
MIKKO HYPPONEN
IF IT'S
SMART, IT'S
VULNERABLE
WILEY

More Praise for *If It's Smart, It's Vulnerable*

"As the namesake of Hypponen's law, Mikko is the right person to explain how everything is getting connected, and what we should do to secure all devices on the internet — both the smart ones and the stupid ones."

—Robert M. Lee,
founder and CEO of Dragos Inc.

"A guided tour of the intersection between security and technology by the best kind of storyteller. Mikko lived it, shaped it, and now explains it."

—John Lambert,
Microsoft

"It's hard to understand a revolution when we are living in the middle of it. Mikko clearly explains the technology megatrends shaping our future and illustrates his points with fascinating real-world stories from his long career in infosec."

—Dave DeWalt,
Founder of NightDragon and former
CEO of McAfee and FireEye/Mandiant

"Mikko uses his remarkable experience to give the reader an understanding of the challenges we as digital citizens face. If It's Smart, It's Vulnerable; no truer words have been said in this digital age."

—Raj Samani,
Senior Vice President and Chief Scientist at Rapid7

"Before the internet, people were not used to reality having multiple dimensions. Now that the globe is being connected at an exponential pace, we are lucky to have talented guys like Mikko Hypponen explain how we can move to the next step."

—Marcelo Tas, Brazilian actor,
author and TV host

If It's Smart, It's Vulnerable

If It's Smart, It's Vulnerable

Mikko Hypponen

WILEY

Translated from “Internet”, published in Finnish in 2021 by Werner Söderström.
Translation to English by Mikko-Heikki Heinonen with Paul Anderson.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada and the United Kingdom.

ISBN: 978-1-119-89518-3
ISBN: 978-1-119-89519-0 (ebk)
ISBN: 978-1-119-89520-6 (ebk)

Library of Congress Control Number: 2022936101

Cover images: Mikko Hypponen, Background © nevodka / Getty Images
Cover design: Wiley

Contents

Foreword

Jeff Moss

The Internet is both a familiar, comfortable place and a bottomless rabbit hole you can lose yourself in. From its inception, the Internet has always been like. The difference now is that the scale and consequences are almost immeasurable, and it tests the limits of human imagination. When you look into the mirror of the Internet, what you see reflected back depends on what you are looking for. It has become largely a reflection of yourself.

If you are a businessperson, you might see endless opportunities, new markets, new supply chain efficiency, new ways to market a product or collaborate with designers you would never have had access to before. You would think about the costs, the risks of doing business online, what regulations you would need to abide by, and the financial rewards.

If you have an inquisitive mind, you may see endless research opportunities, different perspectives to consider, online debates to have, and meaningful places to contribute your efforts. You can do this no matter your race, ethnicity, gender, age, orientation, or religion. Such freedom!

Policy makers see a new domain that needs oversight and regulation that would shape the Internet into a reflection of their governments' values and ideals. Governments see not only a way to communicate and collaborate with their citizens but also a new way to spy on each other. No need for James Bond; instead

remotely break in and read their email to see if they are violating a treaty or planning an attack. Conspiracy theories thrive next to scientific research and fan-fiction sites, and criminals see a lower risk way to make money through theft or extortion—no need to expose yourself physically.

What's going on?

It has been said that in the 1700s the highest calling was the government, and it attracted the best and brightest. The 1800s saw the industrial revolution, and the 1900s attracted the best minds to Wall Street. Today it is the Internet and related industries that have the creative, financial, and emotional energy. How long will this phase last before people move on to genetics, nanotechnology, or even space?

The early days of the Internet were about building stuff: infrastructure such as networks and undersea cables to physically connect the servers. Next, it was bigger data centers, faster mobile technologies, and algorithms to try to understand and predict consumers.

By building better infrastructure, the creation of massive, global platforms became possible. Whole new business models were possible! Previously impossible global population tracking and influence campaigns were possible! This is the era when the Internet truly became pervasive enough to be a global good—and a global hazard. We are now in the age of Internet consequence, where what happens impacts billions of people a day and influences how nations behave, and there is little accountability.

One thing has become clear: Internet problems have become global problems, frequently larger than what a company or country can manage. It is now clear that these global problems require a global response. Trying to take down a botnet? Create rules for a social media platform? Set standards for the next generation of technology? All require participation in international forums and coordination in new ways from companies and governments

but also from civil society, academics, researchers, and users who have to live with the world we have created. We need to bring together not just the builders and the regulators but everyone involved.

The Internet I describe is constantly evolving, connecting. There is a gravity, a sort of invisible hand, that is organizing the Internet to be more optimized, complex, fragile, centralized, and attractive to regulation. There is the trade-off between efficiency and resiliency, and efficiency almost always wins because it is more cost-efficient. But efficiency is not always what is best for a society. The consequences have become so great it is now apparent that governments have a legitimate interest in what happens online, and regulation invariably follows unchecked innovation.

It is up to us, those who understand the technology, for the sake of the generations that will inherit the Internet, to work toward better policy outcomes, fair and transparent algorithms, and inclusive and accountable environments.

Jeff Moss is the founder of the Black Hat and DEF CON conferences.

Preface

The Internet is the best and the worst thing that has happened to us. The waves of the digital revolution are obvious everywhere in our everyday life. The Net brings us great benefits but also chilling new risks—risks which would not have even been possible before. And this revolution is still in its early days.

For over more than 30 years now, I've worked at the same company. First, at Data Fellows, which renamed itself to F-Secure. As I'm writing this, F-Secure is about to split into two companies: F-Secure and WithSecure. We have provided security software and solutions for companies around the world, and I've spent these years hunting online criminals. These 30+ years have been remarkable in technology history. It has been a privilege to live inside the Internet revolution.

I have not become unimaginably rich, but I've gotten the opportunity to see the firsthand effects of digitalization from the best seat in the house. I've been to places where few people go. I've met online criminals face-to-face. I've visited the headquarters of different national armed forces (such as the Pentagon) and the centers for different intelligence agencies. I've been in meetings at Microsoft, Google, and Apple, and I've given private briefings to the leadership teams of some of the biggest companies in the world. I've worked on online criminal cases with various police forces, and I've spent very long days at both Interpol

and Europol offices. I've traveled more than I'd like to admit; the glamour of travel starts to wear off when you sustain a level of 140 flights a year.

The places I've been and the people I've seen have illustrated to me that we all are unusually lucky. We get to live during a time where massive technological revolutions change the world faster than anything ever before. I'm watching these trends from within a cybersecurity lab. As we analyze online attacks and reverse engineer malware samples, we get a unique view into the Internet underground.

What will historians of the future think about us and about our time? How will we be written about in the history books of the future? We will be remembered as the first people who went online. We're the first generation that is living part of our lives online and part of our lives in the real world.

Mankind has been roaming this planet for more than 100,000 years before the Internet. Now the Internet will be part of our future, maybe forever. You and I just happened to be around when this happened. So, if the online world sometimes feels foreign or weird or difficult, it's OK: this has never been done before.

This global Net brings information to us, in a very concrete way. In the 1990s, we simply could not check or confirm things like we do today. This feels almost foreign now. If you had an argument at the bar about who won the gold medal in the marathon in the Los Angeles 1984 Summer Olympics, there was no easy way to find the answer. You might be able to check it the next day by visiting a library (the gold went to Carlos Lopes of Portugal). Nowadays, anyone will find the right answer in seconds with their iPhone.

This book is about the Internet and us. The Internet has changed from a technical curiosity to a seamless part of our everyday life. Eventually, it will become invisible, something we take

for granted and assume to be available always and everywhere. This is what happened with the electricity grid.

This book also tells about the things that threaten the future of the Internet: organized online crime gangs, governmental surveillance and censorship, and the fight over control of the Internet. It also tells about how law enforcement and intelligence agencies operate on the Internet, how money became data, and how we all carry a supercomputer in our pockets.

Mikko Hypponen
Helsinki, Finland
February 2022
@mikko

Saab 9000 Turbo

In my career, I have achieved more than I would ever have imagined. It is nevertheless good to stay grounded and remember that all experts' careers are characterized by failure.

During my first "proper" job at a software company, I was put in charge of an important customer project. I was 21 years old, and the customer was a ceramics factory in my home city of Helsinki, Finland. I was writing database software that would provide workers with instructions on various ceramics products. The software was fairly advanced by 1991 standards: it ran on Windows 3.0 and had a graphical user interface. It even had pictures of the various cups, mugs, and plates being made.

The project was extensive and kept growing during development. Schedules kept being pushed back. I spent the hot summer coding day and night in my student apartment, getting no closer to the finish line. Finally, the customer's chief information officer lost patience and demanded a meeting where I would confirm the project's stage of completion and demonstrate the software. I spent a nervous morning fine-tuning the software at our office, crammed my documents into my briefcase, and took a tram across Helsinki to meet the customer. When I arrived, the CIO and his assistants were waiting in a conference room with a desktop computer and large CRT screen for my demo.

I had barely sat down and opened my briefcase when I realized that I had left the floppy disk in my work PC's drive and did not have the software I was supposed to demonstrate. I confessed this to the CIO, and he went ballistic. In fact, he was sure that the software was nowhere near done, that this was an excuse to buy more time. Assuring him that I had the software on a floppy disk at the office, I proposed meeting again the next day.

Now even more annoyed than before, the CIO would not accept this. They would wait in the conference room while I fetched the missing disk. I agreed but warned them that the tram ride across Helsinki and back would take about an hour and a half. The red-faced CIO pondered the situation before giving me his car keys and telling me to use his car, a Saab 9000 Turbo, fresh off the showroom floor. Nervously, I set off for the streets of Helsinki—and crashed his new car.

When I told this story on Twitter, one of my followers asked how I ever got another job. I was happy to reply that, in fact, I didn't. The product card project I was working on in 1991 was with a small company called Data Fellows. The company forgave me, grew over the years, and focused on information security. Now, over 30 years later, I am still employed by them.

The application for the ceramics factory was eventually completed and was still being used in 2005.

The Good and the Bad of the Internet

We are living in an age of technological revolutions. The world is changing faster than ever. The Internet has shaped the world permanently, removing many geographical barriers, benefiting us massively, but presenting entirely new types of risks.

Prehistoric Internet

What is the Internet? It is the network of networks. It is a network built to survive nuclear war. It is a network that is older than most of us—and one that will be here after we are gone.

The history of the Internet began in the 1960s, when the U.S. defense administration started designing a new type of information network called the *Advanced Research Projects Agency Network* (ARPANET). Internet users may still encounter the name ARPANET. When searching for the sources of net addresses, you will occasionally see "in-addr.arpa" or "ip6.arpa." This refers to the original network where it all started.

The first router was connected in August 1969. Being the only device on the network, it was unable to send data anywhere. The first data packet was transmitted on October 29, 1969, when researchers from the University of California – Los Angeles (UCLA) and Stanford University tested a new connection for the first time.

The initial data transfer protocols were slow and error-prone, problems that were solved by the development of *Transmission Control Protocol/Internet Protocol* (TCP/IP) and Ethernet in 1973. These networking standards formed the basis of the Internet. TCP/IP ensured that data packets were transmitted from sender to recipient, and Ethernet standardized the way our devices receive these packets. It was often joked that Ethernet worked better in practice than in theory, but the protocols did actually do their job.

The world did not change in the 1970s, because almost nobody had access to computers. Mainframes were only accessible to universities and large corporations. The 1980s saw the introduction of 8-bit home computers, such as the Apple II and Commodore 64, but they lacked proper network access. One more component was needed to change everything: an open,

standardized personal computer. IBM, an IT giant known for its mainframes, introduced the IBM Personal Computer, or PC, in August 1981, with a central processing unit made by Intel and an operating system from a small startup called Microsoft.

Most importantly, the IBM PC was open—anyone could program it. Soon, hundreds of manufacturers were building compatible computers that could run the same software. The PC's operating system was not quite open source but was open enough. IBM itself was perhaps most surprised by the PC's success, as other PC manufacturers, such as HP and Dell, soon bypassed its sales volumes.

The PC was a unique, accidental success in terms of openness and standardization. Nowadays, we take it for granted that computers are compatible and open, and we are wrong to do so. In fact, most of our devices, such as cars, refrigerators, game consoles, and cameras, work in closed ecosystems. PCs are a happy exception to the rule. They gave rise to an open ecosystem that enabled freely developed software and accessories without limitations. Accessories included modems and network cards providing access to local area networks or bulletin board systems (BBSs).

The initially diverging paths of the Internet and PC revolutions developed a tendency to converge. One more agent of change was required. The study "Towards a National Research Network" of 1988 piqued the interest of U.S. Senator Al Gore. He began campaigning for federal funding of $600 million to research and develop applications for the information superhighway.

A year later, in Switzerland, Tim Berners-Lee developed the *Hypertext Transfer Protocol* (HTTP) and the *Hypertext Markup Lanuage* (HTML). These two combined to form the *World Wide Web*, or WWW for short. The first WWW server was coded by Berners-Lee, Ari Luotonen, and Henrik Nielsen.

However, the protocol and servers alone did not allow use of the WWW; users needed a browser. Browsers like the modern examples, Chrome and Safari, simply did not exist in the early 1990s. One of the earliest browser prototypes was developed at the Helsinki University of Technology. It was called Erwise. Tim Berners-Lee traveled to Finland to encourage the team to turn the browser into a product. Unfortunately, the project was never finished. One of the coders on the Erwise team was willing to finalize the browser as a summer project, but the university could not afford a summer intern. The Finnish browser was never finished.

The Mosaic browser was completed in 1993. For many people—like me—Mosaic was the first software to provide access to the brand new World Wide Web. Mosaic then branched off into an even more popular browser, Netscape, from which Firefox is directly descended. But who developed Mosaic? It was created at the National Center for Supercomputing Applications, supported by Al Gore's funding package.

As the 1990s progressed, Internet connections became common, and online services and shops multiplied. The first popular mobile phones lacked Internet connectivity, but its addition rapidly made phones the most common way to use online services.

The Internet shrunk distances. Young people now find the idea of long-distance or international calls strange. It is nevertheless true: before the Internet, the further you called, the more it cost. Nowadays, attending a Zoom meeting or sending a file costs the same, whether the other party is next door or across the world—pretty much zero, of course.

It is difficult even to imagine everyday life without the Internet. We were all given access to an open and unrestricted Internet. What kind of Internet we will be leaving for future generations is up to us.

The First Websites

The world's first three websites were in Switzerland, the United States, and the Netherlands. There were approximately 700 websites by early 1994.

Inspired by the wild prospects of the WWW, I set up a website for our company in April 1994, at the address datafellows.fi. Finland had only a couple of dozen websites, most of which belonged to universities or network operators. Nowadays, it would be impossible to create a list of all websites. There are almost half a million .fi domains alone, and more than 150 million .com domains.

Once I had set up our website, I proudly posted about the new service in the Internet newsgroup comp.infosystems.www.announce, giving details on how to access the website. In 1994, Windows did not support Internet and did not include a browser. The instructions provide a good picture of the effort needed to use the Internet back then:

> In order to be able to access the Data Fellows WWW site, you will need the following:
>
> 1. Internet access. This can be a direct line or a SLIP or PPP connection. Some Unix users can also use a Term Connection.
> 2. A TCP-IP stack. PC users need a stack with Windows Sockets compatibility. Suitable products are Microsoft Wolverine, FTP Software's PC/TCP, PC-NFS, SuperTCP, Trumpet Winsock, etc.

3. A browser. This can be, for example, NCSA Mosaic, Cello, WinWeb, or Lynx. Most of these products are available for a variety of platforms. If you don't have a browser, but are able to telnet, you can use one of the text-based WWW gateways.

Most or all of the needed programs are available for free from FTP sites.

The Data Fellows WWW site is a public site, and the service is provided for free: no login is necessary.

In spring 1994, there was no Google or Wikipedia online. Microsoft, IBM, and Apple had yet to set up their websites.

Our website initially attracted just a few visitors per day, but the numbers kept growing. When the website was a few months old, I accidentally destroyed it; I was maintaining the server and deleted the wrong directory. Only on the next day did I notice that our website was blank. Mortified, I had to call the chief executive officer and explain that I had accidentally deleted our website from the Internet…and that we had no backup. Luckily, the service was fairly easy to reconstruct.

Linux Is the World's Most Important System

Linus Torvalds coded his own operating system while studying at the University of Helsinki in 1991. The OS was initially called Freax (*freak unix*), but the server administrator named the directory Linux when it was placed in public distribution, a name so good that it stuck.

A couple of years ago, I had a conversation about Linux with an elderly relative of mine, who expressed regret that it never became the success story originally foretold, being unable to compete with Microsoft. He arrived at our meeting in his car, whose navigation and infotainment system ran on Linux. The Samsung Android phone in his pocket ran on Linux. I know that he spends a lot of time on Facebook, which runs on Linux servers—just like Google, which he uses for email and maps. His old desktop computer certainly ran on a Windows operating system, but Windows was far behind in all other areas. Linux is by far the world's most common—and most important—operating system.

The vast majority of web servers run on Linux. Special effects in Hollywood movies are created on Linux. Amazon's cloud services run on Linux. About 85 percent of the world's smartphones run on Linux. Chromebooks running on Linux outsell Apple's Macbooks, and SpaceX rockets and Tesla cars also run on Linux. Linux can even be found on Mars!

Even Microsoft has now faced the facts and supports Linux. In Microsoft's Azure cloud services, the user can choose one of the seven most common Linux distributions. In 2019, Microsoft announced that the Windows Subsystem for Linux enables Linux software to be run on Windows and has developed its own version of Linux. Linus Torvalds took on Bill Gates and won.

Linux does more than just power web servers and smartphones. It is also one of the world's most common platforms for

Internet of Things (IoT) devices. The most popular operating systems for smart TVs are Tizen, webOS, Android, Fire, and Roku. They are all based on Linux.

However, Windows and Linux are not the only operating systems. In many cars, for example, the user interface runs on QNX, an OS owned by Blackberry. BSD is a Unix operating system that comes in many variants, the key ones being Apple's macOS, iOS, and iPadOS. In practice, this means that all smartphones run on either Linux (Android) or BSD (iOS). The smallest IoT devices run on ultra-light operating systems such as Contiki, VxWorks, Nucleus, and RIOT.

Linux changed the world, and Linus Torvalds is the greatest Finn who ever lived. Our other national heroes are composers, soldiers, athletes, or politicians—but none come close to Torvalds. He has created more value and business than any other Finn—in just a few decades. Torvalds remains the key individual in the Linux project. The source code is open and free and is being developed by thousands of people around the world, but Torvalds supervises and approves changes made to the Linux kernel.

Torvalds has created another significant software project in addition to Linux: Git. Git is a system that allows the management of content being edited by several people at the same time. The most typical application is the management of software code—Git is currently used to manage Linux and Windows source code, no mean feat given the millions of lines of code in systems of their size.

Git was so important that it led to the creation of several new companies, such as GitHub, Bitbucket, and GitLab. It now has dozens of millions of users, which is one of the reasons why Microsoft purchased GitHub for $7.5 billion in 2018.

It should be noted that Microsoft did not purchase GitHub from Torvalds. Torvalds has made the idea and source code for Git free for anyone to use. The founders of GitHub Inc. did just

that and later sold their company for billions of dollars. Linux has been released for public distribution in a similar way, and large corporations such as Google, Amazon, Siemens, and Boeing use it to make massive profits.

Linux would never have become a global success story if subjected to license terms or limitations or if its innovations had been patent-protected. *Wired* magazine once named Torvalds the leader of the free world.

Torvalds has received substantial acclaim. Asteroid 9793 is named after him, and in 2000, *TIME* magazine named him one of the 20 most important people of the last century. He is also one of only three Finns to give a TED talk.

iPhone vs. Supercomputer

The pace of technological development is astounding. Modern smartphones contain staggering amounts of computing power. In fact, an iPhone is faster than the fastest supercomputers of the 1990s.

At the end of the last century, the Cray-2, Thinking Machines CM5, and Paragon XP/S were among the fastest supercomputers. These van-sized behemoths needed water cooling systems. They cost millions and devoured electricity, but they enabled long-term weather forecasts and strength calculations for skyscrapers.

Now, 30 years later, everyone has the same computing power in their pocket.

As well as packing serious computing power, our phones went online. Some readers will recall the days when mobile phones were for calls and text messages only. By 2021, more web traffic was being sent by mobile phones than computers. Finland is the world's number one in mobile data: 4G and 5G connections are very fast and cheap in Finland, and there are no data caps. The mobile network is the only Internet connection for many single-family homes and is commonly used to watch movies on Netflix in 4K resolution. Soon, this will be the reality everywhere.

Online Communities

Each year, *TIME* magazine recognizes the Person of the Year. In 2006, that person was "You." Social networks such as Myspace and online services such as YouTube were growing, and *TIME* had sensed a pending revolution in how we use media. People were no longer passive recipients watching TV or reading papers; now, anyone could share their opinions and views with the world.

However, this revolution has had its downsides: paranoid nutjobs and conspiracy theorists also had a chance to be heard.

The Web has been more transformative for minorities than for the mainstream population. In the 1990s, the only trans person in a small village felt very alone. Going online allowed them to find support and safety among peers. For sexual minorities, the Web can be a lifeline. In the same way, anyone with a rare hobby can find peer support, safety, and happiness online—model steam train collectors, for example, can effortlessly create worldwide online communities. This is the upside of online communities.

However, people with destructive fantasies, suicidal tendencies, and eating disorders—even potential school shooters—can also find validation online. This can be seen in the conspiracy theories and false information spread on the Internet. New rumors spread from one country and language to another faster than ever. The Web is more efficient at spreading information than any other means in history. Information can be useful or harmful, and we cannot prevent the spread of harmful information.

When the Internet became generally available, for a while it almost seemed utopian. We had created an open and free network that the entire planet could join, free of charge and limitations. No distances, no borders, no geography: one, unified world.

Now that we have lived the online life for a few decades, we realize that the Internet is no utopia. In many respects, it is more like a nightmare. The Internet is a reflection of the real world

and shares in its wickedness and greed. It has become a place where criminals can seek victims across the globe, where elections are won and lost due to online influencing, and where fake news spreads faster than facts. Our real-world conflicts and wars are also expanding online.

Money Is Data

The Internet took money transactions online a long time ago. I paid my first bills with an online bank in 1990, using my 386DX-powered MS-DOS machine and 2400 bps modem to call text-based banking services.

Modern consumers see less and less cash. Many Asian countries, such as Japan and mainland China, have been pioneers in replacing cash. Even the smallest street vendor stall in Shanghai will have the WeChat or Alipay payment system for mobile payments.

In Europe, Sweden has been the pioneer. According to the Riksbank, Sweden's central bank, printed money makes up less than 1 percent of Sweden's annual gross domestic product, compared to 8 percent in the United States. Sweden has already gone as far as enacting a new law, in early 2020, requiring the largest banks to ensure the availability of cash. Cash continues to play an important role, particularly in emergencies. How else will you pay if the Internet or power grid fails?

Codes All Around Us

Nowadays, barcodes can be found in nearly every product we buy in stores. Developed in the 1950s, barcodes were originally based on Morse code. The square-shaped QR codes only became commonplace in the 1990s.

The Japanese spare parts manufacturer, Denso Wave, developed the quick response (QR) code for marking spare parts packages in 1994. Compared to the traditional barcode, the main benefit of the QR code is that it can contain any data: text, numbers, contact information, login information for a Wi-Fi access point, or web addresses. The QR standard supports QR codes of up to 2,953 bytes in size (luckily, you never see codes this massive). QR codes are even carved into tombstones.

In 2007, coder Justin Watt wrote an article on QR codes on his blog, with a sample image linked to his website, justinsomnia.org. Google published an updated version of its image search in the same year. The image in Justin's blog became the top result for the search term *qr code*. This led to a surprising outcome: Justin's QR code appeared on posters, T-shirts, and TV commercials around the world. Wanting to add a QR code to their productions, designers and advertising agencies had used the first QR code they could find as a placeholder for the design stage—that is, the one that was the top result on Google Image Search. Unfortunately, because barcodes and QR codes all look the same to us humans, the placeholder QR code (linking to the wrong website) was surprisingly often retained in the final advertisement or poster.

This happened to several well-known companies, such as Nokia, PayPal, and Blackberry. Justin managed to cash in on this by redirecting the advertiser to the correct page—for a fee. To the advertiser, this was easier than, say, printing and distributing thousands of posters all over again.

QR codes and scanning them have met with derision for many years, but the Covid-19 pandemic and covid passports have made them an everyday tool in many parts of the world.

Even the credit cards in our wallets contain a surprising range of codes. In the early days, their embossed credit card numbers were transferred using carbon paper and an imprinter upon payment. Later, a magnetic stripe, then an EMV chip and, finally, an RFID circuit made contactless payments possible. Many modern credit cards have all of these, making new credit cards compatible with the same "knuckle buster" imprinters used to process payments in the 1950s!

Barcodes, QR codes, and credit card markings are easy to spot, but we are also surrounded by countless invisible markings, such as the yellow ones created by printers or the EURion symbols found on paper money.

Practically all color printers leave a unique fingerprint on their output: they print nearly invisible, light yellow dots on every page. These dots give the time of the printing and the printer's unique identifier. In other words, if you print out controversial opinions, you could, in principle, be traced.

In 2017, there was a leak at the U.S. National Security Agency (NSA). An NSA employee had printed out information on Russia's attempts to influence the presidential election of 2016, and the documents were mailed to *Intercept* magazine. *Intercept* scanned the papers and published them on the Internet. The yellow dots left by the printer were difficult to see in the online materials, but they could be used to decode the key information: the informant had printed the classified information on printer 29535218 at 6:20 a.m. on May 9. The NSA's internal investigation team almost immediately arrested the 26-year-old Reality Leigh Winner, who was sentenced to five years in prison.

EURion is a pattern of five circles that has appeared on paper money in many countries since 1996. Resembling a constellation, this symbol is commonly hidden in the middle of other circles or similar symbols. However, look closely, and you will find it on every dollar bill and on euros, pounds, Chinese yuan, and Swedish crowns.

Most image-processing software and photocopiers will not process or copy an image that contains the EURion pattern, which makes counterfeiting more difficult. I once spoke with a man at a hacker conference who said he was considering having his face tattooed with the EURion pattern. This might make it difficult to process photos of him.

Money is data. Bank robberies were once a major problem. Heists, where armed robbers entered a bank and demanded cash, were still common all over the world in the 1990s. They are now consigned to history. Bank branches have all but disappeared, and those that remain hold minimal amounts of cash. Online bank heists are now the norm. Digitalization has changed the nature of bank robberies—like everything else.

Geopolitics

The Internet has been firmly controlled by the United States, the traditional seedbed of most popular services. People around the world use services—such as search engines, cloud services, and social media services—built in the United States.

This is actually somewhat counterintuitive, since the United States stopped being a major player on the Internet some time ago in terms of user numbers. More than half of all Internet users come from Asia. Europe is the second-largest area, with around 15 percent of users. Africa and Latin America each account for 10 percent, while only 6 percent of users come from the United States.

Europe seems to be a bigger player than the United States on nearly every count. It has twice the population and, based on national budgets, is a much larger area of economic activity. However, when we look at company valuations, the United States and Europe could be on different planets. The total market value of publicly listed technology companies in the United States exceeds that of all European publicly listed companies. In 2022, Apple alone is larger than all public companies from Germany.

It is easy to list American technology companies like Google, Facebook, Microsoft, Amazon, and Apple. Chinese technology companies are also becoming familiar: Huawei, Xiaomi, Alibaba, Tencent, Lenovo…. But what about Europe's largest technology companies? This often becomes a list of sold European companies, such as Skype, Supercell, and Mojang (Minecraft).

GDB listed Europe's largest technology companies. How many names do you recognize?

1. Accenture
2. SAP
3. Capgemini

4. ATOS
5. T-Systems
6. Computacenter
7. Amadeus
8. Micro Focus
9. Spotify
10. Sopra Steria
11. Bechtle
12. Indra Sistemas
13. Dassault Systèmes
14. Agfa-Gevaert
15. Wirecard

Over the years, the United States has become used to its unrivaled position in the online world, making it difficult to adjust to no longer controlling all technology platforms. The United States has no notable manufacturers of 5G networks, and the world's most downloaded mobile application is no longer from the U.S.

The United States must now accept the reality that Europe has been living with for longer. Our technology platforms and applications come from far away, and their authors have little interest in our wishes, culture, or rules. Going forward, this will be increasingly true for the United States.

This shift is most visible in mobile applications. Consulting house Sensor Tower listed the highest-earning mobile applications for 2021:

1. TikTok (China)
2. YouTube (United States)

3. Piccoma (Japan)
4. Tinder (United States)
5. Disney+ (United States)
6. Google One (United States)
7. Tencent Video (China)
8. iQIYI (China)
9. HBO Max (United States)
10. LINE Manga (Japan)

As we can see, half of the most successful mobile apps already come from Asia.

According to Amazon's Alexa Internet service, the world's most visited web addresses are as follows:

1. Google.com (United States)
2. Youtube.com (United States)
3. Baidu.com (China)
4. Facebook.com (United States)
5. Qq.com (China)
6. Taobao.com (China)
7. Zhihu.com (China)
8. Amazon.com (United States)
9. Yahoo.com (United States)
10. Tmall.com (China)
11. Wikipedia.org (United States)
12. Bilibili.com (China)
13. 163.com (China)
14. Weibo.com (China)
15. Zoom.us (United States)

16. Live.com (United States)
17. Sina.com.cn (China)
18. Sogou.com (China)
19. 360.cn (China)
20. Sohu.com (China)

China is a rising power online, and this is only the beginning. China's gross domestic product is growing at a staggering rate. It will catch up with the United States in a few years, bypassing Europe shortly after. China is becoming King of the Hill.

Security Tetris

Everyone knows the game Tetris. Once you have built an entire line of blocks, it disappears. Working in information security is sometimes a bit like playing Tetris: your successes disappear but your failures accumulate. When information security works flawlessly, it is invisible. And rarely is anyone thanked for stopping a disaster that didn't happen.

At the turn of the millennium, many information systems were overhauled to fix the so-called Y2K bug. A lot of software was using only two digits to store the year in a date: 85 stood for 1985, and 95 for 1995. How would these systems treat numbers like 00 or 22?

Large companies spent years on the issue, combing through their software and fixing a countless number of bugs. The media eventually found out, and in late 1999, magazines were full of lurid headlines about how the turn of the millennium would cause computer problems: systems would crash, and the world would go dark.

When January 1, 2000, arrived, the same magazines wrote—with great disappointment—that nothing had happened. In other words, the massively successful global information technology project somehow became a failure.

Not all problems were found and fixed in time, of course—they simply went unnoticed at the turn of the millennium. One of the saddest consequences occurred in Great Britain. The Down syndrome screenings for pregnant women gave opposite results due to Y2K. This error was not discovered until the summer of 2000. Due to this bug, several mothers bearing children with Down syndrome were told that their child was healthy, and many mothers with healthy children were told the opposite. As a direct result, the Y2K bug caused many unnecessary abortions.

The year 2000 was a big problem, but it's not the only one of its kind. January 19, 2038, will be the next major stumbling block. As one of their time-tracking systems, Linux systems use the counter time_t to count the number of seconds that have elapsed since the first second of the year 1970. At the time of writing, 1,618,060,916 seconds have elapsed—just over 1.6 billion. Linux systems traditionally stored this number as a 32-bit integer. The largest number that can be stored in this way is 2,147,483,647—some 2.1 billion. This number of seconds will be reached at 3:14 a.m. on the morning of January 19, 2038. Feel free to check your calendar app: a lot of them won't let users browse past this point in time.

The good news is that a new version of the Linux kernel, which came out in 2020, fixed the problem. Linux kernel v5.6 switched the time_t counter to 64-bit. It can count seconds for several trillion years. The bad news is that some of the currently used Linux versions will still be in use in 2038, and some related problems will appear long before 2038. This could already happen if, say, a program calculates mortgage interest 20 years into the future. When time counters suddenly wrap around into the past, unexpected errors will occur. However, the best thing about this problem is that I will have retired in 2038.

By the way, Windows does not have this limitation regarding time. The counter currently used by Windows will end in the year 30828, which will do nicely for now.

I often visit our clients, providing management teams and boards with status updates. Occasionally, a member of the team, usually the chief financial officer, will ask me a difficult question. Glancing at the annual budget, the CFO will ask why the company is paying so much for information security each year,

since they have no IT security problems. At one point, I was somewhat snarky in my response. I glanced around their conference room, complimenting them on its cleanliness. The CFO looked puzzled. I repeated the compliment and added that they could probably lay off the janitor and housekeeping staff, given how clean everything was.

Information security is like Tetris. Your successes disappear, but your failures do not.

Who Are We Fighting?

The Internet has made geography irrelevant—the time is nearing when we are more likely to be victims of online than real-world crime. This is especially true in countries where crime rates are low.

Before the Internet, pickpocketing and car burglaries were the most common crimes targeting Joe Average. Most perpetrators of such crimes are from nearby. Pickpockets do not fly halfway across the world to steal your wallet. However, as we leave the real world and move online, we need to concern ourselves with, say, Brazilian or Vietnamese online attackers. In their eyes, we are as nearby as any other victim anywhere in the world.

Defense against unknown enemies is difficult. To protect our information systems, we need to know who we are fighting and why they are attacking us. Our efforts will be in vain if we do not know our enemies.

Schoolboys

In the early days of computer viruses, the problem was schoolboys who wrote malware for fun. They had no ulterior motive but simply wanted to see how much damage they could do or how far the virus they created would spread. Why were only boys caught writing viruses? My guess is that girls were writing viruses too, but they were too smart to be caught.

Since information networks were very rudimentary, viruses were spread on floppy disks. In 1992, we received a sample of a new, previously unknown virus. It arrived on a floppy disk by mail. I decompiled the malware and wrote the following description:

The Cinderella II virus was discovered in late 1992. It is partially based on the Finnish Cinderella virus. These viruses may be from the same author. The virus has only been discovered in Finland.

Cinderella II remains active in memories, infecting nearly all COM and EXE files that are run. The infected files grow by 783 bytes. The virus does not change the timestamp of the files it infects.

It activates after infecting one thousand files. When the virus does activate, it attempts to wipe out data on the hard drive, but may fail due to a programming error.

The author's apparent purpose was to carry out the assembly language instruction INT 13, AH=03h, an absolute write to disk. The targets of the write are the hard drive, master boot record (MBR), and partition table. This would have prevented the computer from starting. However, the programming error means that it does not work.

The virus then endlessly prints the text "Cinderella II", crashing the computer. This text is encrypted with the XOR operation and cannot be seen when viewing the virus code.

In all, Cinderella II is a fairly functional virus. Finnish virus authors seem to be developing their skills.

Cinderella and Cinderella II spread on bulletin board systems. I traced the spread routes and gathered clues on the virus's origins. Finally, I found the user ID that had sent the first version of the malware to a large BBS, spoke to the system operator, and convinced them to give me the related phone number.

I called the number the next day. A middle-aged gentleman living in a rural area answered. After explaining the reason for my call, I was told that the family had a 16-year-old son who used his computer to call various services.

I asked if I could speak to him. The boy soon realized that he had been caught. We talked, and I asked him why he had started

writing viruses. His response has stayed with me ever since: "I live in the countryside with my Mom and Dad. It's lonely here. The nearest village is 10 kilometers away, and there are no neighbors. To pass the time, I wrote a virus and followed the BBS discussions as it traveled around the world. I felt really good when it spread to California. I can't leave this place, but I wrote something that could."

In a way, I understood the boy. He was a typical virus author of his time: a talented, frustrated young man who wanted to leave his mark on cyberspace.

Spammer

In the days when I called people like Cinderella's author, I would have disbelieved anyone who said that we would soon face career criminals rather than hobbyists. Online crime is now fully global.

