device** tab.
2. Select the **Setup** section.
3. Go to **Management.**
4. Open the configuration for **General Settings**.
5. At the bottom, select **Advanced Routing**.
6. Click **OK.**
7. You will be asked if you are sure, and be warned that the firewall will reboot if you continue. Click **Yes.**

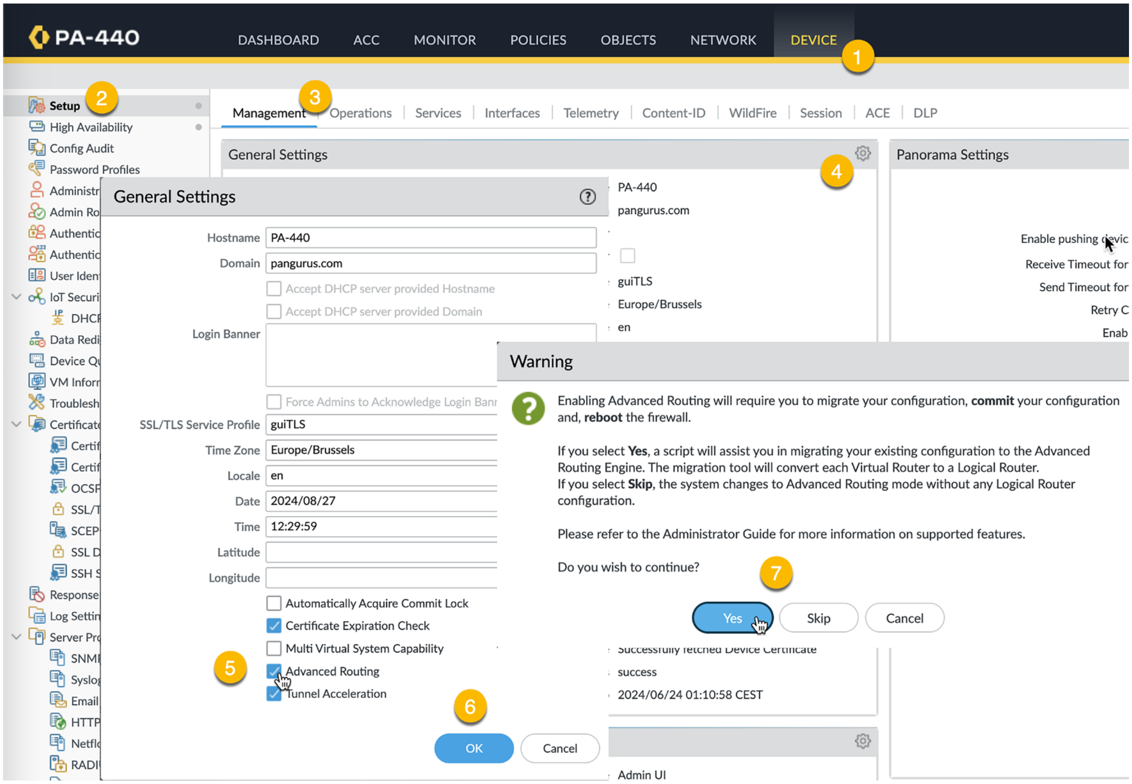


Figure A.1: Enabling Advanced Routing

After the firewall is rebooted, the virtual routers will have been replaced with the ARE. If you navigate to **Network | Routing,** you can now configure **Logical Routers** and add any routing profiles for dynamic routing under their own heading rather than inside a virtual router:

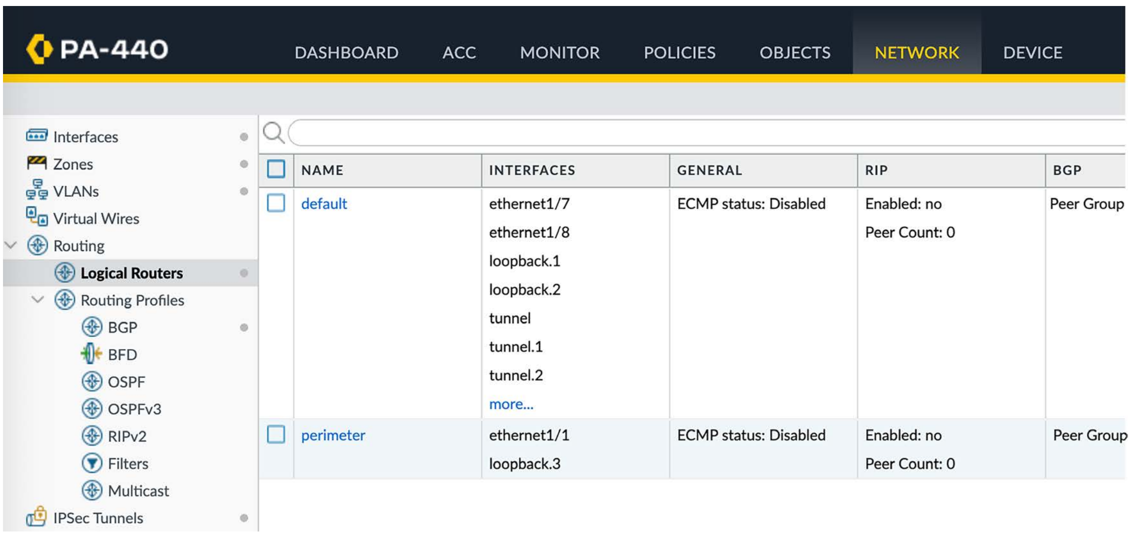


Figure A.2: Logical Routers

Activating cloud logging without centralized management

When deploying Panorama or Strata Cloud Manager, your firewalls will be enrolled in a centralized logging solution, but you can also enable Strata Logging Service on firewalls without centralized management. Just as we covered in *Chapter 8*, you do need to acquire a Strata Logging Service license and then deploy the SLS tenant from the activation link.

Then, you can access the hub at `https://apps.paloaltonetworks.com`, and from **Common Services**, select **Device Associations**:

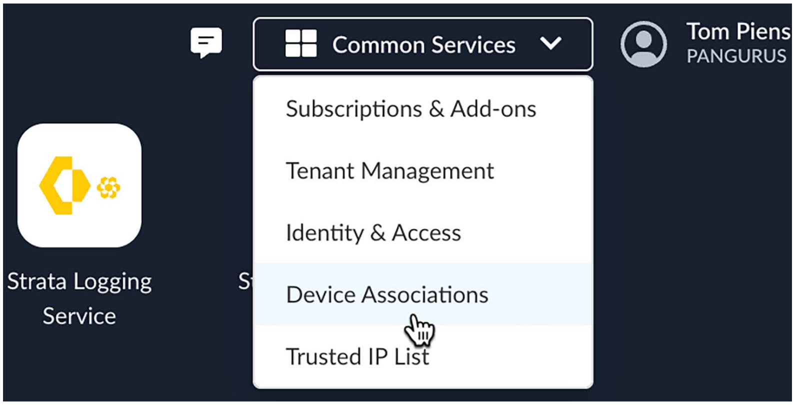


Figure A.3: Selecting Device Associations

In **Device Associations**, you can click **Add Device** to associate new serial numbers to the tenant, and then **Associate Products** for serial numbers with any products you have active in your tenant – in this case, **Strata Logging Service**:

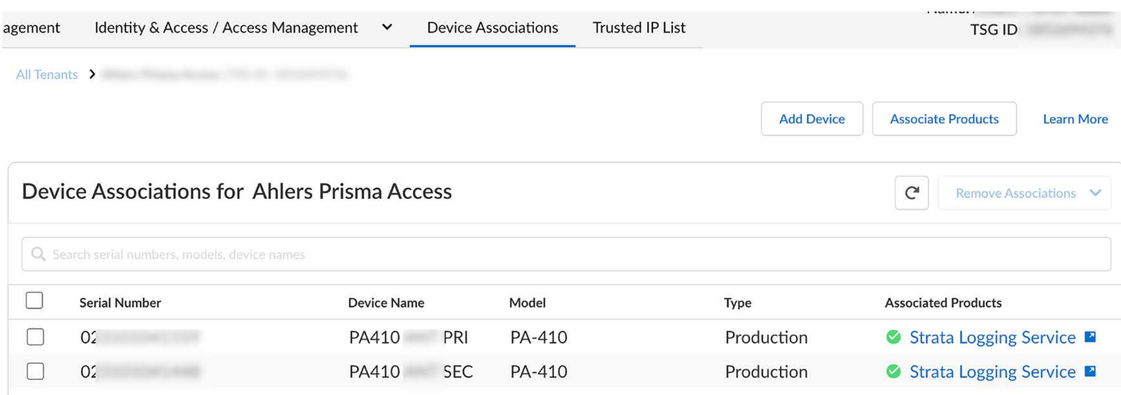


Figure A.4: Device Associations

To complete the connection, go to the device's web interface and navigate to **Device | Setup | Management** and scroll all the way down. There, you will find the **Cloud Logging** section, as you can see in *Figure A.5*. Open the configuration, select **Enable cloud logging**, select the appropriate region (the same as your Strata Logging Service tenant, which should be the only region available), and click **OK**.