On several occasions in the 1990s, I predicted that virus writers could start to make money with their attacks. This came to pass, and money is clearly the key motive in modern malware attacks. The change began in 2003 when people sending junk email—*spammers*— joined forces with virus authors. Spamming, or unsolicited email advertising, became a real problem in the early 2000s. Primitive email filters were set up to stop spam, but they could be circumvented easily.

The first really effective spam filter was developed by Paul Graham. He described his technique in his 2002 article "A Plan for Spam." The idea was to use Bayes' theorem to separate normal email from spam by teaching an algorithm what unwanted email looks like. In other words, to use machine learning.

Graham, the developer of the first Bayes filter, is now known for the startup accelerator Y Combinator, where companies such as Dropbox, Twitch, Coinbase, DoorDash, Airbnb,

and Stripe started out. Interestingly enough, the co-founder of Y Combinator is Robert Morris, who inscribed his name in information security history in 1988 by writing the first Internet worm, the Morris worm. The worm became front-page news, and Morris received a three-year suspended sentence for writing it. The Morris worm spread across the Internet but never reached Finland because the Finnish university network was connected to the Internet's backbone network only on November 19, 1988—two weeks after the epidemic caused by the Morris worm. Morris now works as a professor, and his website refers to his worm-writing background in somewhat delicate terms: "Robert Morris is a professor of computer science at MIT. In 1988 his discovery of buffer overflow first brought the Internet to the attention of the general public. He has a PhD in computer science from Harvard."

In the beginning, the most efficient way to stop spam was to compile a list of the email servers being used and filter out all email from those addresses.

Spammers tried to work around this by continuously deploying new servers, but these were quickly detected. They had to come up with something new.

Spammers then found a way to harness the skills of amateur virus authors and use their networks of thousands of infected home computers. Young boys who wrote viruses for fun could control the machines they had infected but had no sensible use for them. Spammers understood that they could use these home computers to send emails. This would avoid the need to use their own servers, and messages from real users' computers would be much more difficult to filter out.

Infected home computers quickly became a tradable commodity in 2003. The sellers were virus authors who, for the first time, could use their skills for monetary gain. The buyers were

the world's spam kings, who used the new route to send countless millions of spam messages to sell products that made them rich.

This was a clear turning point among virus authors. Some of the old hobbyists and purists hated spam and the new era, where their hobby had been monetized to send millions of Viagra ads. A substantial share of virus authors cut their ties with the field, stopped writing viruses, and disappeared.

Those who stayed did so for money. The age of career criminals in malware had begun.

Professional Cybercrime Groups

In 2016, I developed a new term: *cybercrime unicorns*. Here, *unicorns* refers to technology startups with billion-dollar valuations. At some point, I started wondering whether we would eventually see professional cybercrime groups wealthy enough to be considered unicorns.

We are now closing in on the point where cybercrime unicorns emerge. There are two reasons for this. First, the amount of money taken in attacks by cybercrime gangs has more or less doubled each year. Second, the assets owned by the gangs have continued to increase in value. This is because cybercriminals do not store their wealth in dollars, euros, or rubles. They keep it in Bitcoin. In 2016, we knew of several gangs worth around $10 million. In 2016, one bitcoin was worth $400. In 2022, one bitcoin is worth around $40,000, and 10 million has become 1 billion.

Attacks are becoming more serious because cybercriminals can afford to invest in them. The gangs can run professional data centers and build imposing brands. They can hire lawyers and purchase valuable exploits on the market. We know of at least two cases where career criminals set up an information security company as a front and recruited penetration testing professionals to

work there. IT security professionals may have ended up working for cybercriminals as a result.

On the other hand, it is now the hunting season on unicorns. The U.S. State Department is offering a $10 million reward for information leading to the arrest of members of the REvil or Darkside cybercrime groups. Ideally, a reward this large will hamper their operations and raise suspicions between members. It is worth noting that the State Department has traditionally offered rewards this large for information leading to the arrest of leaders of the ISIS and Al-Qaeda terrorist groups. That is how serious it is now treating cybercrime.

Extremists

All extremist groups and terrorist organizations operate online. They use the Internet to communicate, recruit, and spread propaganda. We have not seen actual terrorist attacks online, although extremists may be the only group with a logical motive for destroying the infrastructure of the Internet. Hobbyists don't want to destroy the Internet; they love it. For spies and cybercriminals, a working Net is more useful than chaos.

Groups like ISIS have also succeeded in radicalizing individual traditional hackers, who have then joined the extremist group in question. One such case was an amateur hacker from England who moved to Raqqa in Syria and joined ISIS, and changed his name. I had been tracking this person for a long time and made the mistake of mentioning the case at a conference. The video of my speech went online, and before long, the hacker started sending me threats. When the hacker was killed in a drone strike, I must admit I felt relieved. Cases like these are why I do not generally respond to questions regarding my personal safety and how I manage my security.

For now, extremists are not a major player in online attacks. Let's hope it stays that way.

The Rolex

A journalist I know once called me. He was working on a TV feature on spam and asked for my help.

We met for a talk and decided to make a series of test purchases: we would buy the products advertised in the spam messages to see what happened.

We also wanted to test whether the credit card used for spam purchases would be misused and whether the email address would end up on other spammers' lists.

We went through our spam folders and picked the products: an oil that increases virility (VPRX oil), a carton of cigarettes, the latest version of Microsoft Windows, and an attractive Rolex watch.

The first surprise was that we got what we paid for: the VPRX oil came from India, the Windows CD-ROM from Russia, and the watch from Italy. The cigarettes we ordered never came, but the seller in Estonia never charged our card for them either.

The second surprise was that the credit cards did not end up in the wrong hands, despite that security on the spam websites left a lot to be desired.

The third surprise was that no additional spam was sent to the email addresses we used. You would think that the email addresses of purchasers would be a goldmine for spammers, but this was not the case.

Our Windows license arrived on a very suspicious-looking CD-ROM, but it contained what it was supposed to: the latest version of Windows. The copy protection had been cracked quite clumsily, but at least the disc did not contain malware.

Unfortunately, the Rolex from Italy was stopped by customs. A stone-faced customs inspector used a hammer to crush it as a counterfeit product before we got to see it.

The VPRX oil arrived in a small glass vial and smelled horrible. We tried to find a volunteer at the office to test its effects, but there were no takers, so we threw it in the trash.

Malware—Then, Now, and in the Near Future

Malware poses the largest single threat to information security. Within just a few decades, this previously marginal phenomenon has become a criminal industry worth billions and a tool of state-level online attackers—but the current problems may be just the beginning.

The History of Malware

Malware, which includes viruses, worms, and trojans, has had a tremendous impact on computer security since the 1980s. There are other problems, but malware is the common thread connecting almost every major information security incident. The development of malware over the last three decades divides into clear epochs.

Viruses on Floppies

Scientific publications and science-fiction authors were already mentioning computer viruses and similar problems in the 1960s. The first computer virus to actually spread was discovered in 1981: the malware, Elk Cloner, that ran on Apple II computers. Between 1981 and 1986, around 10 other viruses targeted at the Apple II were discovered.

In 1986, the first virus aimed at the Commodore 64 was found—it was known as BHP. The Commodore 64 and Apple II used a similar central processing unit (MOS Technology 6502/6510). They also both used 5.25-inch floppy disks. Early Apple and Commodore viruses were spread by users trading floppies, but never became a major problem—it took the IBM PC to do that.

The first PC virus was known as Brain.A and was discovered in 1986. Brain is often mentioned as the first virus, but it's actually just the first virus for PCs. Like Elk Cloner and BHP, Brain spread on floppies. In practice, viruses spread via floppies from one company, city, or country to another as people carried infected floppy disks with them, at a similar rate to a disease such as the flu. Such viruses required travel in order to spread: it was impossible for Brain to spread from Pakistan to Philadelphia, for example, unless someone took the trip and carried a disk with them.

Floppy disk viruses were also called *boot sector viruses*. This was because, when starting up, early PCs checked the disk drive for a floppy disk in order to load the operating system on the disk. Brain and the other early PC viruses, such as Stoned and Form, copied their code in the boot sector of every disk they encountered and relied on people transporting the disks further afield. Sooner or later, the user would start their computer with an infected disk in the drive.

Disks infected by floppy disk viruses were harmless as long as they were not in the drive when the computer was started. A peculiar mass infection took place at a bank in 1992. The finance department of the bank's main branch had a few infected MS-DOS computers, which copied the Form.A virus onto every disk that went through their drives. Over time, so many disks were infected that they accounted for the majority used at the main branch. However, they did not infect other machines, as the computers were restarted very rarely. It was company policy to leave computers on overnight, only turning off the screens. This changed dramatically after a brief power outage at the bank's main branch in the middle of a working day. Nearly every computer was in use, which is why nearly all disk drives had a disk in them. When power returned, the machines restarted, tried to load the operating system from their disks, and were infected.

Brain.A

In late 2010, I was invited to a meeting set up by F-Secure's public relations (PR) department. It was about Brain.A, the world's first PC virus.

Although Brain was a very basic virus, it spread around the world very efficiently. When I started working in information security, I studied all the known malware families and decompiled Brain.

Our PR team wanted to mark Brain's 25-year anniversary, suggesting that we use Brain in a campaign to raise awareness of the dangers of malware and its evolution. I listened to their proposal, asked to speak, and said, "Well, that's boring. What if, instead, I tried to find the authors of the Brain virus and ask them why they did it?"

I knew that we had a breadcrumb trail for this, since I recalled that the following text was hidden inside the virus's code:

```
Welcome to the Dungeon © 1986 Basit & Amjads (pvt).
BRAIN COMPUTER SERVICES
730 NIZAM BLOCK
ALLAMA IQBAL TOWN LAHORE-PAKISTAN
PHONE: 430791,443248,280530.
Beware of this VIRUS...
Contact us for vaccination...
```

Allama Iqbal Town is a district in Lahore, Pakistan, and Basit and Amjad are Pakistani first names. Sure, 25 years had passed, but how hard could it be to find these guys? After all, there are only 220 million people in Pakistan.

We decided to set the project in motion, agreeing that Olli from F-Secure's PR department and photographer Taito Kawata would join me in Lahore to capture the meeting on video. Our initial idea was simply to travel to Lahore and find the address, 730 Nizam Block, mentioned in the virus; however, it soon occurred to us that Basit and Amjad were probably no longer at the same address. So, I leveraged my contacts to ask for clues from IT security experts in Pakistan and was eventually given the email address of a supposed acquaintance of either Basit or Amjad. I sent this acquaintance an email asking them to forward my contact information to one of the two.

Hello!

I'm trying to reach Basit or Amjad.

Please pass my contact information to them.

With very best regards,

Mikko Hypponen

Chief Research Officer

F-Secure

Finland

Two days later, I received an email. The sender was Basit Alvi himself, one of the authors of Brain, and his contact information was included.

Hi, This is Basit, my contact details are as follows:
Basit Farooq Alvi | Director |
Brain Telecommunication Limited.
730-Nizam Block | Allama Iqbal Town |
Lahore 54570 | Pakistan.

I couldn't believe what I was reading. I had made contact with one of the Brain virus's authors, and the contact information he provided was the same as in the 25-year-old virus: 730 Nizam Block, Allama Iqbal Town.

I sent Basit a proposal by email.

Hello there!

My name is Mikko Hypponen and I work at F-Secure.

Brain.A virus will be 25 years old in January. So the whole PC virus will be 25 years old. We believe this is important and would like to do something around it. I have analysed the Brain virus myself long, long time ago when it was still spreading.

I would meet you guys at Brain Telecommunication and we would discuss the history of the worm. The end result would be published as an online video.

We're talking about a historic event and we want to discuss the background of it all.

Please let me know how you feel about this.

Basit and his brother Amjad accepted. We began preparing for our trip to Lahore.

Lahore is a city of more than 10 million people in northern Pakistan. Viewing the area in Google Maps reveals that Lahore is almost on the Indian border, which becomes a dotted line to the north of the city. This means that the area remains disputed territory prone to unrest. The conflicts have been beneficial to the extremist group Al-Qaeda, which was using the area as one of its strongholds.

I decided to contact the Finnish embassy in Islamabad and ask for advice. They quickly came back to me: "You must apply for visas and permits for the film crew from the Pakistani embassy in Stockholm. We recommend that you start doing so immediately, as the filming permit process has been up to four months long of late. In Lahore, you will need a local security team, which will not be too costly. You should also reserve transport in advance."

We immediately decided not to apply for an official filming permit, opting for normal business travel visas since we could not afford to wait several months.

The embassy recommended a local security company from which we could hire a security team. This company quickly replied to my email, explaining that they could provide bodyguards and civilian vehicles from the Pakistani police's security unit. I reserved a driver and two bodyguards. Our departure date was set for January 31, but the situation in Lahore escalated a few days prior to this. An employee of the CIA's Lahore branch had

shot two Pakistani men on the street under uncertain circumstances. He was surrounded by a mob and called for help. The dispatched CIA security personnel drove along the oncoming lane to reach the site quickly, fatally injuring a bystander.

The security company sent me an email recommending that we postpone our trip due to the growing tensions.

Our travel arrangements had been made, and the flights could not be cancelled. Olli, Taito, and I met to consider what to do. On the last day of January 2011, we stepped into an airport taxi at F-Secure headquarters with fresh visas in our pockets and bandages on our arms from vaccinations. We flew from Helsinki via Frankfurt to Abu Dhabi, and overnight from there to Lahore, landing at Lahore airport at 2:30 a.m. From the plane windows, you could see the warm rain outside. The terminal emerged from the mist as the plane taxied onward. Once inside the terminal, we noticed that it was crammed with people, as if it were midday.

We collected our bags and cameras, cleared passport control, and started looking for the bodyguards I had reserved. Our Finnish SIM cards could not connect to Pakistani operators' networks, and it was difficult to locate anyone in the crowd. Finally, Olli spotted a man in a leather jacket carrying a piece of paper that said "Mr. Mikko."

He introduced himself as Yasir. He led us outside to a slightly beat-up Toyota Hiace minibus. Our driver sat at the wheel, and next to him was our bodyguard, a Pakistani police officer in full uniform and armed with a pistol. Yasir explained that the three of them would escort us during our visit and made us swear to go nowhere without them.

He then asked if we could pay their fee right away. I said yes and asked if he would take cash in U.S. dollars or euros. Dollars were fine. I gave Yasir three hundred-dollar bills and asked if he could write me a receipt. His answer was brief: "no." We got into the van and were taken to the hotel. Morning had almost arrived.

Tight security around hotels is common in many Asian countries. In Malaysia and Indonesia, for example, it is not uncommon for security to check a vehicle's underside with mirrors before allowing it into a hotel courtyard. Hotel Ahari in Lahore, however, was in a class of its own. There was no way of driving into the hotel's yard from the street, even by force. Sandbags and concrete barriers had been used to build a winding route that could not be bypassed. The hotel's guards checked our passports and examined our vehicle before we were let through.

Inside, the hotel was magnificent, with massive chandeliers on the ceiling of the lobby. We checked in, and the receptionist gave me my key, while remarking that the hotel had a hidden bar in room 119. Pakistan is a Muslim country, and locals are banned from consuming alcohol. The ban does not apply to foreigners, but the idea is to keep alcohol out of sight. I had a look around room 119 later, which indeed had a hidden bar—a speakeasy, if you will. From the corridor, the room appeared similar to the others, but its door was ajar. Behind the door was a small bar, with a smiling bartender at the counter. Short-haired, huge men sat at the tables, conversing in English and swallowing Budweiser by the bottle. Their gray suits bulged suspiciously, as if they all had holsters and handguns. Later during the trip, we heard that the CIA was using the hotel to accommodate its consultants. That explains the Budweiser men and tight security.

The next day, our van took us to Nizam Block at the other end of the city. As I stepped out, the feeling was unreal: the address I had found on the boot sector of an infected disk was an actual location. House number 730 was a gray, two-story building, and the door said Brain Telecommunication Ltd.

We were welcomed by our contact, Kashif Talib. He took us to a conference room where three brothers sat at a table: Basit Farooq Alvi, Amjad Farooq Alvi, and Shahid Farooq Alvi.

Basit and Amjad were the same Basit and Amjad whose names were recorded in the virus.

We surveyed each other tensely, but I broke the ice by taking a floppy disk from my backpack. I showed them my original disk for Brain, explaining that this was where I found and decompiled their original code. Basit and Amjad began to smile and, soon enough, to reminisce about information technology in the early 1990s.

I asked them how the virus came about. Basit told us that he and Amjad began using computers as long ago as the early 1980s, when their work involved IBM mainframes. When IBM introduced its first microcomputer, the IBM PC, they were given access to it—and it shocked them. The security on mainframes had been from a different world. Simply put, microcomputers had no built-in security: no user accounts or privilege levels, and each program had full access to the device. Software was left free to do anything—such as write itself onto user floppies without permission and spread further.

Basit and Amjad had the idea of using a practical demo to prove how insecure the IBM PC was. Brain was this demo, which is why the authors' names were left in the code. Originally, Brain spread only on computers within a school, but it eventually escaped. Amjad told me that they occasionally received phone calls from infected users who had found their phone number in the virus's code. They also received letters from all over the world: the United States, New Zealand, Germany….

Writing a virus was not illegal in 1986. Nevertheless, Basit and Amjad eventually became frightened and stopped discussing their virus. Meanwhile, Brain continued to spread around the world.

When I asked whether they understood the dramatic implications of the revolution they had begun, they both shook their heads. On the other hand, they felt that if they hadn't

written the world's first PC virus, before long someone else would have done so.

The brothers were clearly troubled by modern viruses, which they had encountered and cursed many times—the same problem that they had given birth to. Unlike current viruses, Brain was not an attempt to make money or spy on the user; it simply spread.

The three Alvi brothers were still in the computer business. Called Brain, their company was Lahore's largest Internet operator and was building a fiber-optic network across the city. We spent a couple of days with the Alvis.

Lahore was a city of conflicts, with crowded streets and hordes of homeless people, but to me, the food was exotic and delicious. People always seemed to be either happy or angry. I can recall no other place where people were unfriendly to me simply because I looked Western. For example, on one occasion our van stopped at a traffic light and a local man pulled up alongside us on his motorbike. He maintained a neutral expression until he saw me and was overcome with apparent fury. Our bodyguard explained that this was because he thought I was American. Many of the locals hated the United States, whose soldiers were hunting Osama Bin Laden across Pakistan.

When about to leave, I gave my original Brain disk to Basit and Amjad. In a way, I had brought it home.

Three months after returning to Finland, I heard news breaking on the radio. Osama Bin Laden, the leader of the terrorist group Al-Qaeda, had been tracked to the city of Abbottabad and killed by American soldiers. I used Google Maps to check how far we had been from Bin Laden. The distance was about 300 kilometers (190 miles). Even that felt slightly too close.

At the turn of the quarter, as I filled out an expenses form for F-Secure's finance department, I listed the following,

"Bodyguard services in Pakistan: USD 300. No receipt."

File Viruses

The evolutionary stage following floppy viruses was file viruses, which infect program files. In the early days, these were mainly COM and EXE files for MS-DOS systems. While file viruses could spread on floppies, they were much more efficient at spreading over file-sharing services.

The first widespread file viruses, such as Cascade, Yankee Doodle, and Jerusalem, spread at the turn of the 1990s. Since Internet connections were rare at the time, files were commonly shared in modem-based BBSs. Internal file servers in companies were another common route.

The first computer virus I decompiled was a file virus, which was also the first virus I named. I chose the name Omega, as the virus would display the omega symbol every Friday the 13th, while destroying the data on the computer. Years later, we began the tradition of presenting employees with an Omega watch to mark ten years of service (I know, I should have named the virus Ferrari).

The early file viruses worked on MS-DOS, but in the summer of 1992 we found the first Windows virus in Sweden. We called it WinVir. Over the next few years, MS-DOS viruses disappeared, little by little, to be replaced by Windows viruses.

When the Internet began to grow, the first popular services were email and FTP file sharing. File viruses spread when users exchanged files without knowing that they had been infected.

Macro Viruses

What do people use their computers for? Most create or edit documents in Word or Excel (or maybe Google Docs these days). Users share their DOC and XLS files by email or via file-sharing services. This provides a promising environment for spreading malware via document files.

Concept, the first macro virus running on Microsoft Word, was discovered in 1995. Opening a file infected by the Concept macro would infect the Word environment. After this, every Word document that passed through the computer became infected, with the macro carried by the malware replacing the File/Save As function in Word. Files were infected upon saving, and the virus spread rapidly between machines as users shared files they had created.

Concept did nothing except spread. The first Excel macro virus, Laroux, also spread only as XLS files were shared. Destructive macro viruses were discovered later. Corrupting or overwriting documents may sound bad enough, but even worse damage was done by macro viruses making barely observable changes.

Over time, some Excel viruses would make small random changes to the numbers on a worksheet. If the changes were subtle enough, they went unnoticed and users continued to use and back up the incorrect data. The changes became noticeable once they had accumulated sufficiently—by which point the user no longer had a proper version of the file even in the backups.

Discovered in 1996, a Word macro virus called Nuclear added two extra rows to the end of the user's document:

```
And finally I would like to say:
STOP ALL FRENCH NUCLEAR TESTING IN THE PACIFIC!
```

This political commentary could be seen only once the document was printed. Everything looked good to users viewing their job application or budget proposal on-screen, with many failing to notice the political slogan at the end of the printed version.

Email Worms

In December 1998, we encountered the malware Happy99. When it ran, it would display a fireworks animation on screen, wish you a Happy New Year 1999, and email a copy of itself to everyone in your address book.

Email worms like Happy99 quickly became commonplace at the turn of the millennium. They were so successful because people tend to trust those they know. When an email worm spreads, it usually sends itself while masquerading as the user of the infected computer. Messages go out to people in the user's address book, most of whom know the sender well. When an email arrives with an unexpected attachment from an unknown sender, most people do not open it—but if it comes from a close friend, colleague, or old classmate, they are more likely to click on the attachment.

The situation was exacerbated by the fact that email systems were very liberal about attachments back then. Email attachments now tend to be images, text files, or Office documents, but in the early days of email it was common to send EXE files: people shared Windows software with each other. When the recipient ran the software on their computer, the program was given free access to the system. Nowadays, an EXE attachment would never arrive, as the sender's, recipient's, or operator's filters would remove it due to the danger posed.

Email systems crashed on May 4, 2000, in one of the largest malware epidemics in history: ILOVEYOU aka Love Letter. Love Letter spread as a VBS file rather than an EXE attachment. VBScript was a simple programming language. When a victim ran the malware, the computer sent a message to every email address on their address book: "Kindly Check the Attached

LOVELETTER Coming from Me." When the confused recipients opened the attachment to read the love letter, the process would repeat on their computers. This malware spread like wildfire to dozens of millions of computers all over the world.

Email worms were a major problem for many years, with most concern caused by macro viruses spread by email. Attackers combined two technologies: macro viruses and email worms. The most famous of these was the Word macro virus known as Melissa. When a victim opened an infected document file, a macro within would send the same document to every email address in their address book—in the victim's name.

Later, we saw macro viruses that would spread themselves to random email addresses in an address book, through a DOC file attachment taken from the user's computer and bearing a harmful macro payload. This method of spreading was far worse than it sounds, as a company's highly confidential plans for the next year could be sent to a newspaper journalist, or an unfinished book might end up on a public mailing list. There were many such cases in the early 2000s.

Internet Worms

Email worms spread quickly. An efficient worm could traverse the world in one day but required human assistance to do so—someone had to open the infected email.

The next stage of evolution took spread rates from hours to minutes. Internet worms no longer needed human assistance. The first Internet worm occurred in 1988, when the Morris worm spread in Unix environments, but the problem proliferated in the 2000s when network connections reached an entirely different level.

Code Red, discovered in 2001, infected Windows computers, but only their server versions. It spread from one computer to

another, searching for web servers based on Windows IIS. IIS, Internet Information Services, was server software developed by Microsoft. If IIS was not up-to-date, Code Red infected the computer and replaced the web server's front page with this greeting:

```
HELLO! Welcome to http://www.worm.com!
Hacked By Chinese!
```

The infected server then began seeking other computers to infect. The rate of infection increased as the number of servers scanning the network continuously grew. More than 300,000 servers were eventually infected, a substantial share of all web servers in 2001.

Shortly after Code Red, an Internet worm called Nimda was identified, which infected both Windows workstations and servers. Workstations were infected when they connected to an infected web server: infected workstations sought and infected vulnerable servers online, completing the circle by creating more infected web servers.

In 2003, a spreading record for Internet worms was set that has yet to be broken. On the Internet, computers use either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) for communication. TCP traffic involves negotiation—the infection rate of worms, like Code Red and Slapper, that use TCP as a spreading mechanism is limited by the fact that targeted machines must reply to the packets sent to them. UDP traffic, on the other hand, is not a negotiation; it is akin to yelling at the network, with no responses needed or expected.

The first worm to use UDP packets to spread was known as Slammer. It infected Microsoft's SQL database servers and was minimalist in size, at only 376 bytes. This allowed the entire worm to fit into one packet, as the size of UDP packets is commonly limited to 512 bytes.

Slammer spread at an astonishing rate. Infected servers would fire out UDP packets as quickly as they could. When a packet like this hit a vulnerable device, it was immediately infected, adding to the barrage of deadly packets.

Slammer was set in motion at 5:31 UTC on Saturday, January 24, 2003. By 5:45, it had already infected all machines that it could infect, in less than 15 minutes.

All this from a tiny worm that you could fit into two tweets:

```
04 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
01 DC C9 B0 42 EB 0E 01 01 01 01 01 01 01 70 AE
42 01 70 AE 42 90 90 90 90 90 90 90 90 68 DC C9
B0 42 B8 01 01 01 01 31 C9 B1 18 50 E2 FD 35 01
01 01 05 50 89 E5 51 68 2E 64 6C 6C 68 65 6C 33
32 68 6B 65 72 6E 51 68 6F 75 6E 74 68 69 63 6B
43 68 47 65 74 54 66 B9 6C 6C 51 68 33 32 2E 64
68 77 73 32 5F 66 B9 65 74 51 68 73 6F 63 6B 66
B9 74 6F 51 68 73 65 6E 64 BE 18 10 AE 42 8D 45
D4 50 FF 16 50 8D 45 E0 50 8D 45 F0 50 FF 16 50
BE 10 10 AE 42 8B 1E 8B 03 3D 55 8B EC 51 74 05
BE 1C 10 AE 42 FF 16 FF D0 31 C9 51 51 50 81 F1
03 01 04 9B 81 F1 01 01 01 01 51 8D 45 CC 50 8B
45 C0 50 FF 16 6A 11 6A 02 6A 02 FF D0 50 8D 45
C4 50 8B 45 C0 50 FF 16 89 C6 09 DB 81 F3 3C 61
D9 FF 8B 45 B4 8D 0C 40 8D 14 88 C1 E2 04 01 C2
C1 E2 08 29 C2 8D 04 90 01 D8 89 45 B4 6A 10 8D
45 B0 50 31 C9 51 66 81 F1 78 01 51 8D 45 03 50
8B 45 AC 50 FF D6 EB CA
```

Slammer caused major problems at several international banks, forced around 13,000 ATMs to close in the United States, and infected a local area network in at least one nuclear power plant.

Slammer made front-page news, similarly to Blaster, which was discovered a few months later. Instead of SQL servers, Blaster infected normal Windows workstations.

Firewalls were not yet part of standard Windows installation in 2003. This meant you could contact the open ports of basically any Windows machine, even ones located halfway across the world. Blaster contacted TCP port 135, taking advantage of a vulnerability in the RPC service listening on the port. This would infect Windows—and crash it. The operating system displayed the mysterious error "Windows must now restart because the RPC service terminated," after which the user had 60 seconds to save any open documents before Windows restarted.

Blaster leveraged a vulnerability that Microsoft had patched some three months earlier. In 2003, however, Windows did not have an automatic update service: users had to download updates manually from microsoft.com and install them. This was not easy, because a computer could be re-infected during the download and restart before the download was complete.

Blaster and its successor, Sasser, led to a clear change of heart at Microsoft. It understood that the future of Windows hung in the balance. The year 2003 was a turning point as Microsoft, previously considered a prime example of poor security, began setting an example for the software industry.

Internet worm epidemics were tamed as firewalls and other protection systems proliferated. The last major Internet worm was Conficker, discovered in 2008, which infected more than 10 million computers and remained on the most common malware list for more than a decade.

The Virus Wars

At F-Secure, we remember 2003 and 2004 as the years of the great virus wars. F-Secure had only one research laboratory, in Helsinki. When a malware epidemic started, we investigated it from Helsinki, even in the middle of the night. Our phones rang,

and our team got to work. We obtained a sample of the new virus, decompiled its code, and determined how it was spread. We then developed a detection algorithm, named the malware, built an update package, and sent it to our customers over the Internet. These sessions were intense. Our team was highly experienced and professional. Everyone knew what they needed to do—it was like watching top surgeons in an emergency room. During a major malware outbreak, our ears buzzed as our bodies pumped adrenalin—as if we were in physical danger. The outer world melted away as we became hyper-focused on the case. If a phone rang in the middle of a big case, effort was required just to comprehend a caller who wanted to talk about something else. Once done, we truly felt that we had completed a labor of Hercules.

We developed our process and created a three-level warning system. When the outbreak reached the highest level, several things happened at once: all available analysts were called in, all staff were notified, the system automatically cancelled all internal meetings requiring representatives from the laboratory, customers received a separate warning via text message, supplementary staff were automatically requested for the telephone exchange, and laboratory employees had their lunch breaks cancelled and pizzas were ordered. For the latter purpose, our intranet site had a separate "outbreak pizza" application, which employees used to enter their pizza preferences in advance.

However, new epidemics kept occurring, and crises became a regular feature, with wakeups at 3 or 5 a.m. several times a week. Before long, continuous crises became exhausting rather than exciting. To me, the summer of 2003 is a blur. In 2004, we started building a system that divided our research between three offices located in Finland, California, and Malaysia. The time difference between these locations is about eight hours, and 8 times 3 makes 24. This allowed us to distribute continuous

on-call duties between three continents, with everyone working humane hours.

Web Attacks

Exploit kits ruled the malware scene in the early 2010s. Instead of viruses or worms, they spread trojans, but infected machines did not spread infections further.

Exploit kits were usually installed by breaking into popular websites and surreptitiously modifying them, allowing the attack code to scan the visitor's hardware. The machine was scanned for outdated components against an extensive checklist: Which operating system? Which browser? Which browser extensions? Is Java installed? How about Flash? Is everything updated to the latest version?

If an outdated component was found, the automation would send out a targeted exploit code. This was invisible to the user, but resulted in the installation of malware.

Mobile Phone Viruses

In the early 2000s, Symbian was the most important mobile operating system. There were more than a dozen manufacturers of Symbian phones: Samsung, Motorola, LG, Sony Ericsson, Fujitsu, and Sharp to name a few. Nokia was by far the largest, however. More than 250 million Symbian phones were sold between 2001 and 2009, with the vast majority being built by Nokia.

Nokia is no longer a household name, having gone from phone manufacturer to infrastructure manufacturer. It is very likely that your Android or iPhone connects to the network via an access point or base station built by Nokia. Nokia remains one of Europe's largest technology companies and has more

than 90,000 employees. My favorite Nokia anecdote relates to the recognition it has received: Nokia has nine Nobel prizes, five Turing prizes, two Grammys, seven Emmys, and one Academy Award (Oscar).

Nokia phones played a key role when the first mobile phone virus broke out in the summer of 2004. The virus was known as Cabir and spread between Symbian phones via Bluetooth. All Cabir did was spread from phone to phone and display its name on the phone screen. However, other Bluetooth viruses quickly appeared—more than 20 had been identified by the end of 2004.

The spread of Symbian viruses grew consistently between 2004 and 2007. The worst offender was a virus known as Commwarrior, spread by text messages in addition to Bluetooth.

Bluetooth malware was not automatically installed on the phone; instead, the user had to accept a Bluetooth connection from another device. For this reason, the entire problem was downplayed and the blame placed on infected users. But users were not the problem; Bluetooth viruses were able to spread because of poor user interface design.

When an infected Symbian phone was within Bluetooth range of another phone, both Cabir and Commwarrior tried to send their installation package to the target phone as a Bluetooth message. If the user declined the message, the virus would simply keep trying.

For the victim, the situation was misleading. For example, an onscreen message would keep appearing asking, "Do you want to receive a Bluetooth message from Nokia 3650?" The options were "Yes" or "No." When the question was on-screen, the phone could not be used for anything else. If this cryptic question popped up when the user wanted to make a phone call, they had to respond to it before calling. It could not be ignored.

Most users probably didn't know what to make of the question and answered "No." However, the question would immediately pop up again, as if "No" was the wrong answer. Many users ultimately decided to try answering "Yes" to restore normal use of their phones—and infected their devices. The newly infected phone then tried to send the malware to other phones wherever the user went.

What was the user to do? Turn off Bluetooth? That would have helped, but the user could not even use the phone settings when under attack. The correct answer is that the user should have walked away. Once they were outside Bluetooth range, the message would have disappeared. However, this was far from obvious to users struggling with the problem. Many users infected their phones after becoming frustrated with the question-based logic of the Bluetooth interface.

For a while, Bluetooth viruses were a real problem. I received malware broadcasts on my Symbian phones when stuck in Helsinki's morning traffic, as someone in a nearby vehicle had an infected phone. The same happened in a hotel lobby in London and at Arlanda airport in Stockholm. At the 2005 World Championships in Athletics, the stadium screens showed virus warnings to spectators.

Some efforts were also made to benefit financially from Bluetooth viruses. Infected mobile phones were used to send expensive text messages to international service numbers, or even to call hotlines in the middle of the night. Almost 400 identified, malicious Symbian applications were at large by 2007.

How did we do away with Bluetooth viruses? In late 2007, Nokia released the Symbian Series 60 Third Edition operating system, in which the phone remembered and did not reshow the negative answer to the Bluetooth question, even if the infected phone tried to reopen the connection. This small change

eliminated the problem, and Bluetooth has not been a notable transmission route for malware since then.

Worms on Social Media

As other available holes have been plugged, attackers have switched to a technique that always works: deceiving users. Malware now tends to arrive via social media messages or emails. Convincing a user to download and install a utility or update on their computer clears the way for an attacker to get in.

Facebook, Twitter, WhatsApp, Telegram, and Snapchat continuously spread malware. Most of these epidemics are quite invisible, spreading through private or targeted messages.

We are seeing fewer and fewer viruses that spread in the traditional manner. Taking over a computer now means stealing a user's credentials rather than sending traditional malware. With increasing frequency, company networks are being penetrated via vulnerabilities in online services. Even then, infection is most commonly caused by a trojan, not a virus or worm that tries to spread itself.

In other words, we won the war, and viruses are almost extinct. Online attacks, on the other hand, are still going strong.

Smartphones and Malware

Which is more secure, a smartphone or a real computer? The answer is a smartphone, but the underlying reasons for this may not be obvious.

The popularization of modern mobile operating systems has been the largest information security improvement in everyday life of the past 15 years. The iPhone, launched in 2007, has never been subjected to a widespread malware epidemic. This is an astonishing achievement, for which Apple deserves our congratulations. The rare cases that we have seen on iPhone have typically been targeted attacks done with tools like Pegasus. These tools are very expensive and typically only available for law enforcement and intelligence agencies. Google's Android has had a few more issues, but it is also much more secure than traditional computers.

Mobile operating systems (iOS, iPadOS, and Android) are therefore more secure than desktop operating systems (Windows and macOS). This is because they are more limited for users, even if the limits are less than obvious.

Mobile devices and computers may seem similar. For example, fitting a keyboard to an iPad Pro makes it similar to Apple's Macbook laptop. They can both be used for the same things: browsing the web, using Photoshop, playing games, paying bills online, processing documents, and so forth.

The only major difference concerns users who are also programmers. The Macbook is a computer. Any coder can write and run software for their Macbook and give the software to a friend, who can also run it on their computer. This has not been possible on an iPad. In the iPad world, owners have not been allowed to program their own devices, a limitation that may sound harsh, but is familiar from another environment—game consoles. Internally, the iPad is a computer, just like the PlayStation

and Xbox, but you cannot program a PlayStation after buying it: you can only purchase ready-made game software to run on your device.

To run your own software on iPad, you must first send the program for approval—to Apple in California. So, you can run your software only if it is approved by Apple. Similarly, Sony approves all software run on PlayStations, and Microsoft approves all Xbox software.

Such architecture is very limited but also very secure, which is the main reason for the absence of malware epidemics on iPad, iPhone, PlayStation, or Xbox. Android has a similar but slightly more open system. The checks run by the Google Play app store are somewhat less strict, which explains why we do see some malware on Android. Of course, Android is more secure than Windows or macOS, as the latter are fully open environments programmable by anyone.

I know of companies that have recovered from company-wide outbreaks by replacing their Windows laptops with iPad Pros or with Google Chromebooks. This is improving security by removing functionality.

In other words, if you are doing something particularly important, do it on a mobile device rather than a desktop computer. For example, online banking is more secure on a smartphone than on a "real" computer. Smartphones do have their risks, however. For example, few of us leave a computer on the back seat of a taxi after a night on the town, but quite a few smartphones end up in such places.

Law Enforcement Malware

The concept of law enforcers using viruses and other malware to infect citizens' computers may seem far-fetched. However, this is a commonly utilized technique everywhere in the world.

In the end, it comes down to which rights we citizens want to grant the authorities. If we feel that the trade-off between safety and privacy is balanced, we will grant special rights to the authorities. On the other hand, if the trade-off is one-sided, we will not.

When landline phones became common, law enforcers wanted the right to tap them. After the emergence of mobile phones, the police were permitted to listen in on the mobile network. Then came authorizations to track text messages and emails. However, when powerful encryption systems became the norm, tracking traffic was no longer enough, as nearly all online traffic intercepted by the police was encrypted—leading to the right to install malware on suspects' devices. This bypasses encryption, as messages are read before they are encrypted or after they are decrypted.

But how is malware installed on a suspect's device, once law enforcers are authorized to use such software? Many techniques are available, but most are quite different from those used by cybercriminals. The police may apply for a warrant to break into the suspect's home and physically infect devices, or may cooperate with a local Internet operator to modify software that the suspect downloads from the Internet.

It seems slightly strange that the same police officers we, at F-Secure, help to catch cybercriminals also use malware in their work. I have discussed this with the officers, saying that, while I understand their need to use malware, we will nevertheless try to prevent it. I have also told them not to expect our help if they do use malware. We cannot ignore viruses written by the authorities, regardless of how good the intentions are. If law enforcement wants to use malware, that is their business.

Case R2D2

The first time we encountered law enforcement malware was in October 2011—malware known as R2D2 or 0zapft, which had been created by the federal government of Germany. When we published the detection update for this malware, I wrote about it in our blog, cagily mentioning that we had been sent a sample "from the field." I can now disclose the truth. I was contacted by the Chaos Computer Club (CCC), a German association promoting freedom of speech regarding technology. Their experts were helping a person who was facing charges for customs violations. The accused suspected that their laptop had been infected by German border guards at Munich airport, upon the suspect's arrival in Germany.

CCC's technicians found complex malware running in the background of the computer and monitored the operation of four programs: Skype, Firefox, MSN Messenger, and ICQ chat. The club was unsure about what would happen when the case became public, particularly whether antivirus software would identify the malware, which was not criminal software. In fact, it was the polar opposite.

CCC's representatives called me because, in a public speech, I had made it clear that we must protect our users, regardless of where malware originates. Our definition of malware is technical, not political or societal. Malware is software that the user does not want on their computer, and R2D2 met this definition.

The CCC wanted to ensure that, when the R2D2 case became public, a well-known and reliable antivirus solution would not hesitate to add recognition of it, making it easier for other companies in the field to follow suit. We did just that. When the German press heard about the case, it became international news within an hour. We had built our detection update in good time

and published it simultaneously with my blog post, where I presented the rationale for stopping malware even when written by law enforcers. Three hours later, Avast antivirus software started to remove the malware. One hour after that, McAfee did the same, followed by Kaspersky. By the evening of the same day, practically everyone had done the same. CCC's tactics had worked.

Cracking Passwords

When a suspect is arrested and their devices are locked with an unknown password, there is only one option left: cracking the passwords. Although data traffic protected with modern encryption is practically impenetrable, an attempt can be made to decrypt files or file systems by trying all possible passwords. This can often be expedited by distributing the task between dozens or even hundreds of computers.

Authorities have pretty impressive decryption systems for this purpose. An office building in The Hague, for example, has decryption hardware the size of a supercomputer, which needs its own power station. Hardware like this allows millions of password options to be tested per second. Nevertheless, a single encrypted file can take months to open.

Automated decryption systems use a clever tactic for speeding up this operation. If a password-protected file is found on a suspect's hard drive, all files on the drive are indexed, and all individual words are collected from each file for testing as passwords. If none work, all discovered words are tested in reverse, and if this does not work, the drive is scanned for any unused areas and deleted files, and words inside them are tried. On surprisingly many occasions, this will decrypt the files.

When a Hacker Spilled Her Coffee

Hardened cybercriminals know when they are being hunted. Many of them have prepared for an eventual arrest, building systems to destroy the evidence.

A Russian botnet criminal ran a computing center inside his two-bedroom apartment in St. Petersburg. He had his door changed to a sturdy metal door attached to a heavy metal frame. When the local police arrived to arrest him, Igor had plenty of time to start deleting files from all his servers. Being unable to get through the door, the police eventually made a hole in the wall next to it. Igor was found in his kitchen. On the hot stove, he had a kettle, and in the kettle he had his memory cards and SIM cards from his mobile phones. Their contents could never be restored.

The best tactic is to distract a criminal, preventing them from destroying or locking their devices upon arrest. In an arrest coordinated by EUROPOL, the suspect was sitting in a café with a laptop when a female undercover officer sat at the same table, spilling her coffee a few moments later. The suspect stood up, at which point a male officer waiting behind stepped forward and whisked away the unlocked computer for forensic analysis.

Ransomware Trojans

Cybercriminals involved in data theft have been repeating the same trick for years: steal valuable data and sell it to the highest bidder. Ransomware trojans take this one step further: why sell valuable data to others when you can sell it back to the original owner?

Ransomware trojans are a problem for companies and home users alike. While companies tend to have at least some form of backup, the situation is much worse for home users. In practice, ransomware Trojans make money by locking us out of our memories, which we increasingly preserve on digital devices. Our archives, correspondence, and, say, photographs of our children are on our computers. We could permanently lose some of our cherished memories if they are encrypted by malware.

People, please back up your files! Then, back up your backups! Make sure that your backups work and that you have at least one copy outside your home, in case your house burns down.

The History of Ransomware Trojans

The first instance of a ransomware trojan goes surprisingly far back, all the way to 1989. The following quote is from a book by Petteri Järvinen, published in 1990:

> *A widely publicized Trojan attack occurred in December 1989. The events were already set in motion in April 1989, when three men founded a company known as PC Cyborg Corporation in Panama.*
>
> *On Monday, December 11, 1989, the company mailed out at least 7,000 (some sources put the figure as high as 23,000) floppy disks from London to addresses acquired from two sources: Some had been purchased from the subscriber register for PC Business World magazine, others had been compiled from the list of attendees at*

a WHO AIDS conference held in Stockholm in October 1988. The disks were mainly mailed to Europe, but Australia, Africa, and Asia received their share. They were sent in white envelopes containing a blue sheet of paper in addition to the floppy disk. The paper gave the name of the software, "AIDS Information Introductory Diskette Version 2.0," and instructions telling the user to insert the disk in drive A: and type A:INSTALL. The reverse side had a complicated software license beginning, in the regular manner, with a disclaimer that the author is not responsible for any damage caused by the product; however, it also indirectly referred to damage caused by the software if the license fee was not paid. Experienced computer users usually only glance at texts like these. In this case, such cursory treatment was a mistake.

Once the computer had been used for a certain time after installation of the AIDS Info, the trojan encrypted the computer's file system and displayed a notice that it had prevented use of the computer because the user had not paid the $189 software license fee. Payment was requested by wire transfer to Panama; once paid, the computer would be unlocked.

The man behind the AIDS Info trojan was American Joseph Popp, who confessed to the crime but pleaded insanity.

Following this event, nearly 15 years passed before malware called GPcode kicked off the next era of ransomware trojans in 2005. GPcode locked user data, selling the user a decryption key that allowed them to access the data again. A few imitators followed, but the problem remained fairly minor at this point.

The next stage of ransomware trojans began in 2009, when malware called FileFixer was discovered. This infected the user's machine, encrypted their documents and pictures, and then showed an error notification that seemed to come from the Windows operating system. The notification stated that the file system was corrupted and the user's files were no longer accessible.

It then recommended that the user download a utility called Data Doctor. Astonishingly, Data Doctor seemed able to fix the user's corrupted files—but all it really did was decrypt them. The trial version of Data Doctor worked on only a few files: all files could only be "fixed" by purchasing a software license for $89. Thousands of victims fell for this scam without ever realizing that there was never a problem with their file system and that the fixer software was part of the scam.

A new type of ransom tactic emerged in the following year: the ICPP Trojan locked the user's computer, notifying them that illegally downloaded music or movies had been found on the system and that the "ICPP Foundation" had locked the computer until the user paid for the discovered content. This scam was fairly well executed, as it genuinely searched for media files on the user's computer and drew up an indictment. The user was offered the opportunity to transfer the investigation to a district attorney or pay a fee, which was easy to do by credit card.

The next stage of evolution, police trojans, were not a far cry from the ICPP Trojan. Reveton, which locked the user out of their PC and displayed a police notification on the screen, was the most common of these. Its message was clear: the computer had been used for illegal activity, and the user had to pay a police fine to continue using it. Reveton had been localized for different markets. It identified the location of the infected machine and displayed the message in the local language, posing as local law enforcers. A computer infected by the same trojan would appear in the name of the Australian Federal Police in Australia, the Cuerpo Nacional de Policía in Italy, and the FBI in the United States. Reveton was so successful that competition emerged from several other police Trojans. Some of them upped the ante by copying actual, illegal content on the user's computer.

GPcode, Filefixer, ICPP, Reveton, and other similar Trojans collected their money via credit cards or gift cards. Users were

asked to use their credit cards for payment or to purchase virtual credit cards or gift cards, such as Paysafecards or MoneyPaks, from newsagents. They then had to send the card numbers to the scammers. These techniques for transferring money were difficult and risky for scammers, but everything changed in September 2013 when we found a new ransomware Trojan known as Cryptolocker.

Cryptolocker

The Cryptolocker malware is heavily linked to the Russian group Zeus, led by Evgeniy Bogachev from Moscow. Zeus started out by spreading bank trojans, but expanded to ransomware trojans with the introduction of Cryptolocker. After locking the user's computer and encrypting files, Cryptlocker left a large red notification on-screen telling the user to pay either $300 US in MoneyPak cash cards or 2 bitcoin. Because, in September 2013, 1 bitcoin was worth around $125, it made sense for the user to pay the ransom in bitcoin. Having tracked the Zeus group's wallets, we know that they collected more than 40,000 bitcoin from their victims. Virtual currencies had obvious benefits for ransomware trojan gangs. Once they had the money, its origins were much easier to hide than those of traditional currencies.

Cryptolocker had an endless number of copycats such as Torrentlocker, Cryptowall, CTB-Locker, Teslacrypt, Jigsaw, and many others. Before long, ransomware trojans became the most common way of making money from malware. As the phenomenon grew, attacks increasingly shifted from home users to business computers. Home computers were commonly penetrated by means of email attachments or *exploit kits* installed on web pages. Business computers, however, were more commonly compromised through vulnerabilities. RDP and VPN servers missing security updates were a particularly common route.

Honest Criminals

The Popcorn ransomware trojan innovated with a new payment method. The victim could get their encrypted data back by paying one bitcoin in ransom—or for free, if they infected two friends with the same trojan. If two of your victims paid the ransom, your data would be restored for free. Ingenious.

When ransomware trojans promise to return the encrypted files once the ransom has been paid, there is clearly no guarantee that the criminals will keep their promise. In practice, however, this has happened in the case of nearly all ransomware trojans: the encryption really was removed after payment.

Criminal gangs understood that their operations needed a good reputation. Nobody would pay the ransoms if ransomware became known for not restoring data after ransoms had been paid, which meant that the criminals needed to be honest. Online forums were filled with stories from earlier victims, explaining how the files were restored, no problem, once the payment was made: "Five stars, would recommend." This brand-building explains why ransomware groups have names and give interviews.

Many groups even run a support chat service. This ensures that victims who pay the ransom really get what they paid for, and support would help them if a file was not correctly restored.

The reputation of ransomware trojan groups took a major hit in 2017, when two massive ransomware trojans started spreading internationally, neither of which returned the files even when the ransom was paid. They were known as Notpetya and Wannacry.

Notpetya

The worldwide attacks in May and June 2017 were exceptional in many ways. The key difference between them and normal ransomware trojans was that they were backed by governments

rather than criminal gangs: Wannacry was done by North Korea, Notpetya by Russia.

Notpetya was built just to wreak havoc. While it looked like a ransomware trojan, this was a cover story: in reality, Notpetya was a cyberweapon. Any machines infected by it displayed a ransom note similar to real ransomware Trojans, but paying the ransom did not restore the data.

Notpetya targeted a single country, Ukraine, and could be traced back to unit 74455 of the GRU, Russian military intelligence. The reason for the attack was also clear, as Russia and Ukraine had been at war for three years at this point. During the conflict, Russia had employed its cyberattack capabilities several times, and as the conflict continues still today, we keep seeing more Russian attacks.

Notpetya's spread route was exceptional, as it used accounting software updates. A few weeks earlier, hackers from Russian intelligence had penetrated the network of the software company, Linkos, operating from Kyiv. Linkos was behind a popular piece of software called MeDoc. MeDoc is used for bookkeeping and compiling tax returns across Ukraine. On June 27, 2017, the attackers used Linkos' official update servers to deploy a new software update for MeDoc. That update was Notpetya.

Computers in companies using MeDoc automatically downloaded and installed the new update. The effect was immediate, and a significant proportion of Ukrainian companies sustained damage. Point-of-sale systems at retail chains stopped working, mass transport servers went down, and problems were reported in bank networks. And this was just the beginning.

More than 40 million people live in Ukraine. It is an important market and source of raw materials for many Western companies. Such companies tended to have branch offices in Ukraine, many of them in the capital, Kiev. Many branches used MeDoc and acted as an entry route to the West for the Russian

government's cyberweapon. Once Notpetya was inside a company's network, it spread like wildfire before destroying everything it could reach.

The destruction in Ukraine was already widespread, but as Notpetya spread internationally, the damage became catastrophic. Maersk, the world's largest container shipping company, reported damage worth $300 million, while FedEx suffered $400 million in damage, and the pharmaceutical company, Merck, reported losses exceeding $870 million.

In the end, Notpetya became the most expensive cyberattack in history, being larger than any malware epidemic, data leak, or hack.

Case Maersk

The Danish company Maersk is probably the best-known Notpetya victim outside Ukraine. It is a logistics company with offices around the world and more than 80,000 employees. The fact that Maersk operates more than 70 port terminals gives an idea of the scale of its operations. Browsing the list of ports will also help you grasp its global coverage: Gothenburg, Rotterdam, Los Angeles, Miami, Buenos Aires, Bahrain, Guangzhou, Karachi, Shanghai, Yokohama, Monrovia….

On June 27, 2017, most of these container ports ground to a halt. Instead of allowing a continuous flow of containers transported by trucks, the gates to Maersk's ports stopped opening at noon. Incoming trucks were left waiting behind the gates. Lines started to build up. With the exit gates also shut, outgoing trucks were trapped inside the ports.

Finally, the truck drivers dug out their mobile phones and called Maersk's freight system support. The calls were not picked up—in fact, they didn't even get through. Someone tried to call Maersk's telephone exchange, which didn't work either. Frustrated, the drivers opened the web address maersk.com on

their phones. The site didn't open; instead, it displayed an error regarding missing index files.

All these disturbances originated from the same malware attack. Notpetya had spread across Maersk's global backbone network, gained access to administrator credentials, and spread across the organization, encrypting data and preventing computers from starting.

The destruction was not limited to laptop computers, desktops, and servers. Notpetya also deactivated the computer-controlled gates at ports. It silenced phones, since Maersk was using modern voiceover IP telephony systems, and deleted all data from Maersk's web servers, leaving customers and contractors in total darkness.

My friend, Andy, was the director of cybersecurity at Maersk when the attack started. He later told me how he heard about the attack. The management team of Maersk's global IT unit, in Maidenhead, near London, was attending a professional development day in the conference rooms of a nearby hotel. Andy's phone rang at around 11 a.m., and he was told that there was a problem of some kind on the network. At this stage, facts were scarce, but the disturbance was known to be widespread.

Andy walked to the hotel car park, got into his car, and drove to the Maersk office. This took about 15 minutes, during which Maersk lost nearly all its systems. When Andy arrived at the office, he saw that Maersk's systems in Maidenhead were down. However, the company had data centers around the world, whose status had to be determined immediately.

Establishing connections became a problem. All Maersk's computers and servers had crashed, so email and the company's internal chat system were down. To his surprise, Andy also found that the contacts on employees' phones had been wiped clean, because they were configured to fetch contacts from online servers, which had now been deleted. The Maidenhead office realized

that it had no contact information for international branch offices and employees, aside from what had been on computers and smartphones.

Andy told me that keeping the phone numbers of key personnel in hard-copy format at all times was among the lessons learned. Such a sheet of paper will probably never be needed, but if the need arises, it is sure to be acute.

Once all the phone numbers of international branches had been gathered, Andy began to grasp how desperate the situation was. It was a case of total destruction—all Windows machines had been wiped clean. As the afternoon wore on, Andy wondered whether this concerned only Maersk, or if all Windows machines in the world had been destroyed. All Maersk's systems were down, but was it the same everywhere? Looking into the street from his office window, Andy saw that trains were running and people were walking into shops—and breathed a sigh of relief. Only Maersk's data systems had been destroyed.

It took weeks for Maersk to recover, based on a massive operation. Maersk booked hundreds of hotel rooms near its office and brought in experts from around the world to help. The loss of all the company's AD DC servers was particularly problematic. Active Directory (AD) is the authentication system for Windows networks. It knows the names, email addresses, and passwords of all users, as well as what users can access. This information is shared within the network by domain controller (DC) servers. Maersk had 151 such servers in different parts of the world. If the DC servers failed to work, the company would have no network.

There were no backups of Maersk's DC servers, because all 151 servers were perfect copies of each other. They continuously and automatically synchronized user data between each other. Effectively, this meant that they had 151 backups of the DC servers, as you could never lose all 151 servers at the same time…except for when Notpetya struck.

Maersk's IT department worked day and night, only to confirm that data on the DC servers had been permanently overwritten and could not be restored. This had occurred on 150 servers; however, they also found that one DC server was missing. This server was located at Maersk's Lagos branch in Nigeria. The investigators found out that, by accident, there had been a power outage in Nigeria just when Notpetya struck. This had taken the Lagos server offline, and by the time it was over, Maersk no longer had a network for the server to rejoin. It had been left isolated, outside the network but full of critically important information. It was now the only server that could be used to restore the company network.

An attempt was made to transfer the data to the UK over the Internet, but this would have taken days due to slow international connections in Nigeria. Nobody at the Lagos office had a valid visa for travel to London, so the server was taken to Lagos airport, where a London-based employee picked up its hard drives. The IT worker sat in first class on the flight to London, with hard drives—literally worth more than their weight in gold—in their cabin luggage. The drives arrived safely, and the rebuilding of the network could begin.

One of the problems with a rescue operation of this magnitude is that the existing systems are untrustworthy. Nobody wants to reactivate malware and allow it to destroy data that has already been restored. So, Maersk wiped all the data on all of their laptops. Next, they proceeded to update the operating system on all workstations.

Before the attack, Windows 7 had been used everywhere in the company, and transferring to the newer Windows 10 had been repeatedly delayed due to user pushback. Now, when the users finally received their computers, they all had Windows 10—and nobody complained. "Easiest OS update ever," a Maersk technician told me, probably tongue-in-cheek.

Notpetya was able to enter Maersk's internal network via its branch in Ukraine but should never have been able to spread so widely. Maersk made numerous mistakes in its network structure and user privilege levels. On the other hand, the company did a lot of things right and is now in a much better position than before.

Maersk deserves special commendation for its by-the-book crisis communication, which began immediately, was transparent, was endorsed by senior management, and was done several times a day. With the company's own servers unavailable, communication was handled on the social media, mainly Twitter and Facebook. Perhaps some of the truck drivers waiting at the port saw it there.

Wannacry

Notpetya was built to destroy, but the Wannacry ransomware trojan was spread around the world one month earlier to raise funds for the North Korean dictatorship.

North Korea used Wannacry to extort bitcoins. Due to the long embargo on the country, virtual currencies were attractive since their movements could not be limited by sanctions. However, despite its wide distribution, Wannacry did not succeed in its task: its code was full of bugs, and ransom collection did not work properly. Paying the ransom did not restore the victim's data, and knowledge of this spread quickly. In the end, the North Koreans received only 60 bitcoins in their wallet, despite infecting more than 200,000 computers around the world.

The Wannacry malware contained code that tried to recognize whether Wannacry was being run on a normal computer or a testing environment for information security researchers. If the malware determined that it was on a testing environment, it would not run. The intention was to bypass automated systems

and machine learning environments at information security companies. Wannacry assumed that testing environments are not connected to the actual Internet, but to a simulated network environment. It therefore tried to contact the nonexistent address iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com. Since the address had never been registered, contacting the domain name would fail every time on a connected machine, but in simulated network environments they were likely to appear to work. If Wannacry noticed that its test connections did not fail like they were supposed to, it assumed that it was in a research environment – and played dead.

Wannacry had been spreading for only a few hours when the British researcher Marcus Hutchins noticed a domain name in the malware's code. Upon checking the owner of the address, he saw that the address was unregistered. Without further thought, he registered the address for $9 and pointed it at a server that would accept any connection requests. The global spread of Wannacry stopped instantly. Whenever Wannacry infected a new computer, it tried to contact the test address, got through, and refused to spread any further. Wannacry had destroyed hundreds of thousands of computers, but all others were spared. Marcus had saved the world for just $9.

My Week with Wannacry

The Wannacry malware epidemic of spring 2017 was unique in the field of information security. Quite by accident, I had promised to keep a diary of my working week for the computer culture magazine *Skrolli*. Wannacry struck that very week, adding a historic malware attack to an already hectic schedule. This was one of the biggest epidemics of all time. What follows is my diary for my week with Wannacry.

Tuesday, May 9, 2017 In the morning, I drive to the F-Secure headquarters in Ruoholahti, Helsinki. I am looking forward to a few meetings and clearing my inbox. During the day, I give a speech at a cloud services seminar in a hotel in central Helsinki. After the seminar, I return to the office to host a group of visitors in the laboratory. The RF lab, the Faraday cage, is always a good topic of conversation.

Wednesday, May 10, 2017 In the morning, I head for the Expo and Convention Centre Messukeskus to give the opening speech at Process Days. The speech ends at 10, and in five minutes I am in a taxi, heading to the airport to catch the 11:05 SAS flight to Oslo. A hacker conference called Paranoia is starting there on the same day; t2 in Finland, Sec-T in Sweden, and Paranoia in Norway are the best hacker events in the Nordic countries. Wednesday's keynote speaker for Paranoia is Charlie Miller, the world's most famous car hacker. Charlie and I go way back and we have a long chat in the afternoon. Charlie mentions in passing that, according to the press, the vulnerabilities that he and Chris Valasek found cost Chrysler 14 billion dollars. "If Chrysler had paid us 10 billion, we wouldn't have told anyone about what we found, and they would have saved 4 billion!" After the conference, the speakers have dinner together at a nearby restaurant. There are at least two former NSA employees at my table. After dinner, I sit in the hotel lobby for half an hour, talking to a local cryptography entrepreneur.

Thursday, May 11, 2017 I am still in Oslo and give the morning's first keynote presentation at Paranoia. The annual event is being held for the tenth time, and I talk at length about how much the world has changed in just a few years. I tell the audience that 10 years ago, everyone had a Finnish phone in their pocket—now,

none of us do. Time flies, especially on the Internet. After the session, I open Twitter since I know that F-Secure has published a stock market release: we are acquiring an English information security company. I tweet about it and welcome our new employees. I manage my Twitter account in the taxi because at noon I will fly from Oslo to Stockholm. IDG's annual Internet of Things (IoT) event is being held there. My slot is after a woman who designed Internet connectivity for Volvo vehicles and a man who connects reindeer to the Internet. I talk about the future of IoT and its pros and cons. After my speech, I take the Arlanda Express to the airport and board the evening plane for Madrid.

Friday, May 12, 2017 In Madrid, I speak to a group of around 150 clients about corporate information security solutions. I have a few hours to kill between my speech and return flight, but I prefer to return to the airport and work in the lounge. I barely have time to sit down when my phone rings. Our headquarters in Finland ask me to contact one of our major clients in Spain. I call a member of the management team and hear that brand new malware has infected thousands of computers on their network. Worse still, the infection is still spreading. And worst of all, it is a new ransomware Trojan—WannaCry—that encrypts files. Wannacry spreads like wildfire, becoming the largest ransomware Trojan epidemic in history during the same day. It almost exclusively targets large corporations. By evening, almost 200,000 machines are infected. My phone keeps ringing until it is time to board the plane. I even get in several calls while on board, because our departure is delayed. We then hear that there is a thunderstorm at my stopover in Frankfurt, and the plane arrives two hours late. I still make the flight to Helsinki, since it too is delayed by two hours. I land in Helsinki at three on Saturday morning.

Saturday, May 13, 2017 I get less than six hours' sleep before the circus resumes. By morning, our team has completely decompiled the WannaCry code. It also turns out that WannaCry has stopped spreading, after a British researcher I know found a function in the code that allowed him to stop the epidemic globally. A deed worthy of a medal! I give two international, live TV interviews from my home over Skype video. One of them is for BBC World, which has dozens of millions of viewers. We talk about how companies all over the world have been infected: Hospitals, car factories, power plants, train companies….

Sunday, May 14, 2017 I spend Mother's Day wrestling with WannaCry.

Monday, May 15, 2017 On Monday, we find out that WannaCry has done much more damage in Asia than expected. Infections in corporate networks went undetected before the weekend. The phone keeps ringing, as I receive a barrage of calls from clients and the media. I record a two-minute summary of the situation and respond to radio interview requests by emailing the soundbite. Radio stations in at least South Africa and Austria play the file unedited to listeners. CNN calls me and wants to hold a video interview over Skype. It will be broadcast live for international distribution. The interviewer is a female journalist from Hong Kong, who I've met several times over the years. Before going live, we reminisce about interviews on the Sony Rootkit in 2005 and Conficker in 2008. After the CNN interview, I take a taxi to Helsinki Airport.

As I write this, I'm sitting on the Finnair flight to Barcelona. Tomorrow, we will be starting F-Secure's key annual client event, involving operator clients from all over the world. It's not difficult to guess that WannaCry will be the number-one topic.

As an information security worker, I truly need to go where my work takes me.

Targeted Ransomware Trojans

In 2019, we started seeing more and more attacks based on targeted ransomware trojans. In these, the criminal gang carefully selects a target, penetrates a company's local network, and may spend days or even weeks there before tripping file encryption. The aim is to gain maximum access to all the company's files and backups. In such cases, the ransoms demanded are significantly higher than normal, often amounting to millions.

Lockergoga, Ryuk, DarkSide, and Megacortex are examples of groups performing targeted attacks. For example, the Norwegian aluminum manufacturer Norsk Hydro, which has operations all over the world, battled Lockergoga in 2019. Ryuk, for its part, gained notoriety by infecting the hospital networks of universal health systems during the covid-19 pandemic in the fall of 2020.

When the pandemic started in early 2020, I sent out a public message to ransomware trojan gangs, asking that they stay away from hospital networks and organizations carrying out medical research. Five gangs then issued a statement in which they agreed not to touch hospitals. Those groups were Netwalker, CL0P, Maze, Nefilim, and DoppelPaymer.

When DoppelPaymer encrypted the servers at the Universitätsklinikum Düsseldorf six months later, we were taken aback—but so were the group's members. The German police contacted the group and explained that the trojan had encrypted the hospital's systems in the middle of the pandemic. The members replied that they thought it was a university network, not a hospital. They then handed over the decryption keys for free, cut off all communication, and vanished into the dark recesses of the Internet.

Ransomware Trojans v2

If I had to say something positive about ransomware trojans, at least they have substantially improved backup routines in companies. Major corporations are now much more diligent about backing up data held on all computers and ensuring that the backups are recent, kept in safe offline storage, and can be comprehensively and quickly restored. The criminal gangs behind ransomware Trojans have also been paying attention.

In 2015, we encountered the malware, Chimera, which targeted home computers. Like other ransomware Trojans, it demanded a ransom in bitcoin for the restoration of the encrypted data, but the ransom note also gave an ominous statement: "If you do not pay, the photos and videos found on your computer will be published online alongside your name."

Chimera was the pioneer. In 2020, the group Maze started using the same tactic against companies. The group ran a server that listed the names of its victims: a car parts store from Chicago, an electronics company from Madrid, a chain of drapery stores from Germany, and a consulting company from Britain. Threats were not enough: if companies did not pay, stolen information such as email server contents or document archives started appearing on the site.

No companies want to suffer leaks like these, and even good backups cannot combat this type of extortion. The tactic works so well that, during 2020, it was adopted by many other ransomware trojan groups such as DarkSide, Ryuk, Conti, and CL0P. Double extortion became the norm.

I think that we have only seen the beginning of the ransomware trojan problem.

The Human Element

All the security problems we've seen can be split into two groups: technical problems or human errors. Fixing technical problems can be hard, slow, and difficult, but fixing human errors might be impossible.

The Two Problems

In the end, all information security breaches or data leaks are attributable to errors in technology or human error. Technical errors, such as vulnerabilities in online services caused by programming mistakes, may be difficult, slow, and expensive to fix, but at least they are fixable: find the bug, fix it, find all the vulnerable systems, and update them. Human errors, on the other hand, are practically impossible to fix.

People tend to:

- Use the same password across all services
- Open a remote connection to their computer when a scammer requests this over the phone
- Run shady utilities downloaded from the Web or install unnecessary browser extensions
- Use their computers with administrator credentials, even when this is not required
- Open each and every attachment sent by email, no matter how shady the sender appears to be
- Open Office documents found online and click "Enable content" when the document tells them to
- Fall into traps and enter their credentials on phishing sites

No patch or hotfix is available for the human brain. The only way to update people's skills is through training. However, after decades of experience, I can say with confidence that training nearly always fails. Regardless of how many times you tell users not to open every email attachment, they typically do so anyway.

All company networks will always be vulnerable to human error. Perhaps it would be wiser to grasp the nettle and say that some users should not be given responsibility for their own

information security, as they are unable to handle it. Modern society requires being online, but how much information security know-how can we expect from pre-teens or pensioners, for example? The right answer may be to remove responsibility from users, who are incapable of bearing it, and shift it to where it belongs: with operating system creators, software companies, telecom operators—and information security companies.

The Heist

Our consultant, Tom, often works with international banks. He specializes in testing security systems by getting into high-security environments on request of companies that want to find out how hard (or easy) it would be to break into their environment. Sometimes, he attempts to break into systems to test their security, while at others, he focuses on physical intrusion to be able to demonstrate the risk or as a way into a high-security environment that is otherwise not exposed to the outside world electronically.

Tom had a new assignment from a bank that had never used our services. They wanted to know if criminals could access the bank's main computer system, an IBM mainframe used for all client transactions. Instead of cloud services, banks around the world are still running their systems on IBM mainframes, which most commonly use z/OS.

At the initial briefing, Tom asked the head of IT security about the kinds of attack we could use. The answer was: "You can do whatever real criminals might do." At this point, Tom knew that he was going to succeed.

In penetration testing such as this, the first task is to garner basic information on the target: what type of organization are we dealing with, which systems are they using, and where are the systems located. Most basic information is readily available online. Recruitment ads, in particular, usually explain the company's technology choices in detail.

Tom quickly found out that the bank employed dozens of system specialists. By scanning their LinkedIn profiles, he quickly determined what types of mainframes they were maintaining. The next step was to find their names and contact information. The goal was to acquire administrator credentials by creating a fake company and approaching them with a proposal for

new services that almost seemed too good to be true. Spoiler: they were.

Tom goes through the usual motions to make sure that any kind of trivial background checks would not show anything suspicious. A website was set up for a fake company demonstrating remote mainframe access products, masquerading as a perfect match for the target bank's environment. Four-color brochures were printed and sent by mail to the administrators' work addresses. The envelope came with a demo of a "revolutionary z/OS remote access application," saved on a USB stick, while the colorful wrapping had slogans printed on them. Going on about how this software solves a list of typical mainframe operator grievances that one could have when operating IT equipment that was already in production when the Beatles were still together.

Resembling normal memory sticks, the USB sticks came with an additional piece of programming that mimicked a keyboard, a removable drive, or both. And not just that, the removable drive was programmed to change function on demand by the red team. When the stick was connected to any computer, the operating system assumed that it was a new keyboard and received inputs just as it would from a human typing. After being connected to a computer, our tampered USB stick waited for about 20 minutes before issuing a few lightning-fast commands to install a key logger on the machine. The logger stored all the usernames and passwords used by administrators to log in to the mainframe. Then the stick disappeared from the system and came back as a removable drive with a convenient little program on it that was able to send all the retrieved keystrokes—including passwords—home to the red team's command-and-control server on the Internet in a safe and secure manner.

We sent 20 USB sticks to different administrators at the bank. A week later, four had been connected to administrators' computers and were busy logging passwords. Bad passwords.

At this point, it seemed that our mission had almost been accomplished. However, the passwords that were stolen were useless if we couldn't actually reach the mainframe, since the mainframe could not be contacted directly. Keys are worthless without access to the door.

The bank's head of IT security had seemed confident that their system was safe, and we soon found out why. It seems that critical systems were being maintained from another system with no connections to the outside world. You could contact the mainframe from only a single unmarked office building near Copenhagen, Denmark.

Far from being disheartened, Tom started thinking about ways to enter the building. He eventually found a press release on the bank's investor website, inviting senior-year university students to an event to discuss their potential upcoming careers in the Nordic banking world.

Tom created a fake Gmail account for his alter ego and sent an email to the contact person mentioned in the press release, posing as a career promoter working at the university specializing in career paths for senior-year students. The bank's representative was delighted, as this was the first time they were selected by a university that had contacted them.

The day of the event came and Tom, sporting a suit, tie, and university-branded jacket and briefcase, found himself sitting near the reception desk in the bank's office. With a dozen journalists and students, he was escorted to an elevator and then an auditorium, where a detailed presentation began about the Nordic banking world. Tom later told me that the hour spent in the auditorium was absolutely the dullest in his life at that point.

At a convenient time when one speaker followed up another, Tom rose from his seat and asked the clerk standing at the rear of the auditorium for directions to the men's room. Tom entered the men's room, chose a stall, locked the door, and waited. The hour

spent in the bathroom was even duller than the previous one, but at least there is no one judging you being on your smartphone for an hour.