Commit your change, and the firewall should start logging to SLS.

> The **Onboard without Panorama: Connect** button is only used when your firewall is running PAN-OS version 10.0 or earlier, but since this version is end of life, upgrade to a supported PAN-OS version first.

> Do not select **Enable duplicate logging** – this only applies to Panorama-managed firewalls.

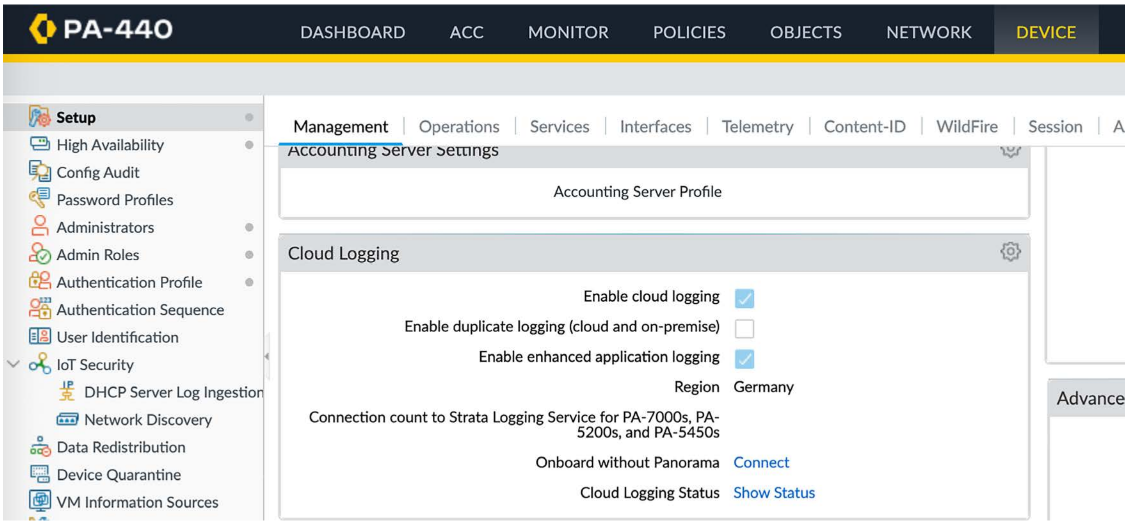


Figure A.5: Enable cloud logging

Next, let's take a look at how to troubleshoot some common issues.

Troubleshooting SLS connectivity

In some cases, the SLS connection may not work as expected. Here are a few troubleshooting steps to solve any issues you might encounter.

Prerequisites

To ensure the system is able to connect to SLS smoothly, there are a few requirements:

- Ensure the system has DNS servers configured that are able to resolve public domains in **Device | Setup | Services**.
- Ensure the system time and timezone are set properly in **Device | Setup | Management**, and NTP servers are added in **Device | Setup | Services** to ensure the time is synced.
- Ensure the device has internet access. It will need to have access to the following App-IDs and services (recommended: you should use **application-default**, but the service ports are listed for your reference):
    - Paloalto-logging-services
    - Paloalto-shared-services
    - OCSP
    - Paloalto-device-telemetry
    - Google-base
    - TCP/80
    - TCP/443
    - TCP/444
    - TCP/3978
    - TCP/5222-5224
    - TCP/5228-5229
- The device needs to have a valid device certificate, which you can check in **Device | Setup | Management | Device Certificate**. If the device certificate is not valid, ensure the above access is granted, and then refresh to fetch a fresh device certificate.