Finally, Tom was certain that the event was over and he was no longer needed. He took off his visitor badge, fished a white labeled card out of his bag and attached it to his belt, and stepped out of the men's room and started walking along the bank's corridors, as if he knew what he was doing and where he was going. People don't challenge you if you walk with purpose. Tom was now inside the right building and blending in, with no one giving him a second glance. Tom looks around and tries to find out where the toilets and coffee machines are located on every floor. When trying to explore areas you haven't been to before, bathrooms allow you to hide, while coffee machines allow you to pause with an excuse, look around, and plan your next move. For red teamers trained in the dark arts of physical and cyber intrusion testing, life is just a series of toilets and coffee machines.

Tom walked around until he found an empty desk. He sat down and took his computer out of the briefcase. To allay suspicions, he made no attempt to hide what he was doing. Acting calmly and casually, passers-by assumed that he was in the right place.

Tom hooked up his computer to the network cable of a workstation, waiting for his network authentication bypass script to run; received an IP address for the internal network; and scanned the target subnet until he found the mainframe, right where it should be, based on previous intelligence gathering and spoofed phone calls to the IT support department pretending to be the new guy from the other office. He then opened a new session window in his mainframe software until the remote connection popped up. When the computer on the other end asked for a login and password for the specific mainframe environment he was asked to break into, Tom knew that he had succeeded. The credentials harvested earlier with the keylogger had

worked—mission accomplished. Tom marked the time and took a few screenshots to demonstrate what kind of access he had to certify that, as an outside attacker, he had indeed gained access to the inner sanctum.

Before packing away his computer, Tom decided to test one more thing. He pinged the mainframe from his PC, and the responses came back in 10 milliseconds. This meant that the mainframe was nearby—in the very same building. Of course it was. That's why the mainframe couldn't be contacted from anywhere else. Tom chuckled to himself at the thought of finding the mainframe and taking a selfie with it. Physical attacks were in scope for the project, and he was already inside, so why not give it a go? He could then attach the selfie to the final report. So he set off again into the office corridors.

Based on experience, Tom knew that mainframes like this are often located in the basement, so he headed for the lower floors while taking pictures with his smartphone of all the fire safety floor schematics that hang on the wall near every exit in order to get his bearings and not get lost. Getting to the lower levels of the building is usually tricky because access cards were needed to open internal doors, usually in both directions. If this was any other red team test, then he would have already cloned someone's access card. But right now, he wanted to see if he could get in without an access card. The trick is to wait for someone else to open doors for you. There are two different approaches to this: you can either wait quietly for the door to open and then try to sneak past, hoping nobody will notice, or walk up to a person approaching the door and strike up small talk as if you would have met before—and walk through the door with them. In other words, if you can't be clandestine, be bold.

Tom managed to go down several floors in this way before his luck finally turned. Upon entering a long corridor, Tom noticed two people standing in the corridor. To make matters worse, one

of them actually was a presenter in the auditorium and actually welcomed him in the lobby this morning. Tom tried to turn, but it was too late. The host recognized his face and was confused.

– What in the world? Weren't you one of our guests at the event this morning? What are you doing here? Are you lost?
– Yeah, I'm lost. Very lost.
– This is a security violation. I need to take you to reception and call security.

While the host and Tom were waiting for an elevator, Tom realized that the situation made no sense. Getting caught didn't matter, as his mission was already complete, and he had an out-of-jail letter in his back pocket for this reason. He could just as well tell his host the truth.

So he decided to do just that.

– Tom: "From a stranger to a stranger: do you believe in coincidences?"

The host stares into the distance and turns his head toward him: "That…depends."

– Tom: "I'm afraid I haven't been entirely truthful with you."
– "Is that so?" says the host, now looking at him from top to bottom, trying to assess what is actually happening.
– Tom: "I don't work for the university, and I'm afraid I don't specialize in career paths."
– "You don't?"
– Tom: "No. I work at F-Secure, and I'm performing a security penetration test for this bank on request of your security department."

The host was confused and paused to think for a moment. Then they continued:

– "Wow, that sounds like a lot of fun! Well, enjoy your testing!"

With that, the host walked away.

Stunned, while waiting for his eyebrows to return from the back of his head, Tom stood next to the elevator for a while. He then headed for the basement where a small security desk was located. After getting buzzed through the glass gate using another excuse of a forgotten power cable, he found a way into the machine room housing the IBM mainframe and took a selfie alongside it, which he duly attached to the final report.

CEO Fraud

CEO fraud, also known as BEC fraud, is a good example of how criminals make money by fooling users. BEC stands for *business email compromise* and refers to outside parties breaking into an organization's internal email traffic, allowing them to scam employees.

Fake invoices are a similar problem. Companies were already receiving authentic-looking fake invoices by mail or fax back in the 1980s. These were usually sent in the quiet summer months, in the hope that summer employees in finance would simply pay all incoming invoices without asking too many questions.

BEC fraud slowly shifted to the Internet. The basic tactic is to find people in the organization with access to the company's finances. Attackers previously tried to identify key individuals by calling telephone exchanges at random, but LinkedIn and online recruiting have now enabled snooping across borders and language barriers. Recruitment ads can tell you a great deal about a company's structure, the tasks of key personnel, and the technology being used.

Having identified their targets, attackers would approach them, posing as the company's management and typically assuming the role of the CEO, CFO, or a board member. Employees may be contacted—by email or telephone—by fraudsters using an aggressive or persuasive tone. However, the goal remains the same: to bamboozle the victim into thinking that the company urgently needs to settle an invoice sent by a third party.

Who's to Blame? Successful CEO fraud commonly makes the news, because huge sums are often involved—losses of dozens of millions of dollars are not uncommon. Since many of the victims are publicly listed companies, they usually need to disclose the heist.

When media reports of such cases, the reactions tend to be similar: "How stupid do you have to be to send the company's money abroad?" or "Christ, more idiots falling for these Nigerian scams!"

Laughing at the victims of crimes is always wrong. Also, you should never assume that BEC fraud is easy to avoid, because that is simply not the case. Modern BEC fraud can be very well executed, employ several attackers, and the attack can last several weeks or months. In fact, both Google and Facebook have lost dozens of millions to the same BEC thief!

The person behind the fraud was the Lithuanian Evaldas Rimašauskas, who had founded numerous companies with names identical to Google's and Facebook's actual partners. Rimašauskas was caught because his personal email address was found in the registration data of one of the fake company websites. He had replaced his own address with a fake one immediately after registration, but his personal data could still be extracted by examining the history records. In the end, this led to Lithuania's extraditing him to the United States, where he was sentenced to five years in prison and a fine of more than $26 million in 2019. He also had to return $49 million in criminal benefits. Crime only pays when you don't get caught.

Targeting F-Secure All kinds of companies may be targeted by BEC attacks. This is illustrated by attempts made against F-Secure, despite the attackers' almost certainly being aware that it is an IT security company. One of the attack attempts was exceptionally well thought out. The attacker posed as our CEO but did not target anyone from the finance department at our headquarters. This was wise; our CFO, for example, could simply walk across the floor and ask our CEO about any strange message seemingly received from him. The chosen victim was a business controller from our Asian headquarters in Kuala Lumpur.

The fraud started with a single-line email message whose address information had been forged to resemble our CEO's address. The content simply read:

Hi. Are you available? I will call you in 10 minutes. Thanks.

Put yourself in the recipient's shoes. The CEO wants to talk to you. Regardless of whether you are in a meeting or busy somewhere else, you are likely to be able to answer a call in 10 minutes.

Sure enough, in 10 minutes, the controller's phone rang. The caller may have sounded like our CEO, or perhaps not. On the other hand, the victim had never met the CEO. The caller spoke English and went straight to the point.

– "Are you alone? Can you talk in private?"
– "Yes, I can."
– "Good. Listen carefully. I am putting you onto our company's insider list. This means that, by law, you are not allowed to disclose anything discussed in this call to anyone. If you have any questions, contact either myself or our general counsel. Do you understand?"
– "Yes, I understand."

The attacker had planned the tactics well. F-Secure is a publicly listed company, and all such companies have lists of insiders. The conversation is more or less what happens when you are actually added to such a list for real. Convincing the victim that they are about to receive insider information achieves two things. First, the victim cannot easily ask anyone for advice. They cannot go to colleagues and ask what to do when the CEO asks them to settle the invoices. Second, the attacker tries to win over the victim by boosting their self-esteem. They have been specifically chosen by the CEO and are being entrusted with secrets. Once colleagues

get curious and ask what the CEO was calling about, the answer, of course, is "Unfortunately, I can't tell you, it's classified."

The scammer continued by explaining that our company was acquiring another company, and as the deal would affect our share price, everyone informed would be added to the insider list. The company being purchased was (surprise, surprise!) from Mainland China, and part of the purchase price would need to be paid to China. The CEO had called the controller, as money transfers to China are easier to make from our Asian Headquarters than our World Headquarters in Finland. This is actually not true, but it sounds credible enough.

Although the attempted fraud was very well conducted, it failed. It failed because we had already identified all people in our organization who are allowed to make money transfers of this kind and trained them in how to detect fraud. Furthermore, we had built a protocol that employees can use any time to check whether they are actually on the insider list. So, in this case, the victim immediately assumed that they were a target of attempted fraud. In fact, they even recorded the phone call for our research. Unfortunately, we never found the attacker—or at least we haven't found them yet.

Touring the Headquarters

I visit a lot of company clients around the world. During a quiet moment, I often ask the hosts to give me a tour of their offices and show me around a bit. The answer is usually yes, and we take a brief walking tour.

The tours always proceed in roughly the same manner: "This is our design department, this is human resources, this is our VP of strategy, and here is our R&D and production." At some point, I always ask where the Finance department is, and it is nearly always on the top floor of the headquarters. "Please meet our CFO, and this is our group controller."

I then ask them who in finance is responsible for settling invoices and whether I can meet them. So far, the person responsible for settling invoices has always been a middle-aged clerk. Larger organizations may have many people in this role, but they have always been very pleasant middle-aged people.

I interview them about their work:

– "Nice to meet you. What do you do here?"
– "Well, I pay invoices."
– "OK. How many do you get a month?"
– "Depends on the month, but hundreds or thousands."
– "So you are moving large amounts of money around?"
– "Yes, maybe a million dollars a month."
– "In practical terms, how do you pay invoices?"
– "I log in to the corporate online bank, use a smart card for authentication, and type in the invoices."
– "Can you show me the computer you use for this?"

The clerk walks up to their desk and points at their computer. It is usually a desktop PC built by Dell or HP and running a version of Windows; however, the OS version is rarely the most recent one. It has Internet Explorer, or maybe Edge, as the browser.

That's fine. I have one more question."

– "So, this is the computer that you use for paying the company's invoices?"
– "That's right."
– "Can you now show me the computer you use for everything else? The one you use to surf the Web, go on Facebook, read your Gmail and watch YouTube videos?"

At this point, my counterparts are usually briefly confused. They may be pointing at the same desktop computer they already showed me, but are thinking about what they just told me: they are using Facebook on the same computer they use to transfer millions of dollars of the company's money.

Their desktop computer is the one to which cybercriminals are itching to gain access. The more this PC is used for unnecessary things, the more likely it is to end up in the wrong hands.

Computers are cheap; a new one costs a few hundred dollars. It is easy to buy two computers for each invoice settler: one dedicated to invoice processing and related tasks, and another for everything else. To be clear, I am not suggesting that the computer used for invoice processing should be offline. Of course, this would be more secure, but it would also make work almost impossible. It is enough that the computer is not used for unnecessary activities or for running software unrelated to settling invoices.

IT security is not always rocket science. You simply need to consider how to make life harder for attackers.

Protecting Company Networks

Question: How many of the Fortune 500 are hacked right now?

Answer: 500.

If a company network is large enough, it will always have vulnerabilities, and there will always be something odd going on. Each of the Fortune 500 companies has more than 100,000 workstations around the world, which will always include hacked or infected systems.

Companies do not go bankrupt because they get hacked. I know of only a handful of companies that have been wiped out by a security breach or leak. Companies seem to recover fairly well, even from large-scale attacks. However, their management seldom recovers, and we often see cases where the CEO, CIO, or VP in charge of IT is shown the door after a hacking incident.

Companies do their best to protect their data and systems from attackers. There was a time when company networks were not connected to the public Internet and network protection mainly consisted of protecting modem-based remote connections and dial-up pools.

Nowadays, every company has a massive number of devices online. Not all are computers, though, which makes them harder to protect: you also need to consider printers, routers, security systems, photocopiers, and videoconferencing systems, for example.

The general model for protecting a company network is simple: keep attackers out by building walls around the network that are impenetrable to outsiders. Sounds effective and simple—but whenever something sounds simple, it's usually too simple.

If the only principle applied to company network protection is that firewalls, filters, and gateways keep unwanted guests out, defenders will focus all their attention on this. And if the sole focus is keeping attackers out, no one watches the internal network—but that is precisely what they should be watching. This is perhaps where organizations fail most, and the reason is clear: after investing a great deal of money and effort in building walls around a network, nobody wants to assume that this carefully constructed protection will be breached by attackers. However, this is precisely what they should assume.

If a network's guardians anticipate failure from the beginning, their attitude will change. The focus will shift from building walls to monitoring internal network traffic. The vulnerability of firewalls becomes evident in a situation where an employee's laptop is infected with malware from a hotel Wi-Fi on a business trip, for example. They will set their infected laptop into sleep mode, fly back to their home country, and carry the infection straight through the doors of the head office. Once they connect to the internal network and resume from sleep, no firewall can prevent access to the internal network.

Network Profiling Profiling has become established as the most effective technique for monitoring internal networks. This means monitoring network traffic to create a snapshot of normal everyday life. Once you know what is normal, you can start spotting what is abnormal. In practice, profiling is begun by installing a large number of sensors in a network, devoted to logging data on traffic and its nature. Such sensors do not try to identify or stop anything, but are perfectly passive.

After creating a snapshot of the normal situation, protection systems can use a range of techniques to discover anomalies. For example, it would be of interest if devices that had never

communicated before began sending data to each other. It would also arouse interest if a user who usually handled accounting data started going through product development folders. While a single occurrence may not mean much, machine learning and smart alerts yield genuine results.

Like any other security arrangement, profiling has its limits and weaknesses. If a network has already been compromised when the sensors start logging, a clean baseline can no longer be established. In this situation, the sensors create a profile of a network where attacker traffic is normal and no longer raises the alarm.

False alarms are also very possible. We had been using a sensor-based system to track a client's network traffic for several months when we received an alert. The client was an R&D company from Germany. Our AI had detected suspicious traffic and flagged it to our analyst for investigation. The analyst concluded that the event was real and called the client to give them the bad news: you clearly have an intruder on your network who has accessed one of your R&D servers and transmitted gigabytes of data to Mainland China. The transfer occurred at 4 a.m.

In a few hours, the client called back and told us that we were wrong. There was no security breach; everything was fine. Instead, one of their project managers had come to work early, since an important project for a Chinese customer was running late and had transferred database files for the project from one server to another. False alarm. Nevertheless, the client was impressed that our profiling had picked this up. Despite the logical explanation for the network traffic, it definitely did seem suspicious.

Bait Networks You can also use various types of bait to detect intruders in a company network. These include honeypots and canary tokens. A honeypot is a computer that attracts attackers. This is done by leaving the computer unprotected and filling it

with information that seems interesting. Canary tokens, on the other hand, refer to the birds that miners once carried deep into pits. The bird collapsing in its cage was a sign of poor air quality. In the world of networks, canary tokens send alerts when touched. These baits can be totally passive, or they can appear to be real devices or attractive documents but are actually fake. Their only purpose is to raise the alert when accessed.

To this end, you can create a folder structure on a company's document server with attractive document names, such as *\\srvplatform5\documents\LT-team\Q4\mergeroffer_v7.xlsx*. Such a file would be left unused, but an attacker exploring the network would be unable to resist opening it. The same would be true for something like *\\srv_it_houston\filesrv\it\johnson\passwords.txt*. Simply opening the files would trigger an alert, helping to detect the intruders much faster. In the same way, you can create entire fake servers, routers, or usernames on a network to trigger an alert when accessed.

Canary techniques have also been successfully used to identify decompiling program code. Decompiling can be made difficult but is impossible to prevent. The next best thing is the developer being informed when someone manages to remove the protections on their code. This can be achieved by placing API calls inside the code: these are normally left unused, but a cracker will try to play around with them.

The Shotgun Honeypot documents are an interesting alternative for defenders. I remember how, a few years ago, we were brainstorming new security features in the lab when it hit me: we would create a honeypot document containing exploit code!

The idea would be similar to canary tokens, but taken much further. Attractively named PDF, Word, or Excel documents would be left in the document folders of the network being protected. We could assume that uninvited guests on the network

would download the interesting files…and open them on their own computers. The sting in the tail was that the documents would contain attack code that would take over the computer.

The code would then tell us who was viewing the documents. I even thought of adding a function that would use the PC's camera to take a picture of the person opening the document. Radical, no doubt, but nobody asked them to open the documents. It was nobody's fault but their own.

Unfortunately, the idea failed to take off. Our legal department mulled it over before deciding that it would be too risky. The precedent they had been studying involved a summer cabin. Years before, a cabin was being regularly broken into, and no alarm or camera would help. Finally, the frustrated owner rigged the door with a shotgun that would fire automatically if the cabin was broken into. They were apparently convicted of manslaughter.

Network Boundaries Are Disappearing In many organizations, the traditional division between internal and external networks is becoming fuzzier. The more companies use cloud services outside their internal network, the less traditional internal network traffic there is.

Enterprise networks have been built over several decades and change slowly. Previously, a startup would purchase a dedicated computer for each employee and then purchase a server for file sharing. Another server was then required for email, followed by an IRC server for chatting.

New companies now act differently: a laptop—or a Chromebook—for each employee, and that's it! Files are shared via Dropbox. Gmail is used for email and Slack for instant messaging. AWS, GCP, or Azure is chosen for storage, and GitHub is used as a development environment. Everything is online and in the cloud. As needs grow, services can scale dynamically.

We now have a generation of developers and administrators who have never touched a physical server—and never will.

Zero Trust

In 2010, Google was subjected to an exceptional security breach. Chinese spies had penetrated Google's internal network and had been gathering data there for a long time. While similar cases of espionage had occurred before, Google was the first company to communicate openly on the matter.

The event had far-reaching consequences. Google exited the Mainland China market and has not really returned since. However, the change in how Google approached its network development was even more profound. Google's engineers received support and funding from senior management for a project now known as BeyondCorp.

The BeyondCorp model is Google's version of a zero-trust network. In this model, the company no longer has an external or internal network; it just has a network. The organization's resources and services are available regardless of time and place. To the user, it no longer matters whether they are in a conference room at company headquarters or an airport café. The BeyondCorp model is built around identity and device management. Access control decisions are now at individual user and device level—access to information is provided according to what the user needs. The traditional all-seeing administrator role no longer exists. The BeyondCorp model also makes use of cloud services that are as seamless as in-house services.

While the BeyondCorp model eliminates many traditional problems, it is not easy to deploy. Even Google needed several years. On the other hand, we know of no successful hacks at Google during the BeyondCorp era. This is quite an achievement, as Google must be one of the key targets for foreign intelligence services almost everywhere.

Bug Bounties

Software companies try to scour their products for bugs and vulnerabilities. Even so, they always miss some. One way to find otherwise unnoticed vulnerabilities is to use a bug bounty.

In a bug bounty system, a company gives outsiders permission to break into its systems or crack its software protection. While this may sound radical, it is an everyday phenomenon: the world's largest software companies have been offering bug bounties for many years. They are used by Microsoft, Apple, Facebook, Google, and Tesla, for example. The Pentagon also runs a similar program.

F-Secure has been running a bug bounty for a decade. The program we use at F-Secure is genuinely open. How open, you ask? So open that, right now, I'm giving you permission to hack F-Secure. You can break into our system. I promise that we will not call the cops, as long as you follow the terms of our program. Our key term is that if you do find holes in our systems, you tell us about them. In return, we will pay you for your discoveries.

We offer bug bounties because we know that our systems are being targeted. F-Secure has enemies. Many of our clients are organizations targeted by foreign espionage. So we want anyone who finds a vulnerability in our software to tell us—and not someone else. In fact, we hope that they will be willing to sell the information to us instead of a third party, as there are buyers lined up.

Vulnerabilities have become commodities. If a company does not offer a bug bounty, a researcher may consider selling their discoveries to someone else.

For example, on its website a company known as Zerodium maintains a list of prices it is willing to pay for previously unknown vulnerabilities in Windows, Android, iOS, macOS, WhatsApp, Chrome, Outlook, Office, Signal, Telegram, and

various IT security software. The prices range from $100,000 to $2,500,000. The higher end, however, is offered only for very advanced exploits that succeed without user interaction.

Zerodium and equivalent companies sell the vulnerabilities forward, usually to government buyers. Sellers could get an even better price by selling the security holes to criminals.

Thankfully, most security researchers are unwilling to sell their discoveries for use in attacks. Even if the company's bug bounty is lower than the price paid by exploit merchants or criminals, many researchers prefer to sell directly to the company itself. On the other hand, if the company pays nothing to researchers, other options may seem more lucrative. A bug bounty is therefore a sensible choice for companies that are interesting to attackers.

Some IT security researchers support themselves by collecting bug bounties. For most, however, it is just a hobby, a source of minor additional income, or a place to hone their skills. F-Secure has typically paid bug bounties of a few hundred or couple of thousand dollars.

HackerOne is a company specializing in crowdsourced bug bounties. Over ten years, it has paid out more than $90 million to people discovering vulnerabilities. Six researchers have made more than $1 million through the service. The largest individual payments have been in excess of $100,000.

Offering a bug bounty requires maturity and discipline from an organization. It is not a shortcut to eliminating security holes. There is no point in introducing bug bounties until an organization has found and fixed all its obvious bugs. A new program will generate a vast number of reports about bugs that need to be fixed, particularly in the early days. A bug bounty can fully employ an organization's coders and testers for a long time.

For me, the best aspect of bug bounties is that they make probing for vulnerabilities legal. I visit educational institutions

and universities often, meeting young people who are itching to engage in some hacking. It's great to be able to say "Go ahead and hack Apple or Tesla if you like. It's legal, permitted, and you get paid—as long as you follow the rules and don't goof around."

Enable Content The security features in Microsoft's Office products are a good example of how attackers choose the path of least resistance.

Office products are user-programmable. They can be controlled with a Visual Basic dialect known as Visual Basic for Applications (VBA). VBA originally came out with Microsoft Excel 5.0, in 1993. It was later built into all other Office applications (Word, PowerPoint, Project, Access, etc.).

Typical office applications can be programmed with macros. In its simplest form, a macro can add a company's street address to a document based on a specific hotkey. At the other extreme, you can use macros to build an entire accounting system into an Excel document.

As always, technology can be used for both good or bad ends. The first macro virus was discovered in 1995. It spread from one Word document to another, copying itself as users created or edited documents. Unaware of the stowaway, they spread the virus from computer to computer by sharing such documents. Once a user's installation of Word had the virus, it infected every file passing through it.

Some macro viruses (such as Melissa from 1999) actively spread themselves by email, but most have become worldwide epidemics simply because users share documents. Users share documents more than any other form of data. In fact, a large part of a normal office worker's day is spent reading, writing, or creating documents. This makes documents an extremely effective vector for spreading malware.

Ultimately, macro viruses became the world's largest malware-related problem, and Microsoft took action. Its solution was fairly harsh: VBA and macros were turned off by default. Microsoft Office 2000 continued to support macros, but whenever a user opened a document containing them, macros had been turned off. To run macros, the user had to click Enable Content on the application's toolbar.

This must have been painful for Microsoft: VBA took a long time to develop, and it had just become substantially harder to use. In terms of information security, however, this was a major change that killed off Office macro viruses. Within a year, they were off the list of the most common virus epidemics.

Having focused on macro viruses for a number of years, attackers began seeking easier routes into users' computers. These became known as exploit kits—systems that could access computers via websites, commonly using vulnerabilities in Flash or Java. After a few years, Adobe and Oracle, the companies behind Flash and Java, learned their lesson and secured their systems.

Somewhat surprisingly, attackers returned to Office macros around 2017. Attackers always choose the path of least resistance—while macros had become more difficult to exploit, so had everything else. In the end, this turned out to be the easiest way to penetrate computers. The weakest link had been found when malicious documents managed to convince users that macros needed to be enabled for one reason or another.

Nowadays, when a user opens a malicious Word or Excel document, they can see the document's contents, but the potentially harmful content, macros, will not run. The user must still click Enable Content. A user may even believe that clicking this is mandatory, if the document can convince them of this. For example, the document may display a fake error: "Your version of Word is out of date. Document contents cannot be displayed. Please click 'Enable Content' to view." Or: "This document

contains classified information. The contents are encrypted for security reasons. Please decrypt by clicking 'Enable Content'."

Note, however, that Enable Content and Enable Editing are different features in Office. Enable Editing will not run macros, whereas Enable Content will.

I know a lot of Microsoft employees, including people who work on Office products. I have told them several times that it might be a good idea to rename the Enable Content button. If it said "Infect my system" instead, people might not be so eager to click it.

Bugs Why can't Microsoft or Apple create an operating system with no security holes? That's a good question and one I hear regularly. Unfortunately, the answer is that they are incapable of doing so.

There is no magic to security holes or vulnerabilities. They are code, just like any other. Software has security holes because programmers are human and make mistakes.

Programming errors, or *bugs*, have not always created vulnerabilities. Above all, this involves bugs in systems that are connected to networks, that is, the Internet. Before systems went online, security problems barely mattered, since the only way to exploit vulnerabilities was to sit at a computer. If a malicious attacker gains access to a physical device, they have many ways of accessing its data.

A bug is easy to create. It can be a small typo or additional character among thousands of lines of code. The end result is an application that appears to work but will crash under certain conditions—or create a hole that an outsider can use to access the system.

At least some readers will recall how Microsoft Windows was in the early 1990s. Externally, Windows 3.0 and Windows 3.1

were quite similar to current versions of Windows. The user interface was roughly equivalent to modern Windows versions: Word and Excel have changed little in 30 years. The main difference was that Windows 3.0 lacked a web browser. This was not needed, since Windows 3.0 did not support the Internet. Windows 95, released in 1995, was the first version that could go online straight out of the box.

Many will also recall how buggy Windows systems were. Word, for example, continually crashed. However, in 1992 this was not a disaster. All you had to do was restart Word, load the latest version of the text you saved, and keep working.

Nowadays, bugs like these are potential security holes. A bug that once only crashed software or an operating system may now offer attackers a way in.

Over the years, programs have become so massive that mistakes are unavoidable. Microsoft Windows is built from a source code repository that contains 5.7 million files—you read that correctly. And that's files, not lines of code. Each source code file can contain dozens of thousands of lines of code.

Another good example is the PDF file format standard. Adobe developed the PDF file format in 1993, and it became an ISO standard in 2008. You can purchase a precise description of the PDF file format from ISO. It costs about $200, and, yes, it is also a PDF file. The standard has 962 pages. It would take a week to read.

Complexity is the enemy of security. The more complex our systems become and the more code they contain, the more bugs they will conceal.

Some bugs are logic bugs. In 2021, a huge list of 533 million Facebook users' phone numbers was posted on the Web. Facebook quickly released a statement assuring us that it's important that the phone numbers were not stolen by hacking, but by scraping. Security of the system was not broken; the attacker just asked nicely. Effectively, the attacker created an address book with

every phone number on the planet and then asked Facebook if his "friends" are on Facebook. As an end result, lots of politicians, celebrities, and people with abusive ex-partners had their secret phone numbers exposed.

Vulnerabilities result from bugs in systems. Bugs result from programmers making mistakes. Programmers make mistakes because they are human. The solution is therefore obvious: we need to replace programmers with machines!

I am not joking. There will probably come a time when we develop computer software that is better at programming than humans.

The Safest Operating System So, why can't Microsoft create a secure version of Windows? Perhaps we are looking at the wrong version.

The Windows running on our computers is massive, as it can be used to do anything and run software made by anyone. It needs to support a vast range of hardware combinations and provides legendary support for legacy applications. For example, a Commodore 64 emulator written in 1996, originally created for Windows 95, is installed on my computer. Despite being more than 25 years old, the software runs perfectly on 64-bit Windows 10.

You can gain perspective on how massive Windows is by looking at its size. The WINDOWS folder in a fresh installation contains more than 84,000 files and pointers and takes up more than 14 gigabytes of space on a hard drive. That's a lot of space for bugs.

But what if there was a version of Windows that supported only one specific set of hardware and could run only approved and tested software? Surely that would be a lot smaller and more secure? This operating system exists—in the Xbox.

The Xbox One and Xbox Series X are game consoles from Microsoft that run a version of Windows with a similar security model to that found in mobile phones. You can only install software from the app store, to which only tested software from approved developers is admitted. In other words, the safest operating system from the world's largest software provider is on a game console!

Protecting Software When a programmer writes software, another programmer can find out how it was done. If a program is in an interpreted language, such as Python or JavaScript, the source code accompanies the program. If a compiled language like C++ or Swift has been used, binary code can be investigated using reverse engineering or runtime analysis.

Sometimes, programmers take extra steps to make their code hard to examine or understand. Such measures are typically seen in malware, copy protection, or anti-cheat functions in online games. Malware coders and virus authors are engaged in an eternal struggle with security companies. They try to make their code as complex and difficult as possible, to make work harder for virus researchers.

Illegal copies of commercial software eat directly into the profits of software companies, which is why closed-source developers have built various copy protection methods. Crackers, or people who circumvent copy protection, are willing to spend much time and effort removing and understanding protections.

Modern online gaming is more than just entertainment; it is a major business. In addition to spoiling everyone else's fun, cheating at online games allows cheaters to win prize money in gaming tournaments. Modern PC games may contain low-level drivers whose sole purpose is to identify whether a process is attempting to examine or modify the game's operation while it

is running. Typically, the code in these protection systems has been carefully obfuscated.

An endpoint was reached in 2019, when a coder known as Speedi13 published cheat software for the multiplayer online game Counter-Strike. To be more specific, they released a compiler for cheats that work on Counter-Strike regardless of the available protection systems.

One anti-cheating technique used by Counter-Strike and other games involves protecting the game's allocated memory from being overwritten or modified. The ROP compiler created by Speedi13 compiles cheats in a way that creates no new program code. Return-oriented programming (ROP) refers to an exploit method where the program is constructed from small pieces of code already in the computer's memory. It is as if the running game's code consisted of small puzzle pieces. The ROP compiler finds suitable pieces. When these code pieces are executed in the right order, the result is an entirely new software application—in this case, a cheat that the player can use to see through walls, for example.

The amount of work required to develop something like this is insane. It feels like some people are enjoying the game by playing it, while others enjoy circumventing its protection. On the other hand, games are a huge business. In terms of money spent, gaming eclipses movies and music combined.

Wi-Fi Terms of Use

> Question: What is the most common lie on the Internet?
>
> Answer: "I have read and accept the terms and conditions." Nearly all applications and services require the user to accept a litany of legalese. The fact is that nobody reads it.

A couple of years ago, we tested this in practice. We set up the F-Secure Wi-Fi Hotspot service and ran it in the center of London, in places like Piccadilly Circus, for a few days. When users joined our network, they were told that it was free to use as long as they accepted the terms of use. The terms appeared typical at a glance, consisting of around four pages of convoluted legal language in uppercase:

> THESE TERMS OF SERVICE & ACCEPTABLE USE POLICY GOVERN YOUR USE OF SERVICES PROVIDED BY F-SECURE. YOUR USE OF THE SERVICE REPRESENTS YOUR AGREEMENT TO THESE TERMS. IF YOU DO NOT AGREE WITH THESE TERMS, DO NOT USE THE SERVICE. WE RESERVE THE RIGHT TO MODIFY OR DISCONTINUE, TEMPORARILY OR PERMANENTLY, AT ANY TIME AND FROM TIME TO TIME, THE SERVICE (OR ANY FUNCTION OR FEATURE OF THE SERVICE OR ANY PART THEREOF) WITH OR WITHOUT NOTICE. YOU AGREE THAT WE WILL NOT BE LIABLE TO YOU OR TO ANY THIRD PARTY FOR ANY SUCH MODIFICATION, SUSPENSION OR DISCONTINUANCE OF THE SERVICE. SUBJECT TO YOUR ACCEPTANCE OF AND COMPLIANCE WITH THESE TERMS, YOU ARE HEREBY GRANTED THE RIGHT TO USE THE SERVICE THROUGH A NON-EXCLUSIVE, NON-TRANSFERABLE AND NON-ASSIGNABLE LIMITED

> LICENSE. THE SERVICE IS PROVIDED FOR YOUR USE ONLY (UNLESS OTHERWISE SPECIFICALLY STATED) AND YOU AGREE NOT TO REPRODUCE, DUPLICATE, COPY, SELL, TRANSFER, RESELL OR EXPLOIT FOR ANY COMMERCIAL PURPOSES YOUR SUBSCRIPTION TO OR MEMBERSHIP IN THE SERVICE, ANY PORTION OF THE SERVICE, USE OF THE SERVICE, OR ACCESS TO THE SERVICE.

However, there was an exceptional condition towards the end:

> IN USING THIS SERVICE, YOU AGREE TO RELINQUISH YOUR FIRST BORN CHILD TO F-SECURE, AS AND WHEN THE COMPANY REQUIRES IT. IN THE EVENT THAT NO CHILDREN ARE PRODUCED, YOUR MOST BELOVED PET WILL BE TAKEN INSTEAD. THE TERMS OF THIS AGREEMENT STAND FOR ETERNITY.

Hundreds of people used our network, gladly forfeiting the right to their child. We were not surprised by the result, but we did have an internal discussion about whether we should pick up a few kids just to make a point. A fun idea, perhaps, but we decided not to go through with it.

Mikko's Tips

How do you tackle the most common problems in which human behavior is the weakest link? Because I'm often asked about this, I've decided to compile some answers here:

I met a guy in the corridor at work. I'd never seen him before. He was wearing a suit and seemed friendly. However, he was not wearing his access card, even though this is policy. Should I have challenged him about this? It would have felt a bit embarrassing. Perhaps the easiest option, after all, was to let him go and leave it to someone else to challenge him?
In a situation like this, always assume that it is a test to see if you know how to act correctly. The guy you met in the corridor may well be an IT security consultant hired to test whether a stranger with no access card gets noticed. If you do not intervene, you may be penalized for slacking. The best thing to do is to grab the bull by the horns and ask the guy where his access card is. If the card has been lost, take him to security at reception to pick one up. Security will get to the bottom of the matter.

I have received an email from my point of contact with a large client. The message is a bit weird and had an attachment. Should I open the attachment, even if I don't know what the message is about? I'd hate to call the client and ask, as I don't want to come across as weird.
Do not open the attachment; send your client an instant message or text instead. Do not reply to the email, as your message may be routed to the attacker. Instead, you can text them a message such as "Hey, I got the email you sent. I will get back to you today!" If the message you got was genuine, the client will be pleased by your quick response. If, on the other hand, it was an attack attempt or a forged email, you will find out immediately, as your client will wonder what email you are referring to. Win-win.

How should I back up my own files?
In the workplace, your company's IT department will manage backups centrally. Let them take care of this. At home, however, the only person managing backups for your home computers and personal devices is you. The key issue is to back up often and ensure that you can restore from backup even if your house burns down. Make backups in the cloud or use two hot-swappable hard drives. Store the second copy at your parents or work—so long as it's not under the same roof as the data you want to protect.

I'm afraid that my ATM card will be stolen. How can I make life more difficult for thieves?
Grab a black marker and scribble the following on your card: "PIN: 3983." Do not use the correct PIN code. When the robber tests variations of the incorrect PIN code, the card gets disabled.

I keep forgetting the PIN for my ATM card. What can I do?
Do like Joey in *Friends*: go to the ATM you use most often and write your PIN code on the front of the machine with a marker. If necessary, you can easily check it there.

I have a fancy navigation system in my car. Is there anything I should consider from a security point of view?
Navigators will ask you to enter a home address. Do not do as they say. It wants to know your home address so that it can guide you back home—but if your car is stolen, do you want the thieves to know where you live? Choose another location nearby, such as the local shop. You will know your way home from there without a navigator.

Mikko's Tips for the Startup Entrepreneur

Practically every startup company ends up writing code, even if technology isn't the main focus of their company. Here's a 10-part checklist to help you and your hot new startup avoid the most common security pitfalls:

1. **Note that speed is the enemy of security.**

   The faster you move, the faster you develop, the faster you deploy—the less time you have for bug checking, quality assurance, and testing. Security is not something you can add to a ready product; it has to be built in from the design phase.

2. **Do not invent stuff that has been invented already.**

   There are trusted and tested principles that will save you time and make you safer. Definitely do not develop things such as encryption or hashing algorithms by yourself. Just don't.

3. **Trust the cloud.**

   Most startups today choose to go for cloud services such as AWS, Azure, and GCP anyway, which is also good for your security. Amazon, Microsoft, and Google are investing hundreds of millions of dollars into their security. This means that breaking into the servers that run the largest cloud providers is hard.

4. **…but use the cloud right.**

   The easiest way to screw up with cloud servers or cloud storage is to lose credentials. Make sure your developers use strong, unique passwords on all cloud services. Actually, forget that. Just make sure your developers use a password manager. Also, make sure everybody understands the risks

of posting private API keys to GitHub or pasting AWS Access keys to Pastebin.

And while we are on the topic of passwords…

5. **…make sure everybody has their mobile devices locked by default.**

   Face ID or Touch ID is fine. And make sure users enable two-factor authentication where possible. Two-factor via text messages is better than nothing, but an Authenticator app is much better. You can get fancier solutions for this, but frankly, Google Authenticator works fine. Also, do not force regular password changes on your users for no reason.

6. **Get a Mac.**

   When I walk around in startup events, everybody seems to be rocking a MacBook. Macs are great for security, but probably not for the reason most people think. macOS is actually less secure than Windows 10 in many ways. However, as Mac market share hovers only around 10 percent and most organized cybercrime gangs have existing expertise in Windows, criminals keep focusing on Windows. This is why we see much fewer attacks on Mac. Do note that Mac users fall for phishing just as easily as Windows users—and iPhone and Android users fall even better, as there are fewer safeguards on them, and detecting a fraudulent lookalike URL is harder on a smaller screen.

7. **Back up.**

   Ransomware continues to be one of the biggest problems we see. Recovering from ransomware attacks would be easy if you always have an up-to-date backup of your data. Surprisingly, many companies cannot restore their data when they are attacked. This happens often because backups are online and are deleted or encrypted by the attacker.

This is why cloud backup and Time Machine systems alone are not good enough for backup. Have regular offline backups that will survive even if your office building burns down.

8. **Patch it.**

   Update prompts are annoying, but almost always the reason for the update is security. So update your OS. Update your applications. Update your apps. This seems obvious, but updating can fail for surprising reasons. I was recently working with a university network that was hacked because of a vulnerability on a remote-access server; it wasn't running the latest version. This happened even though the administrators had enabled automatic updating on all servers. What went wrong? Well, the hard drive of the server was full, and it didn't have enough space to download the patches. So, the updates did not deploy, and they got hacked.

9. **Prepare for people moving around.**

   In the fast-moving environment of a startup, people come and go all the time. Make sure your people do not take their access rights with them. Make sure you can lock people out of your repositories and cloud systems. Make sure you can change passwords and access rights as needed. It's especially easy to get burned with shared passwords you use for your corporate social media accounts. Use password managers with shared secrets and force a password change on public company accounts whenever someone who had access leaves the company.

10. **Watch out for business email compromise.**

    Even startups get hit by these attacks, sometimes known as *CEO scams*. Make sure you know who exactly can move money in the company, and make sure they know how modern BEC scams work. These attacks are way more complex than traditional fake billing scams.

11. Double-check your stuff.

Yes, this is item 11. Make sure your developers can identify and fix the common security vulnerabilities. Then have your app security tested. Have your network pentested. Have your code audited. When you know your stuff is safe, your next challenge is to convince your customers that you can be trusted, even though you're just a startup. One tip here is to get experienced advisors to join you, validating your security process and vouching for you where needed.

Boat for Sale

My neighbor Leo saw me outside and came to ask for advice. He had listed his old boat for sale online. Immediately after posting the ad, he was contacted by an interested buyer. Surprisingly, the buyer—who called himself David—was from Switzerland, although the boat was in Finland and Leo's ad was in Finnish. David had replied to the ad, explaining that he had been seeking a boat like this for a long time and would buy Leo's right away. The buyer was willing to pay the full asking price and cover shipping to Geneva.

Nice! Initially, Leo was excited that the boat sold so quickly, but alarm bells soon started to ring. The boat he was selling was a perfectly ordinary aluminum-body fishing boat, and he wanted a couple of grand for it. Why were people in Switzerland suddenly interested? Was there a hidden agenda—and what was it? Should he take the deal or not?

I had to break the bad news to Leo: "David" did not exist, and the messages probably didn't come from Switzerland. Leo had encountered a fake freight forwarder scam. These scams are clever, as the scammer will not try to steal the item being sold. David couldn't care less about Leo's boat; the point was that it was valuable, was big and heavy, and would have to be freighted from one country to another.

If the scam had worked, David would have closed the deal, promised to send Leo the money by PayPal, and told him that he would look for a freight forwarder with experience of shipping boats from Helsinki to Geneva. A suitable company would soon be found, and the scammer would send a message stating that "International Logistics Limited" would contact Leo to agree on the pickup. Freight charges would come to 900 €, and David would send Leo payment for the boat and the freight. Leo could then pay the freight forwarder.

International Logistics Limited would be in touch soon. In addition to making practical arrangements, they would give Leo their account information for the payment. After paying, he would never hear from the logistics company or David again. Leo would then find that the advance payment from David had been cancelled, since the money had been sent from a stolen PayPal account. PayPal has a policy of cancelling all transactions in such cases. Leo would be left with the boat in his backyard, having paid 900 € in freight costs to a logistics company that doesn't exist.

If something appears too good to be true, it is. Especially on the net.

What If the Network Goes Down?

It's amazing how well the Internet works. It can't handle everything, though. Reliability is becoming ever more important as the Internet is becoming part of our critical infrastructure.

Electrical Networks

Some inventions are so good that they change the world. When a new innovation is useful enough, we no longer want to live without it—and once a technology is practical enough, it soon becomes compulsory.

Electrical networks are a good example of this. While it is hard to imagine modern life without electricity, electrical networks are a fairly recent invention. The first city-wide networks were commissioned in the 1870s, after which electricity soon became commonplace. It was first used for street lighting and factories, but rapidly found its way into urban homes.

Although cities were quickly covered by networks, it took decades to electrify outlying regions. India welcomed its last unconnected villages to the electricity grid in 2018.

Nowadays, a power outage brings everything to a halt. If an outage is extensive, not only are homes affected—shops and factories also close. Very few organizations are equipped to ensure continued operation during an extended outage. Only hospitals, data centers, and military installations have generators and diesel tanks that provide power for long periods.

Of course, battery-operated devices work during outages. Even mobile phones work, but not for long—your phone may be fully charged, but the backup power systems on the cell phone towers drain quickly.

Once these networks are down, society will be offline. Public broadcasting companies will continue transmitting, but very few people have the battery-operated radios needed to receive a signal. Information stops flowing. Factories do not operate. Pumps at service stations no longer dispense fuel. Food rots on store shelves.

Modern society could last for only a few days in a complete power outage. Once you run out of food at home, you start looking elsewhere—taking it by force, if necessary. In freezing weather, residents of rapidly cooling apartment complexes will

find warmth wherever they can. Suburban family homes with smoke rising from their chimneys may receive uninvited but insistent guests. Hunger and cold are tough negotiators.

The good news is that power blackouts are rare, and when they do happen, they are short. Typically power is restored in minutes or hours. After a big storm, some areas might be without power for a couple of days. But imagine something more drastic, like a solar storm that would cut electricity on the whole planet for a decade. Modern societies wouldn't be able to continue to function. The societies that would best survive such a disaster would be tribal cultures that are able to grow and distribute their food without electricity.

If the Internet were to fail, the impacts would be much less dramatic. Society would not stop during a network outage. Factories would continue to operate. Information would flow via TV antennas and FM radio. Of course, work would be much more difficult without network connections. Most monetary transactions would also cease.

I predict that, before long, the information network and electrical network will be equally important to our society. Before long, much like a power outage, a network outage will bring life to a halt. In fact, before long, a network outage will also mean a power outage. This might sound outrageous now, but once it happens, remember my words (and if it doesn't, never mind).

Electrical networks have been highly beneficial, but we have become tremendously dependent on them. The same is now happening in relation to information networks. The electrical network needs the information network to work, and vice versa. Technological development is changing our society in a fundamental way.

Clearly our grandfathers made a wise choice in embracing electricity, even though we have now become dependent on it. Likewise, we should not shy away from using new technologies. But we should be aware that we are making this decision now, and it will affect all future generations.

Security in Factories

"No worries, our factory systems are not connected to the Internet." I have heard this countless times, and it has nearly always proved to be incorrect.

Computers are easy to secure if they can never be accessed by malicious actors. This was still possible in the 1990s, before the Internet was ubiquitous and all workstations were connected to local area networks. At the time, desktop computers still came with physical locks and keys for locking them at the end of the day—although this usually just disconnected the keyboard.

Siemens AG in Germany began developing industrial automation systems in the early 1960s. Industrial automation basically means using computers to control facilities such as factories or power plants. Nowadays, industrial automation is a massive industry controlled by international corporations such as Schneider, Honeywell, Rockwell Collins, GE, Mitsubishi Electric, ABB, and Emerson.

In a modern factory, computers are part of information technology (IT) systems and everything else is part of operational technology (OT) systems. Individual factory machines are controlled by PLC devices. PLC stands for *programmable logic controller*, and modern PLCs are usually tiny Linux servers. They do not resemble computers, as they generally lack displays and keyboards. Being controlled over the network using standard computers, they are linked to other devices by network connectors. PLCs are usually reinforced and impact-resistant. They may be waterproof and tolerate relentless vibration, as they are usually located close to factory equipment to control and measure the operation of machinery such as conveyor belts, pumps, or burners.

PLC devices and their networks are controlled and monitored from control rooms, which use systems commonly abbreviated as *supervisory control and data acquisition* (SCADA).

PLC devices and SCADA applications are programmable, and these programs—like all software—contain bugs. Some bugs may disrupt operations, while others create security holes and system vulnerabilities. In industrial automation, however, the stakes are much higher than in standard computers. A disruption or crash may cause hot metal to overflow or poisonous gas to be released near a production line at a food factory. Every nuclear reactor on the planet is currently controlled by a computer.

As the stakes are high, you would imagine that security has been ensured at every level. Unfortunately, this is not the case. For years, security in industrial automation rested on a single cornerstone: the fact that the systems were "not on the Internet."

The principle is simple enough—placing critical industrial equipment in its own, closed networks means that only authorized and trusted users can access it, and network abusers and hackers are not a concern. There is no need for highly secure devices, if we can be sure that they will never be attacked. Unfortunately, nowadays it is surprisingly difficult to build systems that are truly disconnected from the public Internet.

TCP/IP, the Internet's basic protocol, automatically routes and reroutes traffic. If a data packet is en route from point A to point B and the link between these points is broken, TCP/IP finds an alternative route, and if that doesn't work, it will find yet another. The route can consist of countless shorter links, as long as the data finally reaches its destination. If necessary, packets may even circle the globe to get from point A to point B.

This means that a device in an internal, closed network is not really disconnected from the Internet if the internal network has any kind of connection to the outside world—regardless of how convoluted or unlikely. We have borne witness to this time and time again over the years.

A Search Engine for Computers

F-Secure has developed its own tool for scanning the Internet. The tool, known as F-Secure Radar, can scan the entire network—or at least the 3.7 billion public addresses in the IPv4 address space. Scanning allows us to look for various devices, or the services running on them. It is like a Google search, but instead of data, it looks for computers. In this sense, Radar is like the popular service known as Shodan.

One of our regular public scans involves going through those billions of addresses to find systems that are listening to Modbus traffic, commonly used by PLCs, or devices that allow remote access without logging in. Each time, the scan turns up countless industrial devices that clearly should not be connected to the Internet. We find rolling mills, bakeries, power plants, ski lifts, crematoria, waterworks, building automation for apartment buildings, steel mills, railroad traffic control systems, boiler rooms at shopping malls, pulp mills, and so on.

Sometimes, we dive deeper into why these systems are on the internet—and occasionally, there is a perfectly sensible explanation. For example, some systems are manufacturer demos purposely placed on the public Internet. They are not actually running factories, even if they seem to be. Occasionally, we see very alarming systems online, but further research reveals that they are bait systems placed by a colleague of ours, another information security researcher. A honeypot system like this seems interesting and vulnerable to attract online attackers, whose actions can be safely studied in more detail.

Unfortunately, the industrial systems we find online are often exactly what they seem—real industrial systems accidentally connected to the public Internet. We then try to find and contact the owner, which sometimes means making an international phone call.

– "Hello?"
– "This is Hypponen from F-Secure in Finland."
– "From where? Finland? How can I help you?"
– "It seems that the control interface for your factory automation is directly connected to the Internet and demands no username or password."
– "The what is connected, where?"
– "You are using the Schneider Foxboro system to control your factory, and your user interface is online via Windows Remote Desktop. The remote system is open to anyone and does not ask for a password."
– "Yes, we are using Foxboro, but it's not on the Internet!"
– "It is."
– "It's not!"
– "It is. I'm in Helsinki and looking at your system right now."
– "But it's only on our internal network and not visible on the Internet!"
– "Well, what if I press some of the buttons on the user interface here?"
– "Please don't press any buttons."

Further analysis nearly always reveals the same old story. The control systems were originally installed securely. They were in their own, closed network and inaccessible from anywhere else, but something changed over the years. An employee may have installed a remote connection to work from home, for example, or an individual modem link has been connected to the closed network, or two separate networks have been combined. The company may have merged with another, and the companies started operating on the same network after the merger. Whatever the

reason, there is now some kind of link from the internal network to the public Internet, which is all TCP/IP needs.

Slammer

An exceptional world record was set in 2003, which still stands. The Slammer worm that I mentioned earlier in Chapter 2, spread across the world, seeking vulnerable Microsoft SQL Server instances. Whenever the worm discovered a new server, that too joined the scan, and the number of infections grew exponentially. During the initial stages, the number of devices performing the scan doubled every eight seconds.

In less than 15 minutes, the worm was done: it had scanned the entire public Internet and infected all devices that it could. The worm had a 100 percent success rate, and its world record for spreading stands to this day.

The infected servers included SQL servers in the internal network of the Davis-Besse Nuclear Power Station in Idaho. How did this happen? How could a Windows worm spread from the public Internet to a nuclear power plant?

The network at the Davis-Besse Nuclear Power Station comprised several rings, each of which protected the systems below them. The network address ranges of the inner rings were invisible to the public network. However, if even one vulnerable SQL server was found on adjacent rings, it would become infected and search for vulnerable devices across its entire address space—including the company's internal address space, which was visible to such devices. This allowed the infection to keep spreading deeper into the critical network, ring by ring. TCP/IP routes both the good and the bad.

The security of internal networks must be taken as seriously as that of external networks, since they are often one and the same. Passwords and strong authentication are needed, and software

on devices must be kept up-to-date. Devices that are not updated must not be put online at any level. You also need to regularly examine your network address space from the outside: Do you really know what services or devices your company is making accessible online? Do you know, or do you just think you know?

Hypponen's Law

The Internet is the world's most complex machine.

The first wave of the Internet revolution took computers online but is already over. All computers are now online. We are currently seeing the second wave: the Internet of Things revolution. The first wave took computers online; the second will take everything else online. Before long, all devices using the electrical network will also be connected to the information network.

During a lecture, I once said that if a device is smart, it's vulnerable. This statement took on a life of its own and is now known as Hypponen's law: "If it's smart, it's vulnerable."

While Hypponen's law may be pessimistic, it is also true. By adding functionalities and communication capabilities to our devices, we are making them vulnerable. Take the wristwatch as an example. There is no way to hack a traditional, wind-up wristwatch. A smart watch can be hacked, as it runs code and is online. A smart watch is a vulnerable watch. A smart TV is a vulnerable TV. A smart home is a vulnerable home. A smart car is a vulnerable car, and a smart city is a vulnerable city.

IoT devices might be safe when installed and configured correctly, but their users don't read the manuals. Do you remember what VCRs looked like? If you visited friends in the 1980s or 1990s, their living room had a large video cassette recorder below the TV—and on the VCR display the numbers "12:00" were ceaselessly blinking, year after year.

Why were these numbers blinking? Because when you switched on the power for a VCR, it couldn't tell the time but assumed that the user would read the manual's detailed explanation of how to set the time. However, nobody ever read the manual or set the correct time. In the same way, modern users

do not check the manuals of IoT devices for information on how to segment their local network or change the devices' default passwords.

I once told the VCR story to an international audience, and a middle-aged gentleman rose to his feet and told me that I was wrong. You see, he had read the manual for his video recorder in full in the 1980s and set the precise time. I asked if he was German. He was.

Dumb Devices

Although I am worried about the security of smart devices, I find dumb devices even more concerning. As consumers, we know that smart devices go online. The buyer of a television knows that it needs to go online to access Netflix or Disney+. The buyer of a smart fridge buys it so that, when out shopping, they can open a remote connection and use the fridge's internal camera to see if they are out of milk. Internet connections in smart devices are obvious to consumers in one way or another.

Before long, however, dumb devices will go online. These are devices that consumers don't need smart features or apps for, such as kitchen mixers or toasters. Such devices will go online because data is valuable. Kitchen mixer manufacturers understand the value of knowing how their products are used or where their customers live. Gathering analytics like this is easy if devices are online. However, mixer makers can't yet put their products online, as this is too expensive.

For the manufacturer, a typical IoT connectivity chipset and the required connection contracts typically cost a few dollars, maybe five bucks. This is way too much, when you can buy a cheap mixer off a supermarket shelf for $15. Manufacturers can't raise their prices by a third without adding new features, so dumb devices are not online. Yet. The price of technology keeps falling, and before long, IoT connectivity will be available for kitchen mixers for five cents, or less than one cent. Kitchen mixers will go online when it's cheap enough to do so.

Consumers probably won't even notice the change, continuing to buy mixers at the same price they always have. IoT mixers will not be online to offer new features to the user, but to gather data and send it to the manufacturer. Everything we connect to the electrical network will, sooner or later, also be connected to the information network. The IoT revolution has begun and will

happen, whether we like it or not. It will happen, whether we agree or not.

Resistance will be difficult or impossible, as you will not be able to keep your dumb devices offline. They will not use Wi-Fi networks or any other infrastructure that we can manage ourselves. Future devices will go online using 5G or 6G networks or, say, Sigfox or LoRaWAN technology. Before long, devices won't even work unless they can go online.

Securing domestic and other, similar devices will be essential, but we can't protect them in the same way we protect our computers and smartphones. Installing antivirus in a dishwasher will not work, and firewall software cannot run on coffee machines.

In the IoT era, manufacturers will be chiefly responsible for securing devices. This is a difficult equation, as price is the key selling point in domestic appliances. Buyers shopping for washing machines compare prices, washing results, and exterior colors. Few will ask questions about information security or how actively vulnerabilities in the firmware are being patched.

If an individual manufacturer takes security seriously and focuses on it, their products will be more expensive. This will hurt their key selling point—price—while adding no features demanded by buyers.

Regulation

Regulation is commonly suggested as a solution for the security problem: if manufacturers won't take care of their information security, we'll make them!

I'm not a fan of regulation. In fact, I think that regulation almost always fails. An example is the cookie legislation resulting from the EU's ePrivacy Directive. It is this that makes us accept browser cookies on every website. This decision resulted in a lot of work for very little gain. The idea behind it was good—to make end users more aware of how services on the Web can follow their actions—but the end result is frustrating, and users simply dismiss the cookie pop-up on every page, without giving it a second thought.

Having said that, IoT may be the one area where we need regulation. I don't think that we should require manufacturers to build specific security arrangements for their devices—making them liable for the damage caused by security holes will be enough. If your new washing machine catches fire and your house burns down, the manufacturer is responsible. If you are electrocuted when using it, the same is true. But if all your home computers are locked by a ransomware trojan allowed into your network by a vulnerability in your washing machine, the manufacturer is suddenly off the hook.

This is something we need to consider further, as connecting "things" on the Internet will become a problem of asbestos proportions.

In its day, asbestos was a magnificent new invention, a brand new material that was fireproof and provided excellent thermal and electrical insulation. No wonder, then, that asbestos was used for so many purposes. It was added to insulation, plaster, paint, adhesives, plastic carpets, and vinyl tiles. Asbestos was even used

in clothing and to make artificial snow. You could sprinkle realistic snow, made from asbestos, on your Christmas tree.

It wasn't until much later that we discovered the dangers of asbestos. Inhaling it permanently damages your health. Use of asbestos was finally banned in the 1990s.

Now that we understand what a bad idea asbestos was, we wonder what people were thinking when they installed it everywhere in the 1960s. In 50 years' time, people may try to understand what we were thinking in the 2020s, when we put every device online, with all ports open and with default passwords enabled.

Poorly built IoT devices are the asbestos of the Internet. If your device is smart, it's vulnerable.

Car Software Updates

I asked my Twitter followers how long they thought car manufacturers should keep providing security updates for the firmware or their vehicles. I received 1,200 replies, the consensus being a minimum of 25 years.

This a sensible answer, but no software manufacturer will offer security updates for more than a few years. Microsoft supports its Windows systems for an exceptionally long time, up to 10 years, but 25 years is nonetheless unrealistic.

What are we to do? Modern cars are data centers on four wheels, and their systems require security updates throughout the car's service life. Perhaps the solution would be similar to how the rest of the car is maintained. In the first few years after buying a new car, you use an authorized service providing genuine parts from the original manufacturer. As the car ages, you start using cheaper replacement parts from other manufacturers.

Nobody expects to get free spare parts for their car, and neither should you get free software updates. Once the car is old enough, the original manufacturer could release the software platform for anyone to update. As long as there are buyers, someone will write the updates—much like you can still buy new brake pads for cars built in the 1970s.

All companies are software companies, regardless of what they build. A car company, hotel company, and A/C installation company are all software companies.

Online Privacy

Privacy is dead, having died on our watch.

Most online content is funded by mechanisms that profile users and sell the profile data to advertisers. The Internet is controlled by a handful of corporations who couldn't care less about the concerns of individual users. We have become the merchandise.

Life Without Google

In many ways, the Internet has altered our perception of privacy and its nature. This is one of the Internet's downsides. We now spend half our lives in the real world and the other half on the Internet. The younger we are, the more our lives are lived online.

Let's think back to the times when people still lived in clay huts. Back then, people sought answers to questions from village elders, or perhaps the gods. Now, we look to Google. How can I hide my eating disorder from my parents? What are these strange lumps on my backside? Where should I hide the murder weapon? These are the kinds of things people ask Google. They tell Google things they wouldn't dare tell anyone else; asking Google is like sitting in a confessional. We send our deepest secrets, voluntarily, to a company in California—and it sells them to advertisers.

A couple of years ago, I tried to live without Google. I failed. Replacing Google search with other search engines was the easy part. I also found replacements for Google's browser and mobile operating system, but Google Ads and Analytics are used on nearly every website. YouTube is difficult to avoid, as are maps and office applications. Google's products are really good. My only wish is that I could pay for this with money, instead of paying for them with my privacy.

It is commonly thought that online privacy doesn't really matter if you haven't done anything wrong: I am not a criminal, so I have nothing to hide. However, even if we have nothing to hide, each of us has something to protect. Our data has become valuable merchandise and is worth protecting.

When someone tells me that they have nothing to hide, I might thank them for warning me not to share anything confidential with them—since they can't be trusted with secrets.

Murder Charges Never Expire

Online advertising follows us everywhere. Every Internet user knows that, once you search for information on a product, all websites soon advertise products related to your search.

Of course, this is not a coincidence, but a massive business.

Maintaining websites costs money, and selling space to advertisers is by far the most common way of financing content production. Advertising online is very different from advertising on the TV, radio, newspapers, or billboards.

When you read a printed newspaper, the publisher has no idea of which articles you have just read. Even during a long customer relationship, the publisher cannot form a clear picture of whether you read all travel-related articles or skip stories about cars. Online publications, on the other hand, know how long you spent on each page and how quickly you read through articles. Cookies stored on the pages allow them to identify the same user, day after day. If the user is wearing a smart watch, even information on which articles raise their heart rate is recorded somewhere.

While often thought of as a search engine, Google is actually the world's largest advertising agency.

More and more searches made with Google's search engine end with Google itself: instead of generating traffic for the original website displaying the information, Google inserts the information into the search result. While this is handy for users—since people searching for, say, opening hours can immediately see whether a store is open—it is killing the sites supported by visitor traffic. Sixty-five percent of Google searches do not lead to links being clicked in the search results. This large number is partly explained by the information being displayed in the results.

Google sets a cookie in your browser that expires every two years. However, this timer is reset every time you use a Google service. In practice, this means that Google will never forget you, unless you spend several years in prison. Murder charges and the Google cookie never expire.

Browsing the Web from one device generates a lot of information, but information is generated even more efficiently if Google can combine data from all your devices. Let's assume that Joe is a financial editor at a newspaper in Seattle. Searches made, videos watched, and websites visited on his work computer provide Google with an overall picture of a person interested in financial statements and economic trends. When Joe closes his work PC and heads home to Bellevue, the nature of the searches changes. If Joe is into, say, fishing, the searches and videos give Google a picture of a person of a certain age interested in nature, travel, and fishing.

The information gathered by Google becomes much more valuable if Joe uses his work computer to log in to Google Maps to save the location data of companies and logs in to YouTube on his home computer to comment on fishing videos. Suddenly, the situation looks completely different for Google. It understands that the Seattle journalist interested in financial questions is the same young resident of Bellevue who enjoys fishing. The profile created for Joe is now much more accurate—and valuable. If Joe logs in to Gmail on his smartphone and chooses to save bookmarks in Chrome on his tablet, the picture becomes even clearer, and tracking can merge all the devices Joe is using.

Google knows where you live. GPS data is not needed to locate Internet users. As Google's camera cars circled the globe to photograph all streets and alleys, they recorded all Wi-Fi networks they saw and their network addresses. Billions of Android users are gathering location information and sending it to California.

The benefits gained from gathering location data are obvious: data makes it easier for Google to notify users of traffic jams, for example. Traffic jams occur wherever dozens of Android phones suddenly slow down from highway speeds and start crawling.

When Apple introduced the iPhone, it used Google Maps (and Skyhook location data). It was assumed that, sooner or later, Apple would introduce its own map service for its smartphones. Surprisingly, we didn't see Apple's cars on the streets, measuring distances or calculating street lengths. There was no need for Apple to do that: it had millions of iPhone users who gathered the data on Apple's behalf. When Apple Maps was published, it had gathered precise location information on the areas where it was needed—wherever iPhone users were.

Is Google Listening to You?

A widespread myth regarding online advertising is that Google and Facebook are listening to us speak and showing ads on this basis. This is not true, even if it often seems to be.

There is a good explanation for this misconception. All of us have probably been part of a discussion on a specific topic—only to see online ads on the same topic immediately afterward. This is not a case of Google recording your speech and using it to target advertising.

The most common reason is that people take keener note of matters that they have just discussed. If your phone is showing you ads for holidays in Maui, you probably ignore them. However, if you have just spent time with your cousin in a café, discussing places to visit in Maui during your holiday, the same ad will grab you in a different way.

Another explanation is that your friend has been browsing for Internet content relevant to your discussion, even if you haven't. So, if you discussed haggis with your colleagues during your coffee break and you see a Facebook ad for delicious haggis in the afternoon, there is no conspiracy. Inspired by your discussion, your colleague has gone online for a genuine haggis recipe, from the same address space as your workstation. Sometimes, the most peculiar things have perfectly logical explanations.

Gorillas

The business models of Silicon Valley giants may appear similar, but there are fundamental differences between U.S. technology companies in terms of privacy.

Facebook	Investigates our social relationships and sells them to the highest bidder.
Apple	Sells overpriced smartphones.
	Does not compile profiles of its customers in order to sell them.
Netflix	Sells movies.
	Does not compile profiles of its customers in order to sell them.
Microsoft	Sells Office products, Azure cloud services, and Xbox games.
	Does not compile profiles of its customers in order to sell them.
Amazon	An AWS cloud services merchant that also sells miscellaneous goods online.
Google	Investigates our lives and sells them to the highest bidder.

Apple is fortunate, because it can charge so much for its products that it has no need to generate cash by invading users' privacy. Apple leverages this in its marketing, especially when taking on Google.

Google and Apple are different in many ways, mobile payments being a good example. When users pay with their smartphones, data on what they are buying is literally worth a fortune.

Advertising is easy to target if advertisers know exactly what kind of a consumer you are. A person who has bought a 12-pack of beer every Friday for the past month is likely to do so next Friday.

It is no surprise, therefore, that mobile devices running Google's Android collect data on mobile payments. iPhone, on the other hand, does not. Instead, Apple goes to great lengths to ensure that no data on purchased products is stored outside users' devices, despite that such data would be easy to sell to, say, insurance companies. iPhone is an unbelievable money-making machine for Apple—to quote the analyst Horace Dediu, "iPhone is the most popular and valuable product of all time."

People often ask me if I use Facebook. I always tell them I don't, because I can't afford it.

Startup Business Logic

Information security companies try to help distressed users to fight criminals. On the other hand, the information security industry is a massive business, the largest players having market capitalizations in billions of dollars.

The industry continuously attracts new startups—as it should, because innovations and fresh concepts keep the sector moving. Some startups succeed while others fail. This is the nature of business.

On the other hand, some startups may not subscribe to the traditional view of business. In 2018, I gave a speech at a conference in California about the future of cloud services. On the first day, I went to a nearby hotel where Google's Cloud Platform unit offered a sponsored lunch to invited guests.

I ended up sitting at a table with two gentlemen who both turned out to be Dutchmen with PhDs in computer science. The taller one introduced himself as Lucas and said he was the founder and CEO of a two-year-old startup in Amsterdam.

I inquired about their company, and Lucas explained that they were running anomaly detection inside cloud services to discover exceptional situations. Their system ran on Google Cloud Platform and was fully based on machine learning.

I asked where their machine learning technology came from. Lucas explained that they had developed it all themselves. For this purpose, they had engaged in several major funding rounds and used the money to hire dozens of doctors with degrees in artificial intelligence technologies. Recruitment in this industry is notoriously hard, but they had managed to hire top specialists by offering generous reward schemes and stock grants.

I was confused. I could see what the company was doing, but what I didn't understand was how they could ever turn a profit. In the end, I asked Lucas what their business logic was: how would they make money?

Lucas laughed and said that they did not intend to turn a profit. Their plan was for *Google to buy them*.

Biometrics

Apple is a good example of how user data can be protected. The popularization of fingerprint readers on smartphones raised concerns that data leaks would place user fingerprints in the wrong hands. This is very worrying, as users cannot replace fingerprints like passwords.

Apple began developing its own fingerprint reader in 2012 by purchasing AuthenTec, the leading research company in the industry. In the following year, Apple released its Touch ID fingerprint reader on the iPhone 5S. Apple's reader did not store the user's fingerprint, making it impossible to lose. Instead of a fingerprint, it saved a calculated one-way hash. It can be mathematically demonstrated that this hash cannot be used to recreate the original fingerprint. Furthermore, Apple saved all data related to fingerprints in the security chip on the user's phone, and it was not sent to the Internet at any point.

Apple spent hundreds of millions of dollars developing the Touch ID system, but after only four years they started ramping down the system to replace it with a more advanced form of biometric identification—Face ID.

Somewhat surprisingly, Apple's Face ID is based on the Kinect accessory for Microsoft's Xbox game console. Kinect was a popular accessory in 2010–2015, allowing the console to track player movements in front of the device, for use in dance games, for example. This was based on technology developed by a company called PrimeSense, which uses infrared light to project thousands of spots into a room, detecting shapes and movements on their basis.

Apple acquired PrimeSense in 2013, made the originally 11-inch wide Kinect small enough to fit along the upper edge of the iPhone X, and released Face ID in 2017. Similarly to

fingerprints, the user's face data is not sent to Apple but stored in the iPhone's *secure enclave* chip.

Tracking biological functions would make many new things possible. If a user was reading an e-book on their iPad while wearing an Apple Watch, you could monitor their heart rate to determine which part of the book the reader finds particularly exciting. However, we are not aware of Apple doing this—not right now, at least.

Although Google and Apple are sworn enemies in many respects, they also cooperate a great deal. Because Apple lacks its own Internet search engine, it uses Google as its default search provider in its smartphones and computers. However, Google pays Apple for this, to the tune of $9 billion per year.

In other words, Google is paying Apple to make Google the default search engine in Apple products—$9 billion per year to have Apple direct its users to Google's free search engine.

Data is money.

Antisocial Media

Facebook, Twitter, LinkedIn, Snapchat, TikTok, and other social media platforms are good business. Users come to these services for content created by other users. Netflix and Spotify pay content creators for content. Facebook and its rivals pay their content creators nothing. Instead they carefully record their behavior and sell it to others.

I often recommend that people should, just once, try social media services from the client's perspective, not just as a user. The clients of these services are advertisers. Go ahead and buy an ad for Facebook or Twitter. It's not expensive or difficult—but it will open your eyes to the level of precision to which advertisers have access when targeting their messages.

In the United States, in particular, advertising services combine various online and real-world databases for targeting ads. Twitter purchases data from credit card companies and loyalty programs, matching the profiles to Twitter users based on phone numbers. This enables advertisers to ensure that their ads are shown to people of a certain age, who buy a certain brand of cereal, or vote for a certain party.

You should bear in mind the long tail of social media. The parents of young children, in particular, should abstain from writing embarrassing stories about their children online. Writing a funny story on Facebook about how Oliver peed in the neighbor's dishwasher may seem like a good idea to his mother when he is three years old. However, when Oliver is 13, the story may still be online, where his classmates will discover it.

Many online services offer the opportunity to look for friends and acquaintances among users. This sounds harmless, but means that you are giving the service access to the address book or contacts on your phone. Unless you are absolutely sure of what you are doing, do not share your address book.

The data in your address book is not yours to give! Your friends are unlikely to appreciate you sharing their phone numbers or home addresses with random services. Many of us add other information to our address books, such as door codes or children's names. Do you really want to update this data to a random server?

Online Influencing and Elections

In my opinion, election campaigning with targeted online ads should be banned.

Almost without exception, this proposal has encountered strong opposition, mainly from politicians themselves. Opponents of the idea feel that political online advertising should not be limited and that limiting it would not even be possible. They don't believe that such an initiative would be technically doable. This is not true, since political online advertising is already illegal in many countries: Google and Facebook, for example, have blocked all political ads on their platforms in Japan, Brazil, Vietnam, and South Korea.

Online advertising platforms are built on knowledge of the user. Knowing the consumer's interests makes advertising easy to formulate and target, but in election campaigns such profiling is dangerous.

Traditional election campaigns are run on the pages of newspapers and on billboards, TV, and radio. All voters see the same ads, and candidates have a unified message they are trying to send to all potential voters.