Testing connectivity

Once all the prerequisites are met and you've enabled cloud logging, you can verify if the system is able to connect to the SLS by clicking the **Show Status** button in **Device | Setup | Management | Cloud Logging**. It may show an error message as we see in the following screenshot. This one indicates the system has not been associated with the SLS tenant yet:

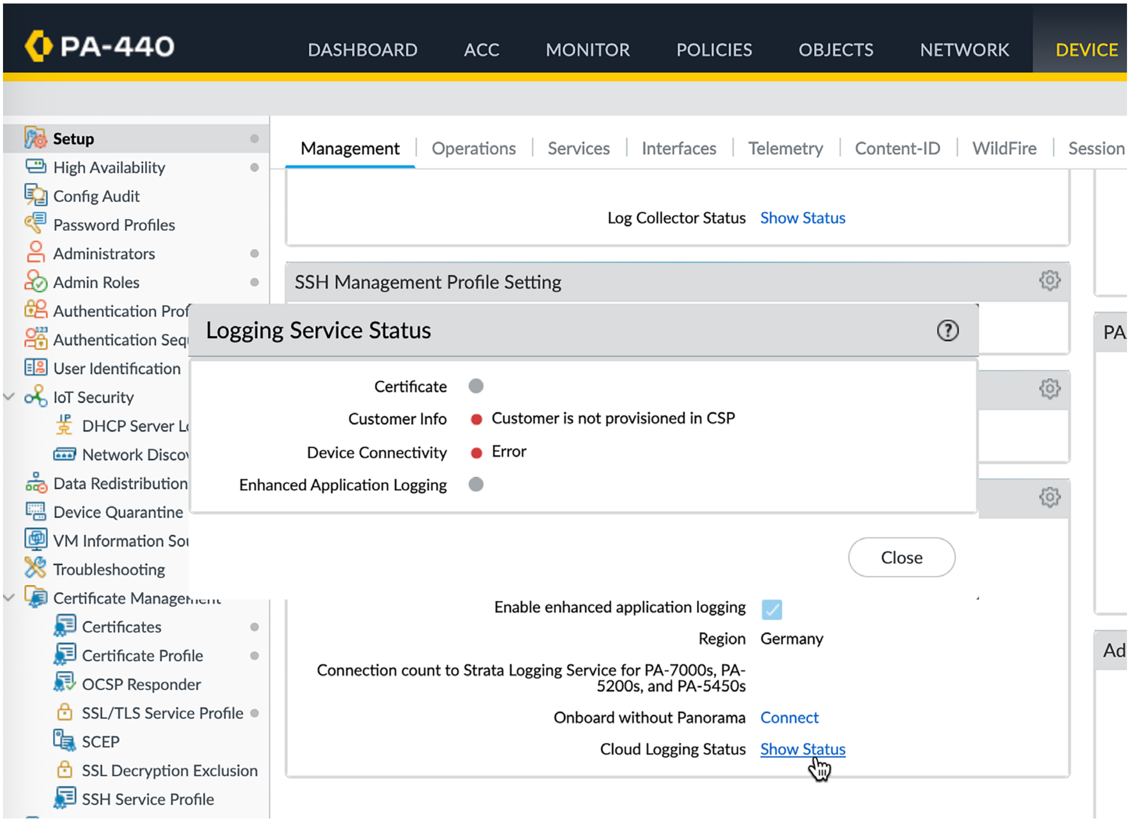


Figure A.6: Show Status

In this case, double-check **Device Associations** in the Palo Alto hub (`https:apps.paloaltonetworks.com`): **Common Service | Device Associations,** as we saw in *Figure A.3* and *Figure A.4*.

You can also run the following command from the CLI to verify connectivity to SLS:

```
request logging-service-forwarding status
```

A common issue after enabling cloud logging is that the communications certificate has not been installed on the firewall, so it is unable to establish a connection. You can manually fetch the certificate using the following command:

```
request logging-service-forwarding certificate fetch
```

In some cases, you may need to restart the management server for the change to take effect. To do this, run the following command to restart:

```
debug software restart process management-server
```

Once a connection has been established, the firewall will also appear in the SLS inventory:

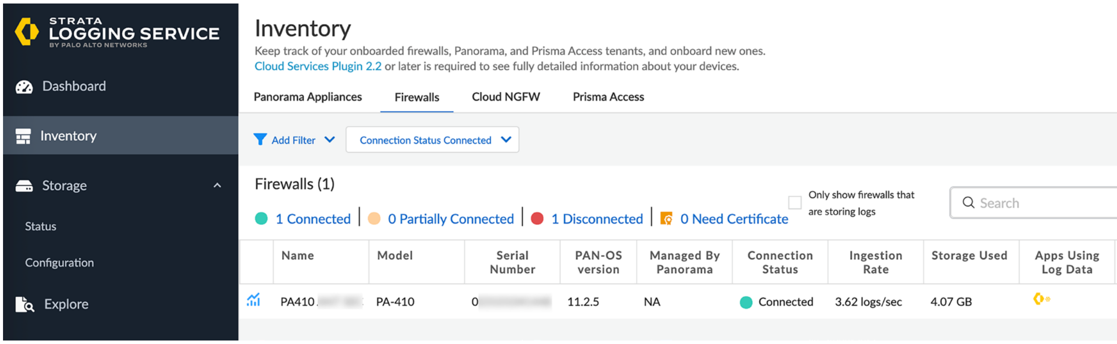


Figure A.7: Strata Logging Service inventory

No logs showing in Strata Logging Service

Once the connection has been established, there is one final issue that may cause logs not to show in the SLS **Explore** menu: the log forwarding profile has not been set up properly.

In **Objects | Log Forwarding**, make sure your log forwarding profile has been set so all logs have **Cloud Logging** enabled:

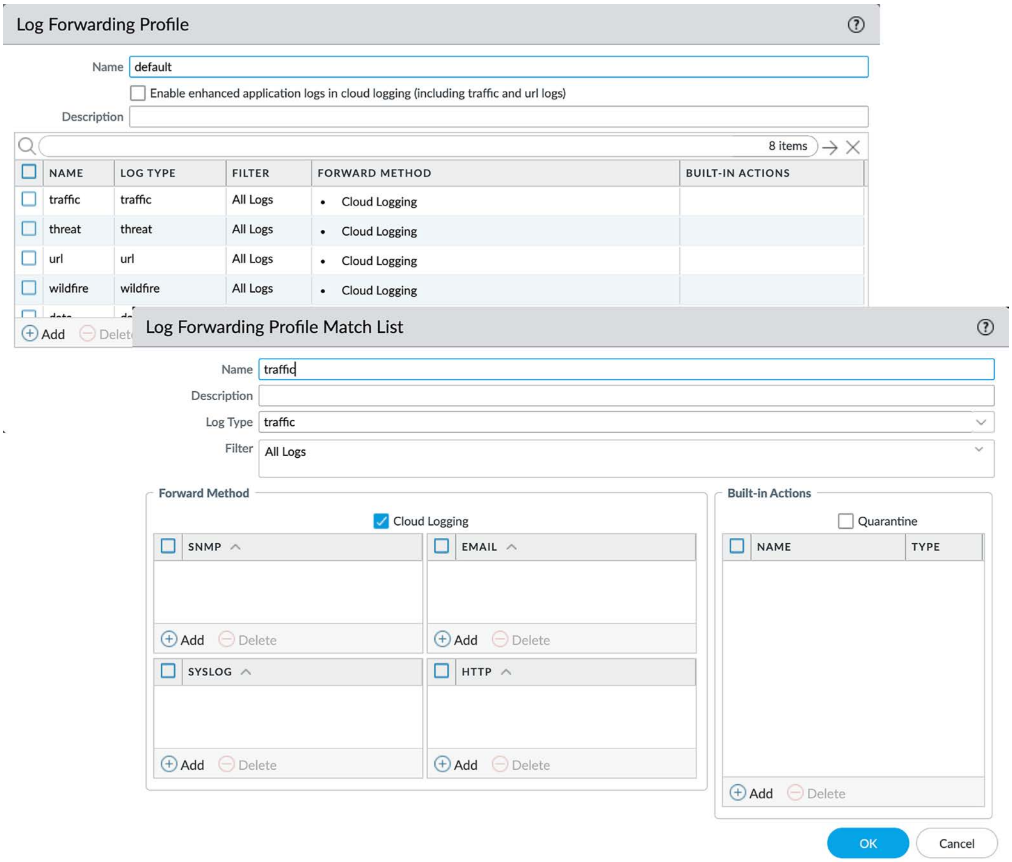


Figure A.8: Log Forwarding Profile

Lastly, make sure all your security rules have the log forwarding profile assigned to them:

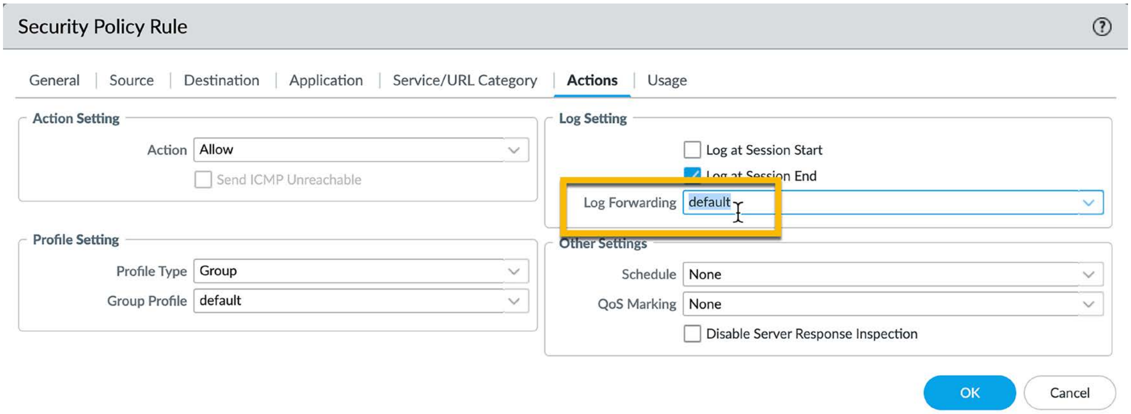


Figure A.9: Log Forwarding profile in security rules

You can now access the logs from the **Explore** section in the navigation menu.

Summary

You're now able to enable the Advanced Routing Engine so that you can benefit from industry-standard configuration methodologies, and set your firewalls to send logs to Strata Logging Service without needing to set up Panorama or Strata Cloud Manager.

Firewalls don't sleep, but with the right config, you might finally get some rest. If you've made it this far, you've wrestled with NAT, wrangled policies, and survived the labyrinth of logs and CLI commands. Take a breath – you've earned it. Stay sharp and stay secure.

packtpub.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

At `www.packtpub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Other Books You May Enjoy

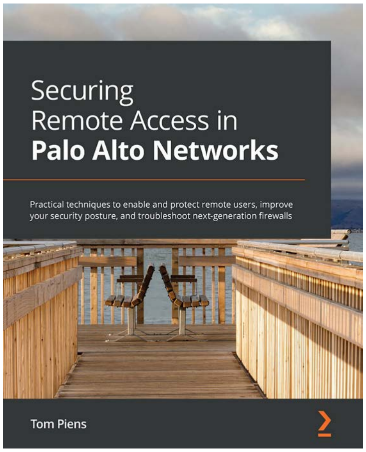
If you enjoyed this book, you may be interested in these other books by Packt:

Implementing Palo Alto Networks Prisma® Access

Tom Piens Aka 'Reaper'

ISBN: 978-1-83508-100-6

- Configure and deploy the service infrastructure and understand its importance
- Investigate the use cases of secure web gateway and how to deploy them
- Gain an understanding of how BGP works inside and outside Prisma Access
- Design and implement data center connections via service connections
- Get to grips with BGP configuration, secure web gateway (explicit proxy), and APIs
- Explore multi tenancy and advanced configuration and how to monitor Prisma Access



Securing Remote Access in Palo Alto Networks

Tom Piens Aka 'Reaper'

ISBN: 978-1-80107-744-6

- Understand how log forwarding is configured on the firewall
- Focus on effectively enabling remote access
- Explore alternative ways for connecting users and remote networks
- Protect against phishing with credential detection
- Understand how to troubleshoot complex issues confidently
- Strengthen the security posture of your firewalls

Looking for more cybersecurity books? Click `here` to browse our full catalog.

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Share your thoughts

Now you've finished *Mastering Palo Alto Networks, Third Edition*, we'd love to hear your thoughts! If you purchased the book from Amazon, please `click here to go straight to the Amazon review page` for this book and share your feedback or leave a review on the site that you purchased it from.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Index

A

B

C

D

G

H

I

K

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere?

Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily.

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below:

   `https://packt.link/free-ebook/9781836644811`

2. Submit your proof of purchase.
3. That's it! We'll send your free PDF and other benefits to your email directly.