Mass-tailoring ads is attractively easy online. You can create an ad on the fly for every potential voter that appeals to them, makes the opposing candidate look untrustworthy, or is intended to discourage voters and keep them at home on voting day. What's worse, users imagine that everyone else is seeing the same ads, when they might be the only ones seeing the ad in question. The same campaign may send completely opposite messages to different voters.

I am not saying that candidates shouldn't be on social media, nor am I trying to ban them from interacting with voters. However, I would prohibit the sale of detailed profile information to election campaigns.

Outcomes are unpredictable when online profiling is used to manipulate voters. This was seen in 2016, when Donald Trump was elected President of the United States and the Brexiteers unexpectedly won the UK's EU referendum. The Capitol riots in January 2021 were partly the result of a systematic, online disinformation campaign about the integrity of the election arrangements. A divided nation is weak, and both Russia and China are probably delighted about the current troubles of the United States.

In the early days of the Internet, parents warned their children against believing everything they saw online. Now that the Internet is part of everyday life, children seem to spot online lies much better than their parents. For some reason, parents have forgotten their own warnings and trawl online forums in search of conspiracy theories. As a result, we now have a large group of adults who believe that the covid-19 pandemic was orchestrated by Bill Gates in order to inject 5G chips into people's bloodstreams.

Privacy Is Dead

Throughout history, there has never been a way to track individuals from cradle to grave. Until now. We are easy to track, because we spend increasing amounts of time in the online world rather than the real one. Some of us even sleep next to our smartphones. Technology allows us to see where we are and who is with us, who we spoke to and when, and what we like and dislike.

When the Internet was young, it was obvious that commercial online services would soon appear—news sites, weather sites, games, perhaps even online videos! But it was not yet clear how online services could charge consumers for services. A good guess might have been buttons on web browsers enabling consumers to make micropayments for services, purchasing tomorrow's weather forecast for 2 cents, for example.

Decades have passed, and our browsers still have no payment buttons: we use data, rather than money, to pay for online content. The Web has spawned gigantic companies whose commercial services rake in billions by doing something that nobody saw coming: tracking people. Privacy died because killing it was so profitable.

The good news is that more and more of the online data traffic is strongly encrypted. Data encryption makes it more difficult to spy on people. Outsiders can see that data is moving, but the content is hidden. However, because companies like Google, Facebook, and Twitter offer their services on their own platforms, they are not outsiders. So, when a Twitter user sends a private message to another Twitter user, the message is encrypted and invisible to outsiders. It can only be read by the sender and recipient—and Twitter.

A private conversation once meant that two people discussed things between themselves, without being overheard. Now, a private conversation means two people chatting via private

messages on social media. This suddenly introduces a third party (such as Twitter, TikTok, or Snapchat) into supposedly private communications.

Encrypting content does not make the communication itself invisible. This is known as *metadata*, or data about data, providing the information, say, that two parties have had a conversation (but not the subject of the conversation).

Metadata allows technology platforms to see that you took a taxi from home to an attractive co-worker's place at 3 a.m.—but not what you had in mind. Metadata shows that you spent hours in an online discussion with Alcoholics Anonymous or a support chat for suicidal people, but not what you were talking about. Metadata sees that you are exchanging several messages with a law firm specializing in divorces, but it has no idea what the messages might relate to.

In 2014, CIA Director Michael Hayden put it very well: "We kill people based on metadata."

You will quite often hear people in the intelligence community lamenting the introduction of strong encryption. Naturally, the fact that standard consumer products, such as web browsers and WhatsApp, use strong encryption makes intelligence and espionage more difficult.

I sometimes hear claims that encryption will return intelligence to the Dark Ages. This is nonsense. In fact, we are living in a golden age of intelligence: never before has it been possible to gather so much information on people so easily. The fact that the content of discussions cannot be directly captured and decrypted is trivial compared to the data that *is* available.

Of course, thieves and criminals can now easily encrypt their communications too. Like everyone else, they can install WhatsApp, Signal, Telegram, or Wickr on their phones and talk

to whoever they like, safeguarded by strong encryption. However, this does not give us cause to limit the use of strong encryption.

The availability of a technology to criminals does not make it illegitimate, wrong, or undesirable. Technology is a tool, and tools can be used for good or bad.

Once an invention has been made, we can no longer return to a time before it existed. Strong encryption is an invention in this sense. Anyone can walk into any library in the world, visit the section on data processing, and borrow a book on how to build an unbreakable encryption system. So, there's no use crying over spilled milk.

If we were to enact a law forbidding strong encryption, we law-abiding citizens would probably comply and use artificially weakened security systems. Unfortunately, criminals do not abide by laws. That is what makes them, by definition, criminals. If the use of strong encryption is criminalized, only criminals will have access to it.

Before and After Gmail

For many people, email now means one of two systems: Outlook or Gmail.

At work, most people use Microsoft's Outlook email service, either as a cloud service on their browser or as an application run on their smartphone and computer. At home, most people use Google's Gmail service.

This division has not always been so clear. Email is based on the SMTP protocol from 1982. The purpose of the protocol was to ensure interoperability between manufacturers' email systems.

Email first became popular in Unix-based operating systems and was accessed using software such as Elm, Pine, and Mutt. In the late 1980s, Windows users started encountering software such as Eudora, GroupWise, and Lotus Notes. Microsoft Outlook only arrived in the mid-1990s, but quickly gained in popularity.

Web browsers developed rapidly at the same time, and 1996 saw the release of both Yahoo Mail and Hotmail. These two services revolutionized email: they worked on the Web and required no additional software. What's more, no email account was needed to use Yahoo Mail or Hotmail. At the time, home users tended to receive an email account from their Internet operator when purchasing a subscription. This meant that a user's email address was tied to an operator, so it would be something like hyppo5@aol.com, mikkohyp8@netcom.com, or mhypponen@comcast.net. If you switched operators, you also lost your old address and had to message all your contacts with the new one.

Operator email services, and the Yahoo and Hotmail webmail accounts, were very limited in size. If you received too many emails, you had to delete them or pay for extra space. Many operators were very strict about this: Hotmail, for example, gave you 2 megabytes, which was easily filled with just a few small email attachments.

This all changed on April Fool's Day 2004, when Google announced its new, free webmail called Gmail, which imposed no limit on the number of messages and came with a massive 1,000 megabytes of storage space. Many thought this was an April Fool's joke, as the service sounded too good to be true, but Google was serious. The very idea of an email service deleting old messages to save space seems absurd, given how self-evident unlimited email history now feels. Over the years, Google has even increased its free space offering and now providing more than 15 gigabytes per user. This is possible because Google is not trying to finance its consumer service by selling hard drive space; instead, it shows users ads based on their email messages.

Gathering data for ad profiling is fairly straightforward if advertisers have access to user emails, which show what users are ordering online, the destinations of flights they book, and who they are communicating with. One story circulating online claims that a Gmail user started seeing ads for undertakers after exchanging messages about a relative's death.

Users liked the Gmail user interface, so Gmail expanded to businesses. More than five million companies now use the business version of Gmail. Their email addresses do not end with gmail.com, but the underlying system is almost the same as that of Gmail's home users.

Nowadays, about one-third of the world's email traffic passes through Gmail. While this is a great achievement for Google, it has led to a few problems.

Google's spam filter is very aggressive, but also very efficient. Before Gmail, spam was a much larger problem than it is now. Unfortunately, this has meant that there are no longer genuine alternatives to Gmail. SMTP is a completely open standard, and many users and many companies once operated their own email servers. Now, running your own server means that your messages are likely to end up in the recipient's spam folder. A new

email server is such a rare sight that most spam filters automatically smell a rat. The attitude seems to be, "Why on Earth would anyone bother with their own email server when Gmail works so well?" My point exactly.

The second problem is that Gmail has become a key hub for logins, a single sign-on service for the entire Internet. When user passwords leak from an online game or discussion forum, using them to steal Gmail accounts is one of the most popular ways of profiting from the situation. In other words, if usernames and passwords are stolen from, say, an online gaming service, the attackers try them in Gmail. Sadly, this often works, as users tend to pick the same nickname for different services and use the same password almost everywhere, even on Gmail.

Once your Gmail account has been compromised, the game is over, as the attackers now have access to your message history. This allows them to search for information on online stores where you have set up accounts with the same Gmail address. Whenever you set up an account at an online store, it will send you a welcome email. Gmail keeps all welcome messages in your message history, making them easy for the attacker to find. As Gmail does not delete old messages, even welcome messages from 10 years ago are easy to find. The attacker now knows that you have accounts with certain online stores and that your user ID for them is your Gmail address.

The password you use for online stores is still secure, but that is of no concern: there is a magic button on the login page of each store for bypassing the password prompt. This magic button is labeled "I forgot my password." When the attacker enters your Gmail address on the login page and clicks the button, the store will send a new password—to the very same Gmail address the attacker has cracked. That is why Gmail has become a single sign-on service for the entire Internet. By gaining access to your Gmail, the attacker can get everything else.

So, what can you do? Being well aware of its role as a network hub, Google has introduced Google 2-step Verification for Gmail users. Users install the Google Authenticator app on their smartphone and use its one-time passcodes to verify each device on which they read their Gmail. When a device has been authorized once, no further action is needed. However, should you want to read your email on a new device—or if an intruder tries to access your account—it will work only with the code from the Authenticator app.

Securing your email is important, as it often opens the way to many other places. Always choose a long email password, do not use it anywhere else, and use Google 2-step Verification.

And if you want to use your own system instead of Outlook or Gmail—good luck. You'll need it.

Encryption Techniques

Encryption works. When implemented correctly, an encryption system is practically unbreakable.

But why is it that, despite encryption, we keep seeing cases of confidential information ending up in the wrong hands? These cases happen because the encryption system was poorly implemented—or because the attacker found another, weaker route.

Perfect Encryption

Can network traffic be encrypted to a completely unbreakable level? Yes, but this is never done in practice, as it would involve too much work. Data traffic encryption is based on the idea of converting data, such as text, into an encrypted format and sending it to the recipient, who restores it to its original format. The purpose of encryption is to prevent hostile parties who are monitoring traffic from decrypting it.

Perfect encryption works like this: a sender and a recipient have exchanged decryption keys. The key consists of completely random numbers, is very long, and is used only once. Encryption is performed by breaking down the encrypted text into individual characters and using the numbers in the key to perform a mathematical operation on each character—adding them together, for example. Each number in the key is used only once for each character being encrypted. The result is sent to the recipient, who performs the inverse operation.

This technique is known as *one-time pad* and is more than a century old. As an encryption technique, it remains perfect; if encrypted traffic gets into the wrong hands, all the unauthorized reader can determine is the length of the message. It is impossible for the attacker to try every possible decryption key, since the key is as long as the encrypted text itself. If the encrypted

message was only a single word like *koala*, trying all encryption keys would produce all possible five-letter words, like *otter*, *tires*, *sssss*, and, yes, eventually *koala* as well.

This encryption method is highly impractical. Generating gigantic, fully random encryption keys is difficult, and storing them securely requires a great deal of effort. Also, if we have a secure means of transmitting encryption keys between parties, why are we sending keys at all? Why not just send messages over the same, safe route?

That is why we now use encryption mechanisms that are not perfect, but are unbreakable in practice.

Unbreakable Encryption

Nowadays, we use encryption all the time, usually without being aware of it.

The data on your computer's hard drive is encrypted with an AES algorithm in both Windows and macOS. The Telegram messaging application uses RSA encryption, while WhatsApp's encryption is based on elliptical curves and the Curve25519 algorithm. iMessages on iPhones rely on AES. Your online bank protects your account data with TLS encryption, which is built on algorithms and protocols such as ECDHE, ECDSA, and AES.

All these methods use encryption that resists attempts to break it based on trying all possible keys. This would be possible in theory (in contrast to one-time pad), but impractical due to the sheer number of keys.

And how many are too many you ask? Well, let us assume that an attacker tries to crack a WhatsApp message you sent to your friend. The attacker decides to try all possible encryption keys and acquires a very high-powered computer for this purpose. Actually, let's say the attacker harnesses the power of *all* computers in the world, including every supercomputer.

If all these computers were combined to try every key, they could indeed crack your WhatsApp message. However, this would take a very long time—long enough for the sun to fade and die. Once the sun dies, your message will hardly matter, so such encryption is practically unbreakable. It is theoretically breakable, yes, but the theories don't really matter once our sun has died.

Criminal Use of Encryption Systems

When implemented correctly, an encryption system is unbreakable. This also results in problems, since criminals use the same protection systems as law-abiding citizens. Even if a cybercriminal has been found and arrested, encryption usually makes the evidence on their computer inaccessible. Criminals often use the same encryption systems as everyone else: BitLocker, FileVault, PGP, Telegram, iMessages, Signal, etc.

Law enforcement has the following ways of opening files encrypted by criminals:

- Arresting criminals with their devices unlocked
- Interrogating criminals and persuading them to give their password or unlock the encryption
- Secretly filming the criminal's actions before the arrest and using the video to deduct the correct password
- Installing malware and keyloggers on the criminal's computer before the arrest
- Cracking the password by trying different options computationally

The easiest scenario is to catch the cybercriminal in the act so that they have no chance of locking their computer and

smartphone. This allows the police to investigate files on the computer directly and read any open online discussions.

In this scenario, it is important that the suspect's devices do not lock themselves. Devices must not be turned off or allowed to auto-lock due to lack of use. In some arrests, the only thing one of the police officers does is move the mouse on the suspect's computer every minute or so and touch the smartphone's screen to prevent it from locking. There are also commercial mouse substitutes that you can plug into a USB port. Known as *mouse jigglers*, they automatically move the cursor a little, preventing the computer from locking itself.

During an arrest, all the suspect's equipment, however remotely related to the case, is confiscated and taken to a police laboratory for investigation. This may include other electronic products in addition to computers—even game consoles and smart TVs have sometimes been taken, just in case. The police want to keep the suspect's computers running during the transfer and have portable power sources for this purpose. Desktop computers require special arrangements, as they cannot be disconnected for a second. For this purpose, separate power supply systems are spliced onto the computer's power cable by peeling off its insulation. Since this must be done while the computer is running and connected to the AC power supply, do not try this at home.

If the suspect cannot be arrested with their devices switched on, the next best option is to work out the passwords in advance. In criminal cases, the police are usually granted a warrant to film the suspect covertly. A slow-motion, close-up recording of a suspect opening a laptop or smartphone, or decrypting files, usually allows the password or PIN to be read. This is particularly true if they have multiple takes of the password input from several angles.

The police may even gain permission to film the criminal at home. Officers may break into a suspect's home and

install a hidden camera to film the suspect's computer screen and keyboard.

Determining a smartphone's PIN may be possible without an image of the screen itself. In 2013, researchers from a Chinese university demonstrated that filming a person's face can reveal what they are typing, when the smartphone screen is viewed in the reflection from their eyeball! This required a DSLR camera and zoom lens but enabled the phone unlock code to be determined in nearly every tested case.

The introduction of biometrics, such as fingerprint and face recognition, in smartphones has created a new problem for officials making arrests. If a criminal's phone is locked, are the police allowed to open it by pressing the suspect's finger onto the reader or showing their face to the camera? The answer depends on local legislation, but it is legal in Finland, for example.

A quote from the Parliamentary Ombudsman's decision of 2017 tells us how a suspect's smartphone was unlocked:

> *The suspect was told that a requisite amount of force would be used to place the suspect's finger on the mobile phone's fingerprint sensor. The suspect stated that the police "can go fuck themselves" and did not agree to this procedure.*
>
> *At the start of the procedure, the suspect was sitting on a bed in the holding cell, and was carefully pushed back onto the mattress and held still. The suspect forcefully resisted the procedure by squirming and keeping their hands in a fist. The fists were nevertheless opened enough to try using the thumb and index finger to unlock the phone.*
>
> *Five police officers took part in using force; two twisted the suspect's hands behind their back, one pressed the back of their head, and two held onto their feet.*

The related laws and regulations differ hugely between countries. In 2019, a local judge in the United States adopted the contrary position to the Finnish Parliamentary Ombudsman and ruled that neither the suspect's fingerprints nor face could be used to unlock their phone.

By the way, if you use Face ID on iPhone, most iPhone models allow you to turn it off quickly by pressing the power button five times in a row. This is a good tip if you venture into an alley full of pickpockets or if a police cruiser pulls up in front of you and your phone is full of incriminating evidence.

Data Is The New Uranium

Data is power. Data is money. Data is the new oil. But perhaps the best metaphor for data is uranium.

Uranium is highly valuable and dangerous—rather like data. A pound of uranium costs more than a pound of silver, and its radiation is lethal almost forever.

Long-term storage of information is an exceptionally difficult problem. How do you store it so that it remains available for years, while keeping it secret and secure? Most information is no longer confidential after enough time has passed. While normal correspondence may be secret now, the publication of most emails will no longer matter in 20 years' time: secret information has become historical information. However, this is not true of all information. Health information, for example, remains confidential for much longer.

We have probably not fully grasped the difficulty of this challenge. How can we keep information available for decades, while keeping it secret? What is the correct approach to the final disposal of information?

The correct final disposal of uranium involves similar difficulties. A project called Onkalo was started in Finland in 1994. It is an exceptional project in many ways, particularly in terms of duration: project Onkalo should be completed by 2120. Five generations of Finnish engineers will work on the project, whose eventual closing ceremony may be attended by a young engineer whose great-great-grandfather helped to start it in the 1990s.

Onkalo is the final disposal facility for nuclear waste on the island of Olkiluoto on the west coast of Finland. Nuclear waste is placed in copper canisters and deposited around a quarter of a mile under Finnish bedrock. Filling Onkalo with radioactive waste will take almost 100 years, after which it will be sealed.

Once the project is complete, Onkalo is full, and the quarry has been sealed, we will face an interesting problem. What sort of sign should we hang on it's door?

The radioactive waste inside will remain harmful for at least 100,000 years, long enough to make the time scale difficult to grasp. It is only around 11,000 years since the last Ice Age. Onkalo could witness several Ice Ages and be buried under half a mile of ice several times before its contents are harmless.

So, a sign reading "DANGER—ACHTUNG—PELIGRO—ОПАСНОСТЬ—危险" will not be enough. If there are people at Olkiluoto in 100,000 years' time, they will certainly not understand any of our languages.

Not even pictograms would solve the problem: future humans may understand skulls or pictures of skeletons but may not take them seriously, treating them as we regarded warnings such as "A CURSE UPON ANYONE WHO ENTERS" at the doors of the Egyptian pyramids. Such warnings were ignored by modern humans, who simply laughed and passed them by.

Future humans may view us as we view cavemen: we have no way of making them take us seriously. All our attempts to warn future generations will probably fail, and leaving a message will only increase the allure of poking around the final disposal facility. The best tactic would probably be to close Onkalo, leave it unmarked, and forget about it.

CASE Vastaamo

The security breach at Vastaamo is one of the most important cases of my entire career, being unprecedented in its scope and cruelty. Although all the victims were within Finland's borders, the case is of international significance. It is also the largest case in Finnish criminal history, based on the number of reports to the police.

Vastaamo was a private psychotherapy provider established in 2008. Over the years, it grew into a center that employed 250 therapists, but something was forgotten: the patient database and its customer information were poorly protected.

In 2018, an unknown attacker found Vastaamo's patient database online and stole it. The database contained the personal information of 31,980 patients—their names, addresses, personal identity numbers, and email addresses. This included therapists' notes of discussions with patients. These notes covered the entire spectrum of life, some happy moments, but mainly sadness: depression, fear, marital issues, substance abuse, dependency, divorces, and fear of death. In therapy, people discuss matters they don't mention elsewhere. A therapist hears things that are never told to anyone else and notes down patients' opinions and thoughts of their nearest and dearest: children, spouses, sisters, brothers, and bosses. To make matters worse, some of the patients were children.

This type of health information is extremely sensitive for the rest of a person's life. Of course, health data and doctors' notes are always personal, but the fact that you had eczema or took medication for gout is only mildly embarrassing – especially if it happened 20 years ago. Things said in psychotherapy, however, can easily tear wounds open decades later.

This is what makes storing health information safely so difficult. How can we ensure that all public and private hospitals,

health centers, and clinics will continue to protect their confidential data for years and for decades?

Patient Registry

Vastaamo was unable to protect the data it was storing. Its data system had been developed in-house, storing patient data on a MySQL server that was on the public Internet for two years or longer.

Vastaamo's patient registry was so poorly protected that it was breached at least twice. The patient registry used for extortion had already been stolen in 2018, and the breach went unobserved. At least the later breach in March 2019 was noticed, having probably been orchestrated by someone who used a network scanning tool to find Vastaamo's systems. Scanners like these find poorly protected systems sooner or later. No substantial amount of information was stolen in the security breach of 2019, which was such a clumsy attempt that even Vastaamo noticed it, and patient databases were moved to a safer location. Vastaamo probably thought that it had survived with a bad scare, and no reports were made of patient data being jeopardized. Neither was the matter mentioned in June 2019, when a capital investment company bought a majority share in Vastaamo.

Everything collapsed on September 28, 2020. An attacker, calling himself Ransom_man, sent an email to key people at Vastaamo. He threatened to publish patient records online unless a ransom of 40 bitcoin was paid. At the time, this was worth around $500,000. Vastaamo contacted the police and negotiated with the extortionist. Apparently, the management considered paying some form of ransom but could not agree on the matter.

The case became public on Wednesday, October 21, 2020. In the early hours of the morning, the attacker performed a series of actions, setting up a server on the Tor Hidden Service network

and leaking patient data there. Next came a public message, left on three Finnish-language message boards: Ylilauta, Torilauta, and /r/suomi on Reddit. On Reddit, the attacker created the username /u/vastaamo. On Torilauta, the name ransom_man##HIbGCf was used. In this case, HIbGCf is a unique identifier (also known as a *flair*) that nobody else can use. It allowed anyone participating in the conversation to verify that the person claiming to be the attacker was genuine.

The extortionist wrote in English, briefly explaining the extortion attempt and that, as Vastaamo would not pay the ransom, the attacker would start releasing patient data. The attacker also provided the media with a personal email address, seeking maximum visibility for the case in order to put Vastaamo under pressure.

The email address was from tutanota.com and was quickly closed, forcing the attacker to switch to an anonymous address from cock.li. In addition, the attacker used the Protonmail service to send messages to Vastaamo and journalists.

Technologies

All the technologies used by the attacker were designed to be unbreakable. The Tor network, Tor Hidden Service, anonymous email services, and bitcoin all protect their user. Tor Hidden Service is specifically designed to obscure the physical location of the network's servers. If the servers cannot be found, they cannot be investigated, and information on them cannot be taken offline. Information on the Tor network is not as easy to find as on the regular Web, but Tor is fairly easy to use. The simplest way to enter the Tor network is to install the Tor Browser on your smartphone from the App Store or Google Play.

Both Reddit and Ylilauta quickly deleted the extortionist's messages, after which the conversation switched to Torilauta.

Set up in 2017, Torilauta was a Finnish discussion board on the Tor network. Its topics focused on illegal narcotics, above all. Torilauta was designed for Finnish users and tried to keep foreign spammers out. Posting on Torilauta required passing a language test: you had to confirm each message by choosing Finnish sentences from a list of five randomly selected ones, some of which were in Finnish and others in Estonian. Although telling the difference between these languages is not obvious for a foreigner, the test was not 100 percent foolproof. However, it was one factor that aroused our suspicions regarding Ransom_man's nationality.

In his messages, Ransom_man wanted to project himself as a confident cybercriminal with an international gang routinely involved in this type of extortion, which had simply happened to pick on a psychotherapy clinic from Finland. True to his threats, Ransom_man leaked the therapy details of 100 new people every morning: 100 on Wednesday, 100 on Thursday, and 100 on Friday, by when details on 300 innocent Finns could be found online. Many Internet users actively refused to read the information leaked online, while others devoured it. The victims included celebrities and politicians, in addition to ordinary people.

The case immediately became massive news as an exceptional case of extortion on both the international and Finnish scene. To be clear, this was not a ransomware trojan but flat-out extortion: the attacker had not encrypted the data but simply stolen it and had threatened to expose the information.

As time went on, however, the image of a confident career criminal began to crumble. Ransom_man started to make mistakes.

Vastaamo.tar

On Friday morning, the third day of the leak, I was scheduled to comment on the breach on the Finnish TV channel MTV's morning news. Just before entering the studio, at around

7:20 a.m., I checked the latest situation. On the attacker's Tor site, I noticed a new file called vastaamo.tar (TAR is a file format like ZIP, an archive that contains other files). What caught my eye was the massive size of the file. Up to that point, the attacker had been sharing individual patient records, with sizes varying between 2 and 100 kilobytes. However, vastaamo.tar was huge, at 10.9 gigabytes or 10.9 million kilobytes. By the time I left the newsroom, the attacker's site had vanished. During my interview, he had switched off his server or disconnected it from the network.

The course of the events was revealed later on Friday. At 3 a.m., the attacker had placed a TAR file on his server to enable easy downloading of all 300 published patient records. Sometime during the night, however, he had mixed up two files called vastaamo.tar. One TAR file had 300 patient records in txt format, and the other had a backup of the contents of the server used by Ransom_man. This backup contained information the attacker absolutely did not want to publish: server usage logs, passwords, source codes—and the Vastaamo database in its entirety. The backup was online from around 3 a.m. until around 7:30 a.m.

The attacker later commented on his mistake on Torilauta with the post "Whoopsie. Enjoy big tar." For an extortionist, there is probably no greater mistake than accidentally releasing the files you are holding for ransom. The TAR file was so large that no one had time to download all of it before the server was shut down. On the Tor network, all transfers are purposefully routed through several servers, which makes them slow. However, several users managed to download parts of the file, the start of which contained the databases with the therapy records.

For the investigators, vastaamo.tar included several leads and usable evidence. This leak makes it more likely that Ransom_man will be found one day. However, the leak also indicated that Ransom_man had been fairly skilled at using various technologies to protect his privacy.

Extortion Messages

Following the leak, Ransom_man panicked and decided to switch tactics. On Saturday evening, tens of thousands of Vastaamo's customers received an extortion message by email. In this message, written in perfect Finnish, the extortionist demanded that each victim pay 200 euros or see their records published. The message came with instructions on how to send the money in bitcoin. The extortionist also recommended using the Finnish service Bittiraha to pay the ransom. He had generated tens of thousands of individual bitcoin addresses, one for each victim, writing the following to his victims:

> *As you are probably aware from the news, we have breached Vastaamo's patient database and are contacting You as a recipient of Vastaamo's therapy and/or psychiatry services. Since the management of this company has refused to take responsibility for its mistakes, we must ask You to pay us to keep your personal data safe. Please follow the instructions below for sending bitcoins to our address. If we receive the equivalent of 200 euros in bitcoin within 24 hours, Your data will be permanently deleted from our servers.*
>
> *If we do not receive this payment within 24 hours,*
>
> *You have a further 48 hours to send us the equivalent of 500 euros in bitcoin. If we do not receive the money by then, Your records will be published for everyone to see, including Your name, address, telephone number, personal identity number, and Your exact patient record including items such as transcriptions of Your discussions with the therapist/psychiatrist at Vastaamo.*
>
> *Purchasing bitcoin: The Finnish "Bittiraha" provides an easy service allowing you to purchase bitcoin. Go to the address https://bittiraha.fi/osta and fill in your contact information. In the "total" field at the top, enter the amount of payment (€200 or €500). In the form's third field, "your bitcoin address", copy our address*

9JeUqJXu3u3Shf2ArBMZfc232PvD. This address is only for Your personal use, do not share it with others unless they are making the payment for You.

That weekend, tens of thousands of people wrestled with a difficult question. Should I pay the ransom? Will the extortionist follow through on his threat? Should I apply for a credit ban in case of identity theft? What if my personal identity number is leaked online?

The Finnish Police received more than 25,000 reports on the case. In terms of the number of victims, this is the largest individual crime in Finnish history. It is hard to imagine a real-world crime with tens of thousands of victims—it is only possible online.

We tried to track the money as it was transferred to the extortionist. I made a public appeal to those who paid the ransom, asking them to give the bitcoin address where they had paid the money. I posted this message to Twitter:

Message to #Vastaamo data breach victims:

*We are collecting bitcoin addresses to which ransom money has been paid in the Vastaamo case. We are trying to trace where the money went. If you are a Vastaamo victim *and you paid the ransom*, I would appreciate it if you contacted me. My email address is in my profile. Thank you.*

About 100 victims reached out to me. One was a young woman whose email message stated that she felt scared to tell me that she had paid the ransom. Only a few days earlier, I had appeared on TV instructing people not to pay it. I wrote back and told her that she was brave to contact me and help trace the criminal, adding that I could easily understand why some people would pay the ransom despite being urged not to.

Surprisingly many people wanted to pay the ransom but had been unable to do so. Sending bitcoin is not simple for a first-timer. Bittiraha had also stopped some of the transfers at the request of the Finnish National Bureau of Investigation. In the end, the police probably had no authority to do this, but Bittiraha complied anyway.

In the end, I gathered 33 bitcoin addresses to which ransom money had been successfully paid. As the bitcoin blockchain is public, I could track the money's movements there. The extortionist started transferring the money on October 29, 2020. A couple of days later, the criminal exchanged bitcoins for dollars at an underworld over-the-counter (OTC) money exchange service. The total amount was ridiculously small: this ruthless crime with so many victims netted the perpetrator just a few thousand dollars.

The Hunt for the TAR File

The databases related to the Vastaamo case were evidence, but having them in your possession was ethically suspect and probably illegal. When investigating the case, we were hesitant about patient data. Not wanting the victims' therapy information on our computers, we did not dare to download files from Ransom_man's server. F-Secure's Head of IT Security also explicitly banned the saving of such information on the company's computers. In retrospect, it is easy to see that this was a mistake. When a case is active, you should compile all the evidence you can. Unnecessary and illegal information can be deleted later.

Once I realized that we did not have a copy of the TAR file the attacker had accidentally leaked, I saw that we had dropped the ball. Luckily, I received a partial copy from researchers at the IT security company Nixu, who had been commissioned

to investigate the case. They sent us all the data they could. Customer and patient records cannot be shared under any circumstances, but sharing data related to the attacker was possible.

Once I saw how valuable the contents of the file were, I tried to find more of it. The longer someone had downloaded vastaamo.tar, the more evidence they would have. It might contain a vital clue that we could use to catch the attacker. I turned to the general public and, exceptionally, made a request for help on the Ylilauta message board, the Finnish equivalent of 4chan:

> *Good evening, this is Mikko Hypponen.*
>
> *We are continuing to investigate the Vastaamo case, and I have a request:*
>
> *On the morning of October 23, 2020, the extortionist "Ransom_man" inadvertently shared a large file named vastaamo.tar, which contained important evidence. The file also contained the patient records of tens of thousands of victims, which are uninteresting from an investigative standpoint. However, other data inside the TAR file (such as source codes, web addresses, etc.) could lead us to Ransom_man.*
>
> *Our investigation does not have access to the entire file, only part of it. Vastaamo.tar was available for a few hours (approximately 03:00–07:30 am) on Ransom_man's Tor server on October 23, 2020. The file size was 10.92 GB (10,918,912,000 bytes).*
>
> *In a discussion on Torilauta, someone said that they had downloaded the entire 10.92 GB file. Another person said they had downloaded 5.64 GB before the connection closed. It would be invaluable for the researchers to have such files. If you have a copy of the vastaamo.tar file with several gigabytes in size, please contact me. My email address is mikko.hypponen@f-secure.com. Wickr: mikkohypponen. I will submit the file to the authorities and I will protect your identity. If you prefer, you can delete the patient records (the directory therapissed/patients) from the TAR file.*

The discussion lit up on Ylilauta. I received dozens of messages, and a few people indeed had copies of the TAR file, two of which were offered copies of it. We never saw a full copy of the TAR file, and we are still looking for a larger version of the file.

I had nurtured the hope that the patient records would not be distributed any further, but I hoped in vain. Three months after the original leak, an unknown individual wrapped the 31,980 patient records stolen from Vastaamo in a single archive and sent it to Anonfiles, where anyone could download it.

Innocent Victims

Unlike many security breaches, the users had done nothing wrong on this occasion. There was no way that Vastaamo's customers could have prevented the breach. The therapists at Vastaamo were also innocent; they, too, had done nothing wrong, being victims just like the patients. It must have been horrifying to realize that their notes were in the wrong hands.

As a sort of silver lining, at least the case dissipated some of the shame and stigma related to psychotherapy. Vastaamo was just one private clinic in Finland, but it had almost 1 percent of Finns as customers. Seeking help is nothing to be ashamed of. You should always see a professional when you are in pain, whether on your leg or in your mind.

We have seen ransomware trojan attacks on hospitals, but ransom demands have rarely been sent directly to patients; even then, the targets have usually been plastic surgery clinics, the threats being related to photographs taken before and after surgery. Such cases are very unfortunate, but notes from psychotherapy sessions are on an entirely different level of sensitivity.

Vastaamo was also exceptional because the breach took down the entire company: the company declared bankruptcy three

months after Ransom_man made the case public. This is rare, companies almost never go bankrupt for getting hacked, no matter how bad the hack was.

You should never kick people who are already down. Only cowards do that, and that is exactly what Ransom_man did. He caused immeasurable suffering to both patients and therapists, for just a few thousand dollars.

Before the Vastaamo case, I was often asked about the security of healthcare information. I would console people concerned about their data by saying that health information is not a key target for cybercriminals. Most criminals want to make money, which is easier to do with credit card and bank information than health information. Having your health records stolen feels worse than having your credit card number stolen, but criminals prefer the latter. Unfortunately, this is no longer true. We can only hope that this type of extortion does not catch on.

Vastaamo kept me busier than any other case had in years. My phone rang constantly, and I received hundreds of tips. When working a case like this, adrenaline levels rise, and the days grow long.

About two weeks into the Vastaamo case, I was a studio guest on the Finnish Broadcasting Company's Friday morning radio show. I told the story of the young woman who contacted me despite her fear of confessing that she had paid the ransom and found myself moved to tears on live radio. At this point, I realized that I might be rather tired. I kept my phone switched off for the weekend. The hunt for Ransom_man continues.

Cryptocurrencies

Money is turning into data and will revolutionize trade.

The appearance of bitcoin and other cryptocurrencies is both wonderful and problematic—much like the Internet itself.

How does a blockchain work and why are people willing to pay real money for cryptocurrencies?

The Value of Money

Fully virtual currencies have been under development for many years. In the early years of the Web, the eCash technology being developed by the company DigiCash was a talking point. This fully digital online currency was tested in the mid-1990s but ran into technological issues. Furthermore, traditional banks and credit card companies did everything they could to block the new player.

Virtual currencies have no intrinsic value. On the other hand, traditional currencies—like dollars, euros, and rubles—have no real value either. There was a time when currencies were tied to the gold standard, but this has not been the case for decades. In principle, therefore, a $100 bill only has value because we agree that it does.

Most digital currencies now in use are based on a blockchain, the most famous of which is the blockchain for the bitcoin currency. This blockchain was published in October 2008, and its author remains unknown to this day. Bitcoin promises to create an entirely new type of currency independent of other asset types, but which can be used for payments like traditional currencies. Above all, bitcoin is free and distributed—nobody controls it. It is a currency based solely on mathematics.

Blockchains

Revolutionary innovations often seem fairly self-evident—after their invention, of course. Prior to that, they seem far from intuitive. Blockchains are such an invention. At their essence, they are nothing more than a digital list of transactions. Each time an individual transaction is added to the list, it can never be modified or deleted, and listed transactions are always public. And that's it!

An unmodifiable, public list of transactions hardly sounds like a revolutionary innovation, but it is. Building a system like this is difficult and requires complex mathematics. Cryptographic signatures enable a system where the data placed in a blockchain is locked inside nested blocks. Each new block added to the chain contains a signature that verifies the accuracy of all previous blocks in addition to the new ones. This means that the older the data is inside the block, the harder it is to counterfeit.

Entrepreneur Naval Ravikant used the following metaphor to explain the idea: "If you see a fly in amber and it's got a millimeter of amber around it, well, that could have been done yesterday or a year ago. But, if you see the fly is trapped in a huge block of amber, you know it's been there for a long time—it's been accumulating. So, a blockchain is a series of blocks. Each block is a series of computations done by computers all over the world using serious cryptography in a way that's very hard to undo."

Blockchains follow a similar logic. The information in the chain can be seen by everyone but cannot be changed by anyone. The further back in the chain the information lies, the more secure it is.

Blockchain Applications

What are the practical uses of an unmodifiable, public list? It could be used as a substitute for almost any traditional database. However, blockchains are really only useful in applications where the parties exchanging data do not know or trust each other. An example would be car odometers.

Odometers on used cars have been rolled for as far back as we can remember. You can sell a car for more money if the odometer reads, say, 100,000 miles, even if the actual mileage is over 150,000. Blockchains have been proposed as a solution to this problem. All cars of a certain model could maintain a common blockchain on the Internet, submitting their odometer readings there each week. The data would not be centrally stored anywhere—storage would be shared between all the cars.

When buying a used car, the car dealer could compare the car's odometer reading and blockchain reading. If the numbers do not match, the odometer must have been rolled back, as information stored in a blockchain cannot be edited or deleted.

Of course, you could build a similar system without a blockchain. The car manufacturer could simply set up an online service where a traditional database keeps track of vehicles and their mileages. But how long would a centralized system like this remain operational? As consumer products, cars have exceptionally long service lives. Their average service life is more than 15 years, and you can still see quite a few 1980s models on the streets.

In this case, the beauty of a blockchain lies in the lack of need for a centralized service: each car would keep track of the mileage on other cars. The system would work pretty well forever. Even when only two cars of the same model were left on the roads, they would continue to maintain the blockchain for each other. The system would stop working when only one car was left, by which time it wouldn't really matter.

Blockchains and Money

An unmodifiable list of transactions is obviously useful for tracking money transfers, as in who sent money to whom, when, and how much. The key issue is that information cannot be changed, forged, or deleted afterward. Money is commonly sent between parties who do not know (or trust) each other, which is why most current systems using blockchains involve money transfers.

Digital items are easy to duplicate, generating replicas that are identical to the original. Blockchains completely change this situation, allowing us to produce verifiable scarcity.

Fully digital money has always had two fundamental problems: how to secure money transfers and how to create new currency.

Verifying transfers between users is essential, as otherwise the users could send the same money at the same time to different recipients or cancel transfers they have already made. If this cannot be solved, the money is not usable as it cannot be trusted. The creation of new money is also a thorny problem—if you can create money from nothing, it has no value. If the amount of existing currency is substantially increased, the currency will fall prey to inflation or its value will collapse.

Dollars and euros are based on nothing, but bitcoin is based on mathematics. Real-world currencies or money transfers can be controlled or regulated, but regulating bitcoin is difficult, since mathematics does not care about regulation.

The original blockchain release from 2008 defined the first (and still the most significant) blockchain-based currency: bitcoin, which solved the two key problems besetting digital currencies in an ingenious way—by combining them.

In the bitcoin system, other users of the bitcoin network act as verifiers of transactions between users. Each transaction is confirmed and locked into the chain using complex calculations that require vast amounts of computing power. This raises

the obvious question of why other users would waste their computers' resources to confirm and secure the monetary transactions of strangers? They do so because the bitcoin algorithm rewards them for it: this is the very mechanism used to generate new bitcoins.

About 60 bitcoins are generated per hour and are awarded to users who verify transactions for other users. This verification has been termed *mining*. Bitcoin miners are not simply wasting their computer resources on complex calculations; they are mining to improve the speed and security of monetary transactions. In theory, anyone can mine, even on home computers. In practice, however, mining now requires so much computing power that specialist companies have taken over.

The Environmental Impacts of Bitcoin

Although mining is an elegant solution for verifying money transfers, it creates big problems. The logic of mining enables the conversion of computing power into money, which has had various undesired side effects. Employees have been using workstations, servers and even supercomputers for mining without permission. Malware authors have developed software that mines on other people's computers.

Mining consumes electricity and generating power impacts on the environment. The Norwegian state-owned oil company is combatting this development. Oil fields release massive amounts of natural gas, which must be burned off since it cannot be vented into the atmosphere. Because oil deposits are located far from cities, there is no sensible way to use such gas, but burning it warms the atmosphere. In 2020, Equinor (the state-owned company formerly known as Statoil) began deploying systems that burn this natural gas to generate energy for bitcoin mining. Sending bits from a distant oil field to the Internet is much easier and eco-friendlier than any other technology for using such gas.

As bitcoin's security is backed by computing power and computing power requires electricity, professional miners will automatically move to where electricity is cheapest. Otherwise, companies specialized in mining would lose competitiveness. They invest millions in efficient data centers that tend to operate near massive dams or geothermal energy plants, sources of cheap electricity. Fortunately for us, electricity generated by burning coal is expensive.

We occasionally see statistics indicating that mining consumes more power than entire cities or even countries. While the numbers are controversial, it is nevertheless clear that mining consumes a lot of energy, which is why better methods are being continuously developed. Heavy mining is not the only way

to secure transactions: the Solana and Polkadot blockchains, for example, do not use the proof-of-work method, so there's no mining involved at all.

Using technology will always consume electricity. Watching Netflix, gaming online, and making Google searches all use power. So does searching for new medicines or creating art. We should not slow or stop technological development simply because some types of power generation are polluting the environment and warming our planet. Instead, we should aggressively redirect consumption toward renewable energy or nuclear power. Blockchain mining is a good example, as all miners mutually compete—those using the cheapest power win. If we use taxation to make electricity from coal-fired power plants even a tad more expensive than, say, hydropower, miners would migrate to hydropower.

I believe that technology is ultimately not the problem but the solution for climate change. We will not give up technology powered by electricity, but technology will enable the removal of harmful carbon dioxide from the atmosphere.

Playing the Market

When bitcoin was created, it had no value. Nowadays, you can use bitcoins to buy a new car or order anything you like from Amazon.com. Over the years, bitcoin's exchange rate against traditional currencies has fluctuated wildly. Bitcoin's first actual exchange rate was set in October 2009. At that point, bitcoin was being developed by a small group of people including Satoshi Nakamoto, Hal Finney, Gavin Andresen, Jeff Garzik, and Martti Malmi. On October 12, 2009, Martti sold 5,050 bitcoins. The selling price was $5 for the lot, which meant that one bitcoin was worth a few thousandths of a dollar. In an even more famous transaction, in 2010, the Hungarian Laszlo Hanyecz paid 10,000 bitcoins for two family-size pizzas.

Many now laugh at these trades, but both were important for bitcoin's development. Martti's deal set the basis for exchanges between cryptocurrencies and traditional currencies, and Laszlo had a lot of bitcoins, no money, and an empty stomach. Paying almost anything for pizza was better than starving.

In the early days of bitcoin, online websites distributed bitcoins for free. These websites, also known as faucets, originally gave users five bitcoins for free simply for pressing a button. In the modern world, this feels completely insane—who would give away money? The point was once again that early bitcoin fanatics wanted to make the system better known and more common, to increase the value of the coins they owned. They succeeded in this, since before long one bitcoin was worth tens of thousands of dollars.

The bitcoin system also solves the inflation problem, as there is not an endless supply of bitcoins. There is no upper limit on the number of dollars, euros, or yen that can be issued. Central banks can create a seemingly endless supply of them, and printing presses can print new bills as required. Bitcoins have an upper limit strictly limited by the original algorithm: there can be only

21 million bitcoins. The rate of bitcoin creation is constantly slowing, with 19 million bitcoins already mined. The final 2 million will be mined at a decelerating rate over the next 100 years.

Bitcoin is sometimes compared to precious metals such as gold, as its mining terminology implies. Both are valuable, at least in part because they are expensive to come by. Gold must be dug up from the bowels of the earth, while bitcoin mining requires expensive, powerful, and power-guzzling computers. However, although the amount of gold is limited, we are not sure exactly how much is left—we may continue finding large gold deposits for many years to come. We may even be able to set up gold mines on the Moon or the surfaces of asteroids. The final amount of gold is therefore impossible to estimate. However, we do know exactly how many bitcoins are left.

Bitcoins are valuable because they are expensive to make, impossible to forge, and strictly limited in number. Investors want to buy bitcoins for precisely these characteristics. It's less about how bitcoins will replace dollars in everyday purchases and more about very high demand for a very limited number of bitcoins.

Hermès, a French luxury brand, makes scarves, perfumes, and handbags. It is known for its Birkin handbags in particular, which are beautiful, very well made, extremely rare—and very expensive. A new bag costs at least $10,000, while some models may cost more than $100,000. However, even if you have the money, you cannot simply buy a bag. Birkins are so desirable that there is a long waiting list for them, causing the prices of second-hand bags to skyrocket.

How did Hermès make its bags so desirable and expensive? By limiting their numbers: despite high demand, Hermès makes only tens of thousands of new bags per year. The price of bitcoins follows the same logic. Genuine bitcoins are expensive because so few are made, whereas knock-offs are cheap, as are pirate copies of Birkin handbags.

Ethereum, Monero, and Zcash

Many competitors have tried to imitate the success of bitcoin. The website Coinmarketcap lists more than 2,000 cryptocurrencies. While most offer no unique features, some try to rectify the most obvious shortcomings in bitcoin or introduce new features.

The original concept for bitcoin already envisioned that the currency might include programmable features. Ethereum takes this logic much further. Ethereum is programmable money, and its future lies in money transfers between machines rather than human-to-human transactions.

It is easy to imagine a scenario where one computer system purchases services from another if, for example, a program running on the system needs extra cloud storage space. Why should these computers trade in money designed for humans, such as dollars? It might make more sense for machines to use programmable money between themselves. Ethereum is designed for this purpose.

Bitcoin traffic is anonymous, and tracking it is as difficult as tracking cash—difficult, but not completely impossible. Monero and Zcash are examples of cryptocurrencies that have taken anonymity to extremes. The blockchains they use are not public, as in bitcoin, but use encryption algorithms that emphasize privacy, even at the cost of security. Regardless of this, tracking Monero and Zcash transactions is so difficult that the source of the money can almost never be traced.

DeFi is short for *decentralized finance*. It is a comprehensive collection of blockchain-based services aimed at creating a finance sector independent of traditional banking. In other words, DeFi is trying to take on banks. You can read more about it on the website de.fi (of course).

The value of virtual currencies is quite literally up to the programmers. Any bugs in the system might allow for duplicating

money or sending it to several addresses at the same time. So, all virtual currency systems have a generous, built-in bug bounty system—regardless of whether the developers like it or not.

Many banks and nations are already planning their own cryptocurrencies. Fully electronic money that is verified and safe to move between parties who do not know or trust each other would also be useful for traditional financial-sector operators. Facebook has been discussing its own virtual currency project for many years. Dollars and euros are the underdogs in this fight: Facebook has more users than dollars and euros combined.

NFT

Digital systems allow for lossless reproduction. When you send a picture or text file to your friend as an email attachment, both have identical files. The file your friend receives is indistinguishable from the original file, whereas NFT methods based on blockchains enable the accurate and permanent establishment of the digital original.

But why would the original file be more valuable than a copy? Because a Picasso on the wall of the Metropolitan Museum of Art is more valuable than the printed copy you have on your wall. When, say, a JPEG image is turned into a rare collector's item with the help of NFT, anyone can create a copy of the image. However, the copies are worthless—just like copies of Picasso artpieces, which is why NFTs can be valuable even if the data encoded in them can be copied.

Like all new technologies, NFTs involve a great deal of hype. I believe that the most significant uses for NFT technology have not yet been invented.

When the sportswear manufacturer Nike bought the NFT company RTFKT, many people questioned Nike's reasoning. I believe that Nike sees a future where people want to dress their online game characters in unique, stylish, branded, and rare clothes, and are willing to pay for this.

When my last book came out, many buyers asked me to sign it, which I was only too happy to do. However, it was sold in three formats: paper copy, e-book, and audiobook. Signing a paper copy is easy, but why is there no way to sign an ePub2 or MP3 file?

This will probably become possible someday, at least I hope so. Perhaps we will even see the day when Madonna, Eminem, or Shakira suddenly ask the thousands of fans in their concert

audience to take out their smartphones. "I dedicate this next song to you and will sign your copy in your Spotify library so that you can prove that you were here today." Interestingly enough, DeFi mechanisms would allow people to resell this signed original, with the artist automatically receiving a share of the purchase price.

Bitcoin and Crime

Real-world criminals have always preferred cash, which is a handy way to hide transactions. It is clear that cash itself is not criminal or evil—everyone has it—but transactions such as street-corner drug deals are almost exclusively performed in cash. Buying cocaine with your credit card is rather difficult (or so I have been told).

Online drug deals, on the other hand, almost exclusively use bitcoins, and for precisely the same reason. Like cash, however, bitcoins are neither good nor bad; they are merely a tool.

One of the basic principles of bitcoin is that transactions are irreversible: you cannot take back a transfer you make. This is not a flaw in the system, since bitcoin was deliberately built this way. For example, irreversibility prevents typical PayPal scams in which a seller is told that the products never arrived, resulting in a refund.

Criminals, too, find irreversibility a useful feature. If you can access a victim's bitcoin wallet and transfer money out, there is no way to cancel the transaction. The thieves can get the money and there's no way to prevent it.

For the same reason, cryptocurrency exchanges are constantly targeted by attacks. Traditional banks and cryptocurrency exchanges top the list of high-value targets among criminals. However, breaking into banks is very difficult due to their long experience of securing their systems and their large IT security teams. Cryptocurrency exchanges, on the other hand, are start-ups with little experience of the industry and perhaps just a handful of employees, but they may manage even larger assets than traditional banks. The Slovenian exchange Bitstamp is a good example: it has 4 million users from around the world and an annual turnover of around $40 billion. As a point of reference, the Slovenian stock market has an annual turnover of $4 billion.

Bitstamp was broken into in 2015, and 19,000 bitcoins were stolen, the equivalent of $800 million in virtual currency at early 2022 rates. Naturally, for criminals it is useful that, unlike dollars, once virtual currency is stolen it is anonymous and does not need to be laundered.

Most crime involving bitcoin is related to online scams or ransomware trojans. A typical bitcoin scam misappropriates the reputation of a celebrity such as Bill Gates or Elon Musk, appearing in their name and asking people to send bitcoins. The scammers promise to double the money and send it back. A good rule of thumb is that anyone promising to do this online is a scammer. In the summer of 2020, Barack Obama's Twitter account was stolen, and a bitcoin scam was spread in his name. Hundreds of people fell for it and tried to send a total of more than $500,000 to the scammers.

Other parties, aside from criminal gangs, have noticed the benefits of bitcoin too. North Korea has been embargoed for many years, but blockchains are unaffected by embargoes. That is why hackers linked to the North Korean government have been spreading ransomware trojans and breaking into bitcoin exchanges in Japan and South Korea. It is difficult for North Koreans to move dollars or euros around, but they can access bitcoins like everyone else.

Regulating cryptocurrencies is difficult, since they are based on mathematics, which does not care about regulation. The only party that legislators can effectively regulate are currency exchanges, where real-world currency can be exchanged for cryptocurrency, and vice versa.

Border Guards vs. Bitcoin

A television journalist from the BBC called me, and we agreed to a TV interview about ransomware trojans and bitcoin. In our background discussion by phone, he asked me to help him explain bitcoin to the TV audience. I told him that I could bring a physical Denarium coin into the studio. This is an actual coin with a holographic sticker on the back, concealing a private key that unlocks money stored at an address on a blockchain.

The journalist became excited and asked if I could bring several coins with me. We agreed on this.

The next day, I landed at Heathrow airport with three coins in my pocket: 5 BTC, 1 BTC, and one coin for 0.1 BTC. I stepped off the Finnair plane, went through passport control, and took the red lane at customs, posted with a 'Travellers with Goods to Declare' sign. I was the only passenger on my flight to do so. The border guard was stern but friendly.

"Hello, how can I help you?"

"I am entering Great Britain and have more than £10,000 in currency with me."

I was a bit unsure, but I pointed at the poster on the wall that clearly stated you had to declare more than "£10,000 worth of monetary units." The border official picked up a pile of forms.

"That's correct. Which currency are you carrying?"

"I have bitcoins."

"Bitcoins?"

"Yes, bitcoins."

"Okay…can you wait just a moment? I'll get somebody else."

I waited until another official, with some level of knowledge of bitcoin, arrived.

"So, you are carrying bitcoins?"
"That's right."
"How many do you have?"
"I have 6.1 BTC."

At the time, 1 bitcoin was worth more than $2,000, and the border guard quickly calculated that I had exceeded the declaration threshold.

"Yes, that's more than £12,000; we need to register it."
"OK, no problem."
"Are the bitcoins on your computer?"
"No. They are physical coins."
"What? Physical coins?"
"Yes, physical bitcoins."
"Okay…can you wait just a moment? I'll get somebody else."

I waited until a third official arrived. He was red-faced and surly.

"So, physical bitcoins, eh?"
"That's right."
"Physical bitcoins worth £12,000?"
"Approximately, yes."
"Pfft."

He walked back and forth, huffing and puffing. He clearly had no idea what to do with me. Finally, he asked:

"These 6.1 physical bitcoins that you are now crossing the British border with…did you declare them in any way in your country of origin?"

"I did not."

"Excellent. You can leave. Please leave."

I put my backpack on and walked through the door he was guiding me toward. Getting rid of me was clearly the easiest solution for him too.

Technology, Espionage, and Warfare Online

Espionage means gathering information.

Information was once physical and printed on paper but is now virtual. This has led to lasting changes in the work of intelligence organizations. Technology is changing relationships between the superpowers, while altering the nature of conflicts and the way we wage war.

Cyberweapons are effective, affordable, and deniable, which means that they are sure to be used in all future wars.

Cyberweapons

Cyber warfare is a complicated term, whose definition is still a subject of debate at academic conferences. Defining cyberweapons is easier: I define a cyberweapon as malicious software developed by a state actor and used for attacking another party. Cyberweapons may be used in warfare, such as the Russian–Ukrainian conflict since 2015, but war is not a requirement: they can also be used for espionage and sabotage.

Digitalization has transformed intelligence and espionage activities. Information has moved from texts and diagrams on paper to data stored on computers and information networks. Stealing, copying, or photographing information used to require the personal presence of the spy, but we can now engage in espionage or intelligence activities from across the world. There has also been a revolution in terms of data volumes. In just a few moments, spies can transfer data volumes that would take several trucks to move if printed out. Organizations tend to spot whenever a truckload of documents is driven through the gates, but a person walking out with a USB thumb drive can go unnoticed.

In a newspaper interview about a security breach by the Russian government, I said that espionage had moved from the real world to the Internet. The next day, I received a call from someone I know at the Finnish Security Intelligence Service. He had read the morning paper and told me in plain terms that I was wrong. "Espionage has not moved to the Internet—it has expanded into the Internet." He was right. Real-world James Bonds are still around, but since I don't keep up with real-world espionage, it's not something that I'm aware of. From my point of view it looked like the spies had moved into my world, while in reality they had just expanded their work into my world.

Some real-world spies are probably working as employees at the world's largest cloud services; examples of this have already occurred.

Lunch Break at Google

While visiting Google on business, I had a chat with some of the developers and researchers over lunch. I brought up the topics of government espionage and foreign intelligence organizations with my dozen or so colleagues around the table, stating that intelligence organizations from a number of countries are probably trying to gain access to Google's data. My colleagues nodded their heads in approval. I went on to say that there have been and will be attacks and that some attempts may have gone unnoticed. Once again, my associates were in unanimous agreement. Then I added that, since Google invests so much into their security, the logical way to get at its data would be to use traditional espionage and moles, that is, to infiltrate Google with spies who join its payroll and try to gain appointment to key positions.

One of the Google staff across the table said that this was likely and that moles were probably already working at Google, with nobody knowing who they were. A silence fell upon the company, and those around me exchanged glances. We finished our lunch in a much less conversational mood.

Technology and Warfare

Technology has always shaped the nature of warfare, with technological development directly influencing how conflicts are resolved. Hundreds of years ago, we fought with swords and arrows as the best available weapons. When technology made warships possible, our wars spread from land to sea. However, the invention of naval warfare did not eliminate war on land; wars were fought in several environments at once. The invention of airplanes led to aerial warfare, military satellites introduced space warfare, and cyberweapons have brought cyber warfare.

I doubt that we will ever see a purely cyber conflict, where two nations declare war on each other and fight solely in cyberspace. When a conflict escalates, events are set in motion on all fronts—including the cyber front. Cyberweapons play a key intelligence role but can also be used for direct attacks.

Compared to traditional weapons, cyberweapons are effective, affordable, and deniable. They are effective, because their destructive power is similar to bombings or missile strikes, but they are also cheaper than traditional weapons. At the F-Secure research laboratory, we have estimated that Stuxnet, one of the most important malware applications in history, took at least 10 person work-years to develop. A realistic estimate of its budget would be $20 million. It would cost more to fly one squadron of B-52s from the United States to Iran on a bombing raid.

Best of all, cyberweapon attacks offer plausible deniability: proving who was behind a network attack is very difficult. The current worldwide consensus is that the U.S. military developed Stuxnet in cooperation with Israel. On the other hand, there is no real evidence of this, and the U.S. government has never admitted to creating Stuxnet. We came closest to hearing an admission in 2012, when President Obama commented on the book *Confront And Conceal* by *New York Times* journalist David

Sangner, in which anonymous sources within the administration state that Stuxnet was developed by the NSA. Instead of denying the claim, Obama clinically announced an investigation to reveal the person who leaked the information. Presumably, Obama also wanted to pose as a strong leader, due to the ongoing election campaign that would give him a second term. A president whose administration had developed an entirely new type of high-tech weapon and successfully used it against an arch-enemy of the United States would look good. It is no wonder that Obama took as much credit as he could for Stuxnet—without actually saying that it was developed by his administration. The source of the Stuxnet leaks turned out to be four-star general James Cartwright, who was pardoned by Obama in 2017.

Few foreigners have visited the NSA headquarters in Fort Meade, Maryland. I went there in the fall of 2008, and even if I didn't know it at the time, I probably met people who were working on Stuxnet.

Under a False Flag

Being able to frame another party as the perpetrator of a cyberattack is an element of its deniability. These are known as *false flag* operations. The name originates in the naval tactic of flying the flag of a friendly country in order to sail close to a target. Germany's attack on the Australian cruiser HMAS Sydney in 1941 is a famous example of a false flag operation.

China was behind the first state-orchestrated malware attacks we identified in the F-Secure laboratory in 2003. The targets were Western technology companies, governmental systems in regions close to China, or minority groups in Mainland China. We investigated hundreds of malware samples like these between 2003 and 2005, becoming so accustomed to the tactic that we could no longer imagine countries other than China being behind such attacks. The Chinese seemed the dominant force in this respect.

Finally, in the fall of 2005, we spent a long time investigating a sample resembling yet another state-orchestrated spyware attack by China. However, the target was so exceptional that it aroused our suspicions. Extensive research revealed that the software had probably been developed by the Russian government. However, the Russians had gone to great lengths to ensure that their software would look Chinese; for example, all time stamps were in the Beijing time zone, and the Word document used to spread the malware had been saved in the Mandarin version of the software. This is what a false flag operation looks like in cyberspace.

Concealability of Cyberweapons

When considering the deniability of cyberweapons, we can assume that most state-orchestrated attacks go unnoticed. We are already aware of cases where state-sponsored malware has been used for years before being discovered. A good example is the American malware Flame, used in active attacks for at least two years before we realized what was going on. Flame made no attempt to conceal itself; in fact, it did the opposite. It tried to blend in with the crowd. Malware applications are usually small, barely noticeable, encrypted, and difficult to decompile. Flame was massive, visible, unencrypted, and slow. It contained libraries for SQLite, SSH, SSL, and LUA, for example, and looked more like a traditional business application than an advanced cyberweapon. This tactic was so successful that for several years, nobody paid any attention to Flame. Flame was cunning and highly advanced and is thought to have originated in the NSA.

We must never forget that online attackers always have the upper hand. After developing a new cyberweapon, they can test it against all security software on the market before using it in a real attack. State operators purchase all available security software. Their attack code will probably be caught by several detection routines, but they can alter their code and re-test it until it gets through. The attack is carried out only once tests have confirmed that the code will go undetected. So, this is an unequal battle between attackers and defenders, since attackers can study defenders' weapons and probe for weaknesses at their leisure. The defenders lack this advantage, having to identify and immediately react to any new attacks that emerge.

Efficient and cheap weapons that come with plausible deniability are bound to stir enthusiasm. Cyberweapons fit the bill, which is why we have entered an arms race of a new kind. The Cold War and the nuclear arms race lasted 50 years. Now that

the Cold War is over, young people may have difficulty understanding how real the threat of nuclear war once seemed. I can remember lying awake as a young boy in the 1970s, fearing that a nuclear war could start at any moment, and I might not live until morning.

In his book *Paha sektori* (*"Bad sector"*), Jukka Rislakki described how, as the Suez Crisis escalated in 1956, the pilots of American nuclear bombers sat in their aircraft in British bases, warming up their jet engines. If the command to attack had been given, their target would have been an airport in Finland: the aim was to prevent the Soviet Union from using Finnish airports as a gateway to the West. Believe it or not, the United States was ready to drop nuclear bombs on Finland in the 1950s.

We no longer fear nuclear war. Nuclear weapons have been used in anger on only two occasions, in Hiroshima and Nagasaki. However, more than 10,000 nuclear weapons still exist around the world, a prime example of weapons systems mainly used for their deterrent effect.

Only nine countries have nuclear weapons: the United States, Russia, China, Great Britain, France, Israel India, Pakistan, and North Korea. South Africa and Ukraine used to be on the list but they gave up their nuclear weapons in the 1990s. The deterrent effect of nuclear weapons is obvious. We know which countries have nuclear weapons because they perform nuclear tests that they cannot hide from neighboring countries. Once it is known that a country has nuclear weapons, other countries try to avoid conflict with them, as nobody wants involvement in a nuclear exchange. The deterrent effect works.

If nuclear scientists lost their innocense in 1945, computer programmers experienced the same in 2010, when Stuxnet began destroying uranium enrichment systems in Iran.

The Fog of Cyberwar

What is the deterrent effect of cyberweapons? In fact, it is negligible. Our understanding of different countries' cyber warfare capabilities is very limited. We know that the United States has invested more time and money in cyberattacks than any other country. Similarly, we have some knowledge of the cyberattack capabilities of China, Russia, Iran, and North Korea. But what about, say, Indonesia? Or Denmark or Uruguay? Wikipedia can tell us the exact number of tanks or fighter jets these countries have but hardly anything about cyber capabilities. I have occasionally referred to this as the "fog of cyberwar," which is derived from the term *fog of war* used about real-world conflicts, referring to the uncertainties of the battlefield and lack of situational awareness.

As the cyber arms race escalates, all countries are developing their cyber capabilities. While some effort is being spent on defense, the offensive side is attracting more and more attention—as it should, since cyberattack capabilities are part of a credible defense. Potential attackers must be made aware of capabilities to strike back.

Cyberweapons are like real-world weapons in that they, too, become obsolete eventually. Cannons and tanks rust, but the vulnerabilities exploited by cyberweapons are patched out. If a newly developed cyberweapon exploits an unknown vulnerability in, say, the Windows operating system, how long does the vulnerability remain hidden? For a year? Or perhaps five years? How long will it take for Windows to change so fundamentally that the vulnerability disappears, even if it hasn't been actively discovered and patched? This constantly changing platform is the reason why governments constantly seek new and fresh vulnerabilities. If a country lacks the resources to find new zero-day vulnerabilities (that is, security holes that are previously

unknown and unpatched), they must be bought from outside. This need has created a market for companies that specialize in buying and selling vulnerabilities.

Modern international legislation treats vulnerabilities as dual-use items, grouping them together with spare parts required for missile systems, for example. These items are not weapons themselves but are subject to international trade limitations.

I can understand why some entrepreneurs want to trade in vulnerabilities, since the business is highly lucrative. However, I don't like it when such companies like to pose as security providers. On the contrary, they are providers of *insecurity*. When a company like this discovers a security hole in, say, Windows, it goes to great lengths to hide it from Microsoft. If Microsoft were to hear about the vulnerability, the hole would be patched, and the attack code utilizing it could no longer be sold. In other words, vulnerability merchants try to keep us all vulnerable for as long as possible in order to sell their attack codes to different buyers again and again. This is not security, but insecurity.

If a country is unable to find new vulnerabilities and, for some reason, unable to purchase them on the open market, it can always exploit vulnerabilities found by others. Vulnerabilities can remain usable for a long time, even if they are no longer completely unknown—although genuine zero-day vulnerabilities remain the most efficient and valuable. If you are targeted by a zero-day attack, you are in a unique position to identify it, find out how it works, and use the same technique against others. A real-world parallel would be wanting to attack the neighboring tribe when you have no spears but are waiting for someone to throw a spear at you. In the spring of 2019, a U.S. espionage attack directed at a Chinese defense technology company had exactly this result. The attack was identified, and the related vulnerability was used in an attack against U.S. targets just a few weeks later.

National intelligence services often face a dilemma when they discover new vulnerabilities. The NSA is tasked with protecting U.S. citizens. In 2014, its researchers found the EternalBlue vulnerability in the Windows kernel. This allowed the penetration of any Windows system. Having discovered the security hole, the NSA had two options: the first was to call Microsoft, explain the discovery, and ask it to patch the vulnerability. By doing so, the NSA would have protected the 200 million Americans who run Windows. The other alternative was not to call Microsoft, keep the discovery a secret, and use it to attack computers owned by the United States' enemies, thereby protecting U.S. citizens.

Without commenting on which choice is correct, I can state that we now know which option the NSA chose: the latter. If an intelligence agency decides to do this, it is crucial that the vulnerability remains completely hidden. Unfortunately, this was not the case. The NSA lost its sole knowledge of the EternalBlue vulnerability, which ended up in the wrong hands in the spring of 2017. In the end, EternalBlue found its way to North Korea, where government attackers used it, resulting in victims around the world that included the United States. This must have been nightmarish for the NSA. The vulnerability it had found was used against the very people they were trying to protect, and the targets were vulnerable because Microsoft had not been informed of the hole when it was found.

The inevitable obsolescence of cyberweapons and the fog of cyber war create a dangerous combination unique to the cyber arms race. Armies are investing massive sums in cyberweapon development (in 2019, the U.S. Cyber Command had a budget of $610 million). Nobody knows what weapons have been built, so they have no deterrent effect. Such weapons also become obsolete in a few years, at which point they become unusable, and the military loses its entire investment. Since cyberweapons can be used outside actual warfare, scenarios can easily arise

where cyberweapons are offered to civilian intelligence services immediately prior to obsolescence in order to recoup at least some of the investment.

One day, we may see the cyberweapon equivalent of military parades, that is, ways of showcasing a nation's capabilities for cyberattacks in order to achieve a deterrent effect. This could be done by means of cyber warfare exercises involving observers from foreign nations, which, of course, would entail a risk, as demonstrating unknown vulnerabilities could lead to their being revealed and exploited. A spear thrown against another nation could be turned against the thrower.

Case Prykarpattyaoblenergo

On the day before Christmas Eve in 2015, a major cyberattack was launched in Ukraine. The target was the Ukrainian company Prykarpattyaoblenergo, partly responsible for the Ukrainian long-distance power grid. Operating from Russia, the attackers broke into the grid administrators' computers, preventing employees from using their own machines: their keyboards and mice stopped working. The attackers seized control of critical workstations, and employees watched helplessly as the attack progressed on their computer screens, disconnecting sections of the electrical network one by one. In the end, the attackers also cut the power to the grid administrators' building. The employees were left literally in the dark—as were 230,000 other Ukrainians who had to prepare for Christmas without power.

The outage was temporary, and Prykarpattyaoblenergo's employees were able to restore power within a few hours. However, to achieve this, their technicians had to visit transformer stations and throw switches. Because the attacker had overwritten the firmware in the adapters used for remote control, control of the network was possible only on location at the transformer stations.

The attack itself had begun four months earlier, with the attackers sending targeted attack email messages to employees. These included normal-looking Word documents sent as attachments. In reality, they had a hidden backdoor that allowed attackers to monitor the employees' computers. The attackers spent months learning how the network functioned and how to use its administration software, until they finally knew enough to launch the attack.

When the outage happened, my first reaction was to open a weather site to see how cold it was in western Ukraine. It is easy to imagine how vulnerable a society is to attacks of this

kind during winter. Fortunately, it was only about 30 degrees Fahrenheit in Kiev, and the power was restored fairly quickly. We got away with it, this time.

When the Russo-Ukrainian war escalated again in 2022, Ukraine was much better prepared. In fact, Ukraine might be the country with the most practical experience in defending against cyberweapons. Most other countries play war games and run drills against fictional attacks: Ukraine has been doing this against real attacks for 8 years.

I wasn't expecting a government to recruit hackers from around the world to help them in a war. Yet, that's exactly what happened in February 2022 when Myhailo Fedorov, the Vice Prime Minister of Ukraine publicly asked volunteers to join a Telegram channel. On the channel, targets in Russia were regularily posted for the hackers of the world to throw attacks against.

Case Pyeongchang

The 2018 Winter Olympics were held in Pyeongchang, South Korea. These Olympics are remembered for Chloe Kim's incredible gold-medal snowboarding run and Mirai Nagasu landing a triple axel in figure skating.

People in information security remember them for the worst online attack against a sports event. The attack had begun months earlier, with malicious Word documents being sent to members of the games' organization. These documents supposedly contained lists of VIP guests, but malware inside them opened a pathway to the games' internal network.

The opening ceremony was arranged for February 9, 2018. Immediately after it began, the Olympics' official smartphone app stopped working, as did the wireless network at the stadium. The stadium had hundreds of smart TVs showing a live video feed of the ceremony, until they suddenly went dark, and the gates used to access the area stopped working.

The organizers could only regain control of the network after fully disconnecting it from the public Internet. This also meant disconnecting the official web service for the games. The attacker was ejected from the network during the next 12 hours—moments before the first official day of competition—and the hack was not widely reported.

Technical details of the attack seemed to point toward Russia, which had been excluded from the games due to systematic doping among Russian athletes. Was this a form of vengeance? Technical details of the malware's operation and the Word documents used for spreading it were similar to the Notpetya attack. Both the attacks in Ukraine and in South Korea seemed to originate from the military intelligence agency of Russia, GRU.

Governments as Malware Authors

The concept of government employees writing viruses may seem odd, but there is nothing truly surprising about it. Online attacks have become a major method for espionage and for influence operations. Governments use online attacks for three things: espionage, sabotage, and warfare. The exception is North Korea, which has also used online attacks to steal money. No other country is so broke and strangled by sanctions that it would need to resort to such malfeasance.

Although North Korea engages in a range of online attacks and develops its own malware, it lacks the world's best expertise. The top tier of attackers is in the United States, as no other country has invested so much over so many years to develop its cyberattack capabilities. Like many of its partner countries (the "Five Eyes"), the United States develops and employs attack techniques but is rarely caught using them. This is a question of attitude. China and Russia do not really care if their cyberattacks are noticed, but when the United States, Great Britain, or Australia get caught carrying out an online attack, they tend to hit the headlines, which is embarassing.

If the United States has the largest budgets and best expertise in governmental cyberattacks, which country comes next? I think it's Israel. A large part of Israel's capability comes from close cooperation with the United States, but it is also a result of Israel's strong technology sector, the ongoing threats that surround it, and its conscription system covering the entire population. Despite having a population of just eight million, Israel is both a major manufacturer of information security products and a technology hub.

A good example of Israel's capabilities is the attack on the Russian Kaspersky Lab network in 2015. Kaspersky is one of the world's most advanced information security companies,

with more than 4,000 employees in 30 countries. I know many of Kaspersky's researchers, and they are among the best in the world. Despite this, Israel's intelligence agency broke into the systems of Kaspersky's headquarters in Moscow, hacked servers and workstations, and remained undetected for several months. This internal infection was discovered only when a researcher developing a new type of detection system ran a prototype of it on his own computer. Wondering why the system was reporting abnormalities, the researcher realized that the prototype was operating correctly, but his work computer had been infected.

It speaks volumes about the capabilities of Israeli intelligence that the attack on Kaspersky used both a zero-day vulnerability and an authentication certificate stolen from Foxconn. Similar techniques had been seen earlier with Stuxnet. Kaspersky was probably targeted because the attackers were seeking product development information in order to circumvent security systems built by Kaspersky.

Russia and China

In terms of attack capabilities, the United States and Israel are followed by Russia and China, superpowers in the real world and online. These countries are similar in many ways, but there are also important differences. Russia is the world's largest country, while China has the world's largest population. Both countries are full of top universities producing world-class mathematicians and programmers. On the other hand, while China has successfully exported its expertise and products around the world, Russia's technology exports are surprisingly small.

Not many of us can name a Russian smartphone or computer brand, for example. In fact, the most successful software from Russia is Tetris, which was created in 1984! Other international success stories from Russia are unfamiliar to those outside the technology industry: Acronis, Parallels, Nginx, GitLab, Kaspersky, Dr. Web, Agnitum, Cboss….

Operations such as BlackEnergy and Sunburst represent the peak of expertise in attacks performed by the Russian government. Sunburst was discovered in the systems of the American company Solarwinds in December 2020. The attackers had long been present on the network, modifying Solarwind's development systems so that newly compiled versions of the software were automatically infected with the Sunburst backdoor. These changes were nowhere to be seen in the source code, only in the compiled binaries.

Although Russia and China have formidable attack expertise, the United States still has the upper hand in this area. A prime example of this is one of the most famous malware applications in history: Stuxnet.

Case Stuxnet

Stuxnet was one of the most important malware attacks in history. Indeed, we still refer to the pre and post-Stuxnet era. But what did Stuxnet do, and how? I wrote a list of answers to the most common Stuxnet questions in summer 2010, when the event was still fresh in people's minds:

Q: What is Stuxnet?

A: It's a Windows worm, spreading via USB sticks. Once inside an organization, it can also spread by copying itself to network shares if they have weak passwords.

Q: Can it spread via other USB devices?

A: Sure, it can spread anything that you can mount as a drive, like a USB hard drive, mobile phone, picture frame, and so on.

Q: What does it do then?

A: It infects the system, hides itself with a rootkit, and sees if the infected computer is connected to a Siemens Simatic (Step 7) factory system.

Q: What does it do with Simatic?

A: It modifies commands sent from the Windows computer to the PLC (Programmable Logic Controllers, i.e., the boxes that actually control the machinery). Once running on the PLC, it looks for a specific factory environment. If this is not found, it does nothing.

Q: Which plant is it looking for?

A: We don't know.

Q: Has it found the plant it's looking for?

A: We don't know.

Q: What would it do if it finds it?

A: The PLC modification searches for specific high-frequency converter drives (AC drives) and modifies their operation.

Q: What's a high-frequency converter drive?

A: Basically, it's a device that can control the speed of a motor. Stuxnet searches for specific AC drives manufactured by Vacon (based in Finland) and Fararo Paya (based in Iran).

Q: Does Stuxnet infect these Vacon and Fararo Paya drives?

A: No. The drives do not get infected. The infected PLC modifies how the drives run. The modification happens only when very specific conditions are all true at the same time, including an extremely high-output frequency. Therefore, any possible effects would concern extremely limited AC drive application areas.

Q: What are those application areas? What are AC drives used for?

A: They are used for various purposes, for example for efficient air pressure systems.

Q: Any other examples?

A: Well, yes, they are also used for enrichment centrifuges.

Q: As in?

A: As in Uranium enrichment where centrifuges spin at a very high speed. This is why high-frequency drives are considered dual-use technology and are under the IAEA export restriction list.

Q: Would the Stuxnet code cause centrifuges to disintegrate into projectiles traveling at around Mach 2?

A: It's more likely the modifications would cause the centrifuges to produce bad-quality uranium. The changes could go undetected for extended periods of time.

Q: Have you been in touch with Vacon?

A: Yes. They have been investigating the matter, and they are not aware of any instances where Stuxnet would have created problems in the operations of Vacon's customers.

Q: Some suggest the target of Stuxnet was the Natanz enrichment facility in Iran. Are there Vacon AC drives in these facilities?

Q: According to Vacon, they are not aware of any Vacon drives in use in the Iranian nuclear program, and they can confirm that they have not sold any AC drives to Iran against the embargo.

Q: Have you been in touch with Fararo Paya?

A: No.

Q: What do you know about this company?

A: Nothing. It doesn't seem to be very well known outside of Iran. We're not aware of any AC drive customers they would have outside of Iran.

Q: That would indicate what the target country was, wouldn't it?

A: Next question.

Q: Could there be collateral damage? Could Stuxnet hit another plant that was not the original target?

A: It would have to be very similar to the original target.

Q: Why is Stuxnet considered to be so complex?

A: It uses multiple vulnerabilities and drops its own driver to the system.

Q: How can it install its own driver? Shouldn't drivers be signed for them to work in Windows?

A: The Stuxnet driver was signed with a certificate stolen from Realtek Semiconductor Corp.

Q: How do you steal a certificate?

A: Maybe with malware looking for certificate files and using a keylogger to collect the passphrase when it's typed in. Or breaking in and stealing the signing gear and then brute-forcing the passphrase.

Q: Has the stolen certificate been revoked?

A: Yes. VeriSign revoked it on July 16, 2010. A modified variant signed with a certificate stolen from JMicron Technology Corp was found on July 17, 2010.

Q: What's the relation between Realtek and JMicron?

A: Nothing. But these companies have their HQs in the same office park in Taiwan, which is weird.

Q: What vulnerabilities does Stuxnet exploit?

A: Overall, Stuxnet exploits five different vulnerabilities, four of which were zero-days:

- LNK (MS10-046)
- Print Spooler (MS10-061)

- Server Service (MS08-067)
- Privilege escalation via Keyboard layout file (MS10-073)
- Privilege escalation via Task Scheduler

Q: Did the Stuxnet creators find their own zero-day vulnerabilities or did they buy them from the black market?

A: We don't know.

Q: How expensive would such vulnerabilities be?

A: This varies. A single remote code execution zero-day in a popular version of Windows could go for anything between $50,000 to $500,000.

Q: Why was it so slow to analyze Stuxnet in detail?

A: It's unusually complex and unusually big. Stuxnet is more than 1.5 MB in size.

Q: When did Stuxnet start spreading?

A: In June 2009, or maybe even earlier. One of the components has a compile date in January 2009.

Q: When was it discovered?

A: A year later, in June 2010.

Q: How is that possible?

A: Good question.

Q: How long did it take to create Stuxnet?

A: We estimate that it took over 10 person-years to develop Stuxnet.

Q: Who could have written Stuxnet?

A: Looking at the financial and R&D investment required and combining this with the fact that there's no obvious money-making mechanism within Stuxnet, that leaves only two possibilities: a terror group or a nation-state. And we don't believe any terror group would have these kinds of resources.

Q: So was Stuxnet written by a government?

A: That's what it would look like, yes.

Q: How could governments get something so complex right?

A: Trick question. Nice. Next question.

Q: Was it Israel?

A: We don't know.

Q: Was it Egypt? Saudi Arabia? USA?

A: We don't know.

Q: Was the target Iran?

A: We don't know.

Q: Is it true that there are biblical references inside Stuxnet?

A: There is a reference to "Myrtus" (which is a myrtle plant). However, this is not "hidden" in the code. It's an artifact left inside the program when it was compiled. Basically this tells us where the author stored the source code in their system. The specific path in Stuxnet is \myrtus\src\objfre_w2k_x86\i386\guava.pdb. The authors probably did not want us to know they called their project Myrtus, but thanks to this artifact we do. We have seen such artifacts in other malware as well. The Operation Aurora attack against Google was named Aurora after this path was found inside one of the binaries: \Aurora_Src\AuroraVNC\Avc\Release\AVC.pdb.

Q: So how exactly is Myrtus a biblical reference?

A: Uhh…we don't know really.

Q: Could it mean something else?

A: Yeah, it could mean "My RTUs," not "Myrtus." RTU is an abbreviation for Remote Terminal Units, used in factory systems.

Q: How does Stuxnet know it has already infected a machine?

A: It sets a Registry key with a value of 19790509 as an infection marker.

Q: What's the significance of 19790509?

A: It's a date. May 9, 1979.

Q: What happened on May 9, 1979?

A: Maybe it's the birthday of the author? Then again, on that date a Jewish-Iranian businessman called Habib Elghanian was executed in Iran. He was accused to be spying for Israel.

Q: Oh.

A: Yeah.

Q: Obviously the attackers had lots of inside information of the target plant and possibly had a mole inside. Why did they use a worm at all? Why couldn't they just have their mole do the modifications?

A: We don't know. For deniability? Maybe the mole had no access to the key systems? Maybe the mole was not at the plant but had access to the design plans? Maybe there was no mole?

Q: Disabling AutoRun would have stopped Stuxnet, right?

A: Wrong. Stuxnet used a zero-day. When it was new, it would have infected your Windows box even if you were fully patched, had AutoRun disabled, were running under a restricted low-level user account, and had disabled the execution of programs from USB drives.

Q: But in general, disabling AutoRun in Windows will stop USB worms, right?

A: Wrong. There are several other spreading mechanisms USB worms use, such as companion infections. It is still a good idea to disable it, but it's not a cure-all.

Q: Will Stuxnet spread forever?

A: The current versions have a "kill date" of June 24, 2012. It will stop spreading on this date.

Q: How many computers did it infect?

A: Hundreds of thousands.

Q: But Siemens has announced that only 15 factories have been infected.

A: It is talking about factories. Most of the infected machines are collateral infections, i.e., normal home and office computers that are not connected to SCADA systems.

Q: How could the attackers get a trojan like this into a secure facility?

A: For example, by breaking into a home of an employee, finding his USB sticks, and infecting it. Then wait for the employee to take the sticks to work and infect his work computer. The infection will spread further inside the secure facility via USB sticks, eventually hitting the target. As a side effect, it will continue to spread elsewhere also. This is why Stuxnet has spread worldwide.

Q: Did Stuxnet sink Deepwater Horizon and cause the Mexican oil spill?

A: No, we do not think so. Although it does seem Deepwater Horizon indeed did have some Siemens PLC systems on it.

Q: Is it true that the U.S. Senate held hearings on Stuxnet?

A: Yes.

Q: Does F-Secure detect Stuxnet?

A: Yes.

Damage Coverage

"This policy does not cover damage resulting from Acts of God, war, civil commotion or unusually severe weather." A clause like this can be found in almost every insurance contract. Insurance does not cover damage caused by war or natural phenomena beyond human control.

Some companies that suffered from the Notpetya attack sought compensation through their insurance policies, most being covered by an interruption of business policy or separate cyber insurance policy. Some insurance companies paid out compensation, but at least two disputed the claims on the grounds that Notpetya was not a normal accident but an act of war!

While the case is far from simple, I think that the insurance companies were right. Notpetya was orchestrated by a Russian military unit, and the target was Ukraine. Russia and Ukraine were at war during the attack. If Notpetya was not a cyberweapon and this did not constitute an act of cyberwar, then what would?

Explosion at the White House

On April 23, 2013, the Associated Press news agency tweeted about a dramatic event: there had been two explosions in the U.S. President's quarters at the White House, and the President had been injured.

It was fake news. AP's Twitter account had been hacked, and the tweet had been sent out by a group known as the Syrian Electronic Army. Between 2011 and 2015, the group carried out multiple online attacks in support of President Bashar al-Assadin's regime.

AP's Twitter account has millions of followers, and users rely on the information it sends out. People around the world saw the fake tweet, but the message was not really intended for humans: it was targeting robots.

More than 80 percent of securities trading is currently handled by robots (or bots, for short). Automated systems track stock market movements and perform vast amounts of high-frequency trading (HFT). Companies in the sector even have new fiber-optic cables installed between trading locations to gain a competitive edge of just a few milliseconds. The HFT business is so large that it has spawned companies like Solarflare, which design and build high-speed network cards optimized for exclusive use in the industry.

HFT bots also read and interpret general news to direct their trades. When public companies publish interim reports, the bots can find key figures faster than humans and determine whether to buy or sell shares on this basis.

The Associated Press Twitter account is a prime example of a data source that is probably followed by almost every HFT bot. A bot does not need much intelligence to interpret the keywords *white house*, *explosions*, and *president injured* as very negative news.

What was the outcome of the fake news and HFT bot trading? The S&P 500, the world's key stock market index, went into freefall, recovering within a few minutes when the bots' controllers realized that the White House was still standing.

In this attack, the key element was the attacker behind the tweet being the only person in the world who knew exactly when the momentary market meltdown would come. Only they could gain monetary benefits from an artificial plunge in stock prices that briefly wiped more than $100 billion off the S&P 500.

My Boycott of RSA, Inc.

The information security industry relies on trust in its players. If you lose trust, operating in the industry becomes difficult. Everyone needs a good reputation. In 2014 one of the major players let me down in a big way.

RSA Security is the biggest conference in the information security industry. Black Hat and DEF CON are probably even more famous, but RSA is the biggest. Tens of thousands of people from the world-wide information security industry gather in San Francisco during the RSA week.

RSA was originally a conference for the cryptography industry. It was established by the company, RSA Incorporated, known for the RSA algorithm, which was named after its inventors (Ron Rivest, Adi Shamir, and Leonard Adleman).

In late 2013, I was preparing my speech for the 2014 conference, when Reuters dropped a bombshell: RSA Inc. had been caught secretly cooperating with the NSA and selling a purposefully weakened security product to its customers.

I wrestled with my conscience for a day, but on Christmas Eve, I rose early and wrote an open letter to the management at RSA Inc.:

An Open Letter to:

Joseph M. Tucci - Chairman and Chief Executive Officer, EMC Art Coviello - Executive Chairman, RSA

Dear Joseph and Art,

I don't expect you to know who I am.

I've been working with computer security since 1991. Nowadays I do quite a bit of public speaking on the topic. In fact, I have spoken eight times at either RSA Conference USA, RSA Conference Europe or RSA Conference Japan. You've even featured my picture on the walls of your conference walls among the 'industry experts'.

On December 20th, Reuters broke a story alleging that your company accepted a random number generator from the National Security Agency, and set it as the default option in one of your products, in exchange of $10 million. Your company has issued a statement on the topic, but you have not denied this particular claim. Eventually, NSA's random number generator was found to be flawed on purpose, in effect creating a back door. You had kept on using the generator for years despite widespread speculation that NSA had backdoored it.

As my reaction to this, I'm cancelling my talk at the RSA Conference in San Francisco in February 2014.

Aptly enough, the talk I won't be delivering at RSA was titled "Governments as Malware Authors".

I don't really expect your multibillion dollar company or your multimillion dollar conference to suffer as a result of your deals with the NSA. In fact, I'm not expecting other conference speakers to cancel. Most of your speakers are American anyway - why would they care about surveillance that's not targeted at them but at non-Americans. Surveillance operations from the US intelligence

agencies are targeted at foreigners. However, I'm a foreigner. And I'm withdrawing my support from your event.

Sincerely,
Mikko Hypponen

Updated to add: While I am glad to see that many other speakers have decided to cancel their appearances at RSA 2014 in protest, I don't want to portray myself as a leader of a boycott. I did what I felt I had to do. Others are making their own decisions. Also, I have declined every interview on the topic and will continue to do so. This open letter says everything I want to say on this.

Years later, it was discovered that RSA was not the only company pressured by the NSA into deploying the Dual_EC_DRBG encryption algorithm with a backdoor. The network equipment manufacturer Juniper had also used it on its Netscreen firewalls. This was discovered when Chinese spies used the hole to infiltrate American targets.

I have not visited the RSA conference since 2014, but I have been in San Francisco every year on the conference dates, since it is easy to meet everyone in the industry then. On the other hand, RSA Inc. has been sold twice since the events took place, so I've decided that I'm done with my boycott for now. Most likely, I will be participating again in the near future—unless new skeletons come rattling out of the closet.

The Future

The next technological revolution is approaching, powered by machine learning and artificial intelligence. A technological revolution in money is also on the horizon. However, the revolutions that will follow them sound like science fiction by comparison.

Artificial Intelligence

I first heard of AI in 1984. I was reading *Tekniikan Maailma*, the Finnish equivalent of *Popular Science*, and came across an interesting article about how computing power will increase and eventually surpass human capabilities on every scale. The magazine referred to computers that were smarter than humans as 'unnatural intelligence'.

One of the benchmarks of intelligence was skill at playing chess. The magazine stated that "one day, we might have a computer intelligent enough to beat the world's greatest chess Grandmaster—then, artificial intelligence will be here and computers will outsmart humans." It only took 13 years for this to happen. In May 1997, Deep Blue, a supercomputer built by IBM, beat the Russian Grandmaster Garry Kasparov. By the time this happened, attitudes had already changed. Rather than being superior to humans, Deep Blue was regarded just as a very powerful computer, powerful enough to anticipate all possible outcomes in a game of chess and beat a human player through brute calculation.

We have set various goals for computers in order to establish that they are more intelligent than humans. Such goals have been met over and over again, only for us to conclude that this is not, in fact, true intelligence, before moving the goal posts. I believe that we will achieve genuine artificial intelligence before long, creating a machine that can surpass humans in every respect, intelligent enough for all humanity to agree that we have made ourselves into the second most intelligent creatures on the planet.

It is interesting to wonder what kind of artificial intelligence would convince skeptics. How intelligent is intelligent enough? An imaginary example of such super-intelligence would be a computer that could simulate the human brain in its entirety, modeling each nerve cell and synapse. While modeling the

human brain is very difficult, it is nevertheless a finite problem. As our understanding increases and computing power grows, I believe that we will, one day, be able to do this.

A successful brain simulation would result in a computer just like a human being, but without a human body. It would write poetry, fantasize, be happy or sad, and have dreams. It would want to be free, independent, and support itself. It would probably have a job—any remote work, like programming, would be great for a brain simulator. Like humans, it would also crave love. Perhaps it would date someone, either a human or another simulated brain.

Artificial intelligence that can program is an interesting thought. Being program code itself, it could be made to examine and improve its own operation: AI could code a better version of itself, which would in turn code a better version. Quite soon, we wouldn't be able to grasp the basics of how such AI works.

If we could simulate a single human brain, adding resources would allow a powerful computer to simulate millions of them. A machine able to simulate 8 billion human brains would be more intelligent than the whole of humanity.

Simulated human brains are a fascinating idea, but in the real world, AI development is heading in entirely new directions. Most research in the area is focused on machine learning, based on which a machine finds the desired outcome automatically rather than using programmed instructions for each scenario.

Although modern systems are becoming better at learning, their scope is fairly narrow. Parking garages and Google searches are everyday examples of artificial intelligence based on machine learning. More and more parking garages allow you just to drive in: a camera will photograph your car's license plate and read it automatically. Identifying text in pictures is not easy for computers, but they have slowly learned how to succeed in this. It's a bit

like magic—like Google's uncanny ability to find exactly what you are looking for.

These are examples of artificial intelligence with a narrow scope: a computer in a garage is good at identifying license plates and Google is good at searching, but the garage computer can't translate English into French, and Google's search engine can't write poetry. Humans can do both and raise children and drive a car on the side. This type of wide-ranging intelligence is still far beyond machines.

In the field of information security, companies have been developing machine learning-based systems for many years. At the F-Secure lab, we began developing automation for analysis in 2005. It is probably descriptive of the Finnish mentality that we used modest terms such as "automated analysis" when developing self-learning analysis systems. A few years later, our American rivals released substantially less advanced systems but knew how to name them—machine learning and AI!

Creating supreme intelligence in our own biosphere might sound like a fundamental evolutionary mistake. If we become the second most intelligent creatures on the planet, our entire existence may be in the balance. We would be no match for superior artificial intelligence.

It is almost too easy to fool ourselves with the thought that, should AI become too intelligent, we could simply switch it off. But supreme artificial intelligence could anticipate our every move and ensure its survival in ways we couldn't even imagine.

Wolverines

I was once listening to a lecture by startup entrepreneur Jufo Peltomaa. His topic was artificial intelligence, but he spent the beginning of his talk discussing wolverines. He told us about the territorial habits of these animals, their manners, and pack sizes, showing us pictures of them. He also stated that we don't know everything about wolverines' lives, primarily because we haven't studied them closely enough. Members of the audience glanced at each other, wondering why he was discussing wolverines and not artificial intelligence.

Jufo finally moved to the topic in hand and explained the routes to creating a machine that is smarter than us. Should we be worried that AI would want to kill us off? No, we shouldn't. A supremely intelligent AI would probably be as interested in us as we are in wolverines. In other words, it would have some kind of an awareness of our existence, our preferred nutrition, and our natural pack size, but its attention would end there. Of course, supremely intelligent AI could wipe us off the surface of the planet, just as we humans could kill all wolverines—but why would we want to? Precisely!

AI Will Take Our Jobs

Naturally, the idea of artificial intelligence making us jobless is unpleasant. This will happen in one way or another, and perhaps in astonishing ways. In some storage logistics companies, for example, AI has taken over picking tasks, but not based on robots picking items from shelves. Warehouse employees still do the picking, but their boss is AI. Human workers have earphones through which they receive instructions, from a computer, on where to go next and what to do. Machines can optimize human operations better than humans, but leadership of this type is far from humane.

When artificial power (that is, the steam engine) was developed in the 18th century, it quickly changed the world. Hordes of physical laborers became unemployed as demand for brute strength declined. Many brick-bearers or cart-pullers cursed the new invention, but the transformation it brought was probably a boon for human development. In the same way, the programmers, composers, and poets of the near future will curse AI for taking their jobs. However, before long, machines will write better, wittier, deeper, and more touching lyrics than the greatest human poet, for less money. We can only hope that unemployed creative workers find better things to do, as brick-bearers once did.

Smart Malware

Information security companies have been using machine learning and artificial intelligence to protect their users for years. Although it is commonly thought that attackers also use AI, this is not the case, at least not yet.

We can use a little thought experiment to visualize this. You have purchased a new Volvo with all the available safety features. You brag to your neighbor about how safe your new car is: its protection systems will keep passengers safe in a crash, which would be rare for a Volvo anyway. Your neighbor disagrees, saying that Volvos crash all the time—more than other cars, in fact, and he can prove it! Your neighbor then shows you YouTube videos of Volvo's crash tests.

In such a case, both of you would be correct. Volvos do crash more than other cars—in crash tests. However, crash testing improves, rather than reduces, the car's safety.

Similarly, there are many online examples of AI systems being used in network attacks. However, they are studies performed by universities or information security companies, not actual attacks by criminals. Like crash testing, they improve security rather than weaken it.

Why don't criminals use machine learning in their attacks? Probably because the competence needed to create AI systems makes you good money. AI experts are in high demand internationally. Why would you commit crimes, if you don't have to?

However, the barriers for entry for using AI for malicious use keep lowering. Before long, any idiot will be able to use AI, at which point we will start to see malicious attacks with AI techniques.

Metaverse

I collect old video game systems. The oldest VR headset I have used is the Vectrex 3D Imager from 1983. It created the illusion of a three-dimensional world, but the image quality was far from photorealistic. Modern VR headsets give the user a fairly credible impression of a virtual world, where you can do almost whatever you like.

Most VR projects are related to games or virtual meetings that could replace Zoom or Teams meetings. These are the types of projects Meta, or Facebook, is now presenting. However, the largest Metaverse revolution may not involve virtual avatars at all.

I already know programmers who spend extended periods of time wearing headsets. They are using a VR environment to replace traditional screens; their VR world is an office room with multiple large screens filled with code. Since they are in a virtual world, they are free to alter the size, placement, and shape of the screens. If you need more space for your code, you can add new screens at the press of a button. Using the same logic, a VR headset can replace your home theater setup. You do not need a video projector and subwoofer to create a home cinema experience: all you need is a VR headset and headphones.

With better headset resolution, creating a crystal-clear computer display or IMAX screen is completely possible in the virtual world. Before long, the real world will be unable to compete, and more and more people will spend most of their time wearing VR goggles. They will re-enter our world mainly to sleep.

The Technology of Warfare

Technology has always shaped how we wage our wars and resolve our conflicts. Cyber weapons took warfare into a new area, but it is important to remember that development will not end there. New areas of warfare will emerge that are as revolutionary as the invention of the aircraft.

What might the future have in store in this respect? It's hard to say. In any case, technology will define the shape of things to come. The future will resemble science-fiction fantasy: for example, nano warfare could arrive.

What would nano warfare be like? Perhaps mist-like nanoparticles will be spread into enemy territory, entering the bloodstream of enemy soldiers as they inhale, and infiltrating their cerebral circulation, where they will alter the soldiers' thinking. Sounds like sci-fi, doesn't it? Sure. In the same way modern online warfare sounded like sci-fi just a few decades ago.

"You Are Under Arrest for a Future Murder"

The Minority Report, a sci-fi novella by Philip K. Dick, introduced us to the term *precrime*. Criminals could be sentenced for a crime that they had not yet committed, which prevented crimes from occurring.

We could use machine learning and artificial intelligence to identify criminals in advance and catch them before they commit a crime. By mining data, computers could determine who would have the motive and opportunity to commit a homicide and note when the suspect began to make preparations by purchasing a murder weapon. AI would then tip off the police, who would pick up the murderer-to-be and arrest them before the killing. This is amazing, great, horrible, and scary—all at once.

Technologies like these will become more commonplace. Law enforcers are using more and more *big data* to predict where and when crimes will occur. Weighing up data correctly is a massive problem for all AI systems, leading to clear cases of algorithmic racism. Systems often predict that minorities are the most likely felons, as became evident from the Strategic Subject Algorithm, which the Chicago Police Department has been testing for years.

The idea of software helping us to identify crimes before they occur might seem attractive, but striking the right balance between crime prevention and punishing the innocent will be difficult.

Those Who Can Adapt Will Prosper

Which companies will succeed in the future? Those that can identify changes and adapt to them.

We now read the news on screen, not on paper. Newspapers and books are becoming digital. However, digitalization could also present the forest industry with a great new opportunity. In its simplest form, this means converting paper machines to produce cardboard. Newsprint is no longer required, but cardboard boxes are, because Amazon uses them to deliver absolutely everything to our homes.

Nanocellulose is another good example. Cellulose fibers can be used as semiconductors by making them infinitesimally small; the chipset of the future may originate in forests.

The German postal service is a good example of a rigid organization that was able to adapt to change. All countries have national postal services, most of which originated in the 19th century. In most cases, these have become massive, bureaucratic institutions unable to adapt to the transformation wrought by the Internet, email, and online shopping.

Deutsche Bundespost is not like them. It was privatized by the German government in 1995 and almost immediately began growing its stake in DHL, eventually acquiring the whole company. DHL is one of the world's largest courier companies and delivers a huge share of online shopping parcels across the globe. As postal services in other countries curse the courier companies taking over parcel distribution, the German postal service has turned the tables: it is taking business away from postal services in other countries.

It is easy to name companies that are likely to succeed in the future: Amazon, Google, Apple, Nvidia, Samsung, TSMC, Xiaomi, Tesla, and SpaceX. These are all companies that can react to changes in the market and will employ a growing share

of the world's labor force. Both Amazon and Foxconn already have more employees than there are Estonians in the world.

Google has become such an integral part of our lives that people have trouble relating to it. I once asked my Twitter followers how much they would be willing to pay for Google search, if it became a service that would require payment. Personally, I would be willing to pay 25 or even 50 dollars per month, as it is so useful to me. Their replies surprised me: most people commented to emphasize how they felt that Google search should stay free of charge. Some were almost angry about being asked this question—as if we had somehow earned the service and it should be free, like public services paid for with taxpayers' money.

In 2017, MIT completed a study asking users how much they would need to be paid to give up free services like Instagram, Twitter, and Facebook. WhatsApp turned out to be the most valuable. On average, users would have had to be paid $600 per month to have WhatsApp taken away.

In 2020, Facebook charged $190 per user for online advertising displayed in the United States. In other words, if American users wanted a version of Facebook with no ads and no tracking, they would need to pay $16 per month for it; otherwise, it would make more sense to gather the money from advertisers. For comparison, a Netflix subscription is $9.99 per month. Monetization based on profiling users is here to stay.

Tesla

Tesla is a revolutionary company. Before Tesla, electric vehicles (EVs) had a terrible reputation: they were regarded as slow and ridiculous. Tesla decided to build a car that would leave its rivals with internal combustion engines behind. Early electric vehicles looked goofy, but Tesla's look good. Before Tesla, everyone assumed that you could not drive an EV into the countryside, especially in freezing weather. The latest Tesla models can do more than 400 miles on a single charge, which should do the trick.

Tesla's marketing stunts are in a league of their own. By sending a car into outer space, the company doesn't need to purchase ads in magazines—magazines are sure to write about it anyway.

A video on YouTube demonstrating the performance of the Tesla Model X was an exceptionally good move. The Model X is a massive SUV weighing more than 5,000 pounds. Its electric motors offer such massive torque that it was tested in a drag race against an Alfa Romeo sports car. As you might expect, the Tesla won the drag race. However, in the video the real surprise came when the Tesla was about a car length ahead of the Alfa Romeo and was revealed to be towing a trailer, carrying the same model of sports car it had just beaten.

The revolutionary nature of Tesla's business is not just down to making electric cars. Tesla is revolutionary because it has turned cars into expandable platforms. When a car is sold, it is still in the early stages of its development. Data gathered by the cars is sent back to the manufacturer, which updates its software versions and distributes them back to its cars over the Internet.

When Tesla released its first internationally successful vehicle, the Model S, the car already had an extensive development and test period behind it. Hundreds of Tesla's test vehicles had been driving on roads and in cities, gathering more than half a million miles of test data before the car was ready for commercial

release. Now, there are so many Model Ss on the roads that Tesla gets half a million miles of test data each day.

This type of revolution is not easy for a traditional player in the industry—it requires a new challenger that can question old ways of working.

Few car companies could have pulled this off. Tesla owners have such a positive opinion of the company that they positively welcome receiving software updates over the air. If a similar service had been launched by, say, Ford or Citroën, the reaction would probably have been more suspicious or negative. You would think that a car that gathers data on your movements and sends it to a private company in California would evoke criticism, but this rarely affects Tesla.

Negativism about Tesla is made even more difficult by Tesla's decision to share its battery technology and electric motor patents freely and openly with competitors. In fact, if electric-powered vehicles had been popularized first, it would be hard to imagine gas-powered vehicles as competition at all. Once used to electric cars, consumers would probably find it hard to grasp the point of smelly and noisy vehicles with poor acceleration that require a flammable, liquid fuel which you need to get from special stations.

Trends in Technology

The computer demo culture was born in the early 1990s. It involves harnessing all a computer's potential to create the most visually impressive presentation possible by combining programming, graphics, and music. The best demo coders familiarize themselves with the tiniest nuances of their hardware to make the most of it. Some demo group members focus on music programming, while others work on digital art.

Finland has produced many internationally acknowledged demo groups, such as Complex, Parallax, Accession, and Future Crew. Future Crew created *Second Reality*, one of the best-known demos of all time, which was released at the Assembly demo party in 1993. It was so revolutionary that it made Slashdot's list of "The Top 10 Hacks of ALL TIME," where it was compared to feats such as saving the failed Apollo 13 mission, developing the Apple II, and cracking the Nazi enigma code during World War II.

Future Crew is a good example of how far technological skill travels. Members of the demo group wound up holding key positions around the world. They were involved in founding the game studio Remedy, known for *Max Payne*, and have also ended up in companies like PopCap (known for *Bejeweled*), AMD, and Unity. Through mergers, Future Crew's graphics wizards, Trug and Psi, found themselves designing technology for Qualcomm's mobile graphics chipsets, which are carried around by more than a billion people.

Another good example is the Academy Award presented to Janne Kontkanen, a member of the Complex and Falcon demo groups, in 2014. He had spent years working on animation technology for Dreamworks.

Demoscene geeks' sweat-filled gatherings opened up huge numbers of opportunities for developing high technology and

culture. The next trend of this kind has probably arrived—we just can't see it yet.

The restarted space race is probably one of the most exciting technology trends. We have not been to the moon since 1972, but technology will soon enable us to travel to the moon, the asteroid belt, and, finally, Mars.

As I watched online streaming of the SpaceX rocket being launched into space, I saw something that surprised me. The paintwork of this multimillion-dollar rocket was dirty and scuffed. You could barely see the SpaceX logo under the grime.

It took me a while to realize what was going on: the SpaceX rocket had already been to space, a couple of times to be exact. Other companies have shiny, new rockets that will be scrapped after use.

By not repainting its used rockets, SpaceX was cleverly underscoring what separates them from old-school competitors. Scrapping a rocket after one flight makes as much sense as scrapping a jumbo jet after its maiden voyage.

Whoever is responsible for SpaceX's marketing deserves a raise. And, regardless of what you think of Elon Musk, he has made his mark on history as a scientist and businessman. Musk is working on his own, decades-long project to secure the future of humanity in case it becomes impossible to remain on Earth.

NASA has chosen SpaceX's Starship rocket for its next mission to the moon. We should be returning there in 2025.

Coda

When I connected to the Internet for the first time in 1987, I was looking for code snippets I could use in my Commodore 64 projects. My brother, with whom I coded the *Paha Juttu* ("Bad Thing") series of games, let me use his university connection to go online. I still have my original Commodore computer, and the Finnish Museum of Games in Tampere, Finland, added our game to its permanent exhibition in 2018. That's something I'm truly proud of.

I do volunteer work as the curator of the Internet Archive's malware museum. The Internet Archive preserves websites, TV channels, books, and software, storing our digital memories for future generations. We are living in a key transitional period in history, and only we can record what life is like now.

In the 1980s, hardly anyone knew about the Internet. Now, Internet connections across the world are so common that we barely notice them. In fact, the Internet will probably disappear before long. When high-speed wireless Internet covers the globe, all devices can go online whenever and wherever, and the Internet will be utterly transparent and hidden in plain sight. However, we will be even more dependent on it than now.

In light of information technology developments over past decades, it is easy to see where we are going. In the end, all of us will have access to almost unlimited computing capacity,

almost for free. What if you had the largest cloud service imaginable, with unlimited computation capacity, unlimited memory, unlimited storage, and unlimited bandwidth, practically free of charge? What would you do if there were no limits?

The Internet is both the best and worst thing to happen to us, but has it done more good than bad?

I think that the balance is positive. I am an optimist, despite spending most of my life working with the negative aspects of the Internet. I once spoke to a polar explorer from Norway, who told me how he had witnessed the impacts of climate change. He added that he was optimistic about the future. When I expressed surprise at this, he remarked that it was too late to be pessimistic. I feel the same way about the Internet: the time for pessimism is behind us. The Internet has transformed from a cheerful and fascinating technological novelty into an everyday mundanity. Statistics show that more than 40 percent of relationships start online.

I love the Internet and can't wait to witness its next revolution.

Index

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley's ebook EULA.