

**THE  
GORILLA  
GUIDE TO...** ®



# Enterprise Security Fundamentals

Trevor Pott

---

## INSIDE THE GUIDE:

- Consider the human side of IT security
- Explore modern security technologies
- Examine the role of business processes in IT security

**HELPING YOU NAVIGATE  
THE TECHNOLOGY JUNGLE!**



**ActualTech Media**

[www.actualtechmedia.com](http://www.actualtechmedia.com)

**THE GORILLA GUIDE TO...**

# Enterprise Security Fundamentals

## **AUTHOR**

Trevor Pott, eGeek Consulting

## **EDITOR**

Keith Ward, ActualTech Media

## **LAYOUT AND DESIGN**

Olivia Thomson, ActualTech Media

Copyright © 2018 by ActualTech Media

All rights reserved. This book or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the publisher, except for the use of brief quotations in a book review.

Printed in the United States of America.

## **ACTUALTECH MEDIA**

Okatie Village Ste 103-157

Bluffton, SC 29909

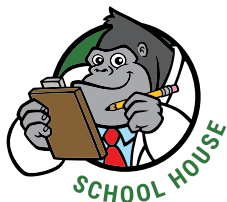
[www.actualtechmedia.com](http://www.actualtechmedia.com)

# ENTERING THE JUNGLE

<b>Introduction</b>	<b>7</b>
Human Considerations	7
Don't Panic	9
Network Effects	11
<b>Chapter 1: Basic Principles</b>	<b>12</b>
Threats	12
Human Nature	13
Physical Security	15
Unified Authentication	17
Principle of Least Privilege	19
Defense in Depth	20
Protocol and Standards Creep	22
<b>Chapter 2: Network Security</b>	<b>24</b>
Networking 101	26
Edge Security and Introspection	28
Firewalls	29
Microsegmentation	30
Port-Based Network Management	32
<b>Chapter 3: Endpoint Security</b>	<b>33</b>
Best Effort	34
Presumption of Compromise	35
Endpoint Management Solutions	36
Patch Management	37

Anti-Malware.....	38
Host-Based Intrusion Detection (HID).....	39
Infrastructure as Code.....	41
Separate Management VM.....	43
<b>Chapter 4: Process, Auditing and Compliance.....</b>	<b>45</b>
Versioning of Configurations.....	46
Security Information and Event Management (SIEM).....	47
Encryption.....	48
Data Protection.....	49
Asset Detection and Management.....	51
Vulnerability Scanning .....	51
Support Calendars.....	52
No Plan Is Perfect.....	52
<b>About the Author.....</b>	<b>54</b>

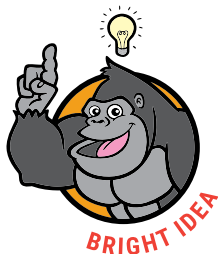
# CALLOUTS USED IN THIS BOOK



The Gorilla is the professorial sort that enjoys helping people learn. In the School House callout, you'll gain insight into topics that may be outside the main subject but are still important.



This is a special place where you can learn a bit more about ancillary topics presented in the book.



When we have a great thought, we express them through a series of grunts in the Bright Idea section.



Takes you into the deep, dark depths of a particular topic.



Discusses items of strategic interest to business leaders.

# ICONS USED IN THIS BOOK



## **DEFINITION**

Defines a word, phrase, or concept.



## **KNOWLEDGE CHECK**

Tests your knowledge of what you've read.



## **PAY ATTENTION**

We want to make sure you see this!



## **GPS**

We'll help you navigate your knowledge to the right place.



## **WATCH OUT!**

Make sure you read this so you don't make a critical error!

# INTRODUCTION

In IT, there is an exception to every rule; even the most basic rule of IT, which states that the answer to every question is “it depends.” Security is no different, and because it touches on all aspects of IT from the physical out to the cloud, anything to do with security is going to be rife with exceptions, including this book. There is a lesson in the middle of this confusion, however, which is that all good IT security implementations have to start somewhere.

Even if you stop reading this book after this paragraph, I hope that you take away from my efforts the idea that IT security starts with doing something—anything—to address the massive technical and business process debt that threatens to overwhelm us all. It doesn’t matter if you’re a one-man band or Amazon itself, every one of us has flaws in our IT somewhere, and we cannot allow this reality to overwhelm us.

## Human Considerations

Before we begin to address any of the technical details of IT security, I want to take a moment to focus on the people part of security. Here, I do not mean to engage in a rant about end users or managers. Nor do I intend to get into a DevOps-like philosophical discussion about “people, process and products.”

When I talk about the human considerations of IT security, I’m talking about the toll that IT security takes on you, the person reading this book. If you have this book in front of you, you’re probably an IT practitioner. And if you’re an IT practitioner, then you’ve got horror stories to tell, sleepless nights you’ve lived through, and probably more than one moment in your career where you’ve had a really good ethical debate about doing what you’re told.

# The Impostor Phenomenon

The Impostor Phenomenon was identified from clinical observations during therapeutic sessions with high achieving women by Dr Pauline Clance. Despite objective evidence of success, these women had a pervasive psychological experience believing that they were intellectual frauds and feared being recognised as impostors. They suffered from anxiety, fear of failure and dissatisfaction with life.



Source: Sakulku, J. (1). The Impostor Phenomenon. The Journal of Behavioral Science, 6(1), 75–97. <https://doi.org/10.14456/ijbs.2011.6>

“Imposter Syndrome”—that deep-down feeling that you really don’t measure up—is quite common among IT professionals, especially when we turn our eyes to security. While some of us genuinely believe in our own infallibility, most of us are aware that even industry luminaries like Bruce Schneier<sup>1</sup> don’t—and can’t—know it all.

Fear and doubt can quickly lead to paralysis. If someone who devotes their career to IT security can’t possibly know it all, how can we ever be expected to secure our networks? If the nerds at Google,<sup>2</sup> Amazon<sup>3</sup> and so forth can cause cloud outages, what chance do we have of getting everything right?

Most of us don’t work at a hyperscale cloud provider. If we’re really lucky, we work at some place like Equifax that at least paid lip service to—and threw money at—IT security.

<sup>1</sup> <https://www.schneier.com/>

<sup>2</sup> [https://www.theregister.co.uk/2016/03/01/google\\_cloud\\_wobbles\\_as\\_workers\\_patch\\_wrong\\_routers/](https://www.theregister.co.uk/2016/03/01/google_cloud_wobbles_as_workers_patch_wrong_routers/)

<sup>3</sup> [https://www.theregister.co.uk/2017/03/02/aws\\_s3\\_crash\\_result\\_of\\_fatfingered\\_command/](https://www.theregister.co.uk/2017/03/02/aws_s3_crash_result_of_fatfingered_command/)



I understand if you're startled that Equifax would be mentioned in anything resembling a positive way in a piece devoted to security. Equifax was responsible for one of the worst<sup>4</sup> IT security breaches in history, and as a result it's a popular social media blood sport to excoriate Equifax from top to bottom.

The part nobody wants to talk about, however, is that for all that Equifax failed both spectacularly and catastrophically, Equifax did invest time and resources into IT security, however limited. And that investment is more than the majority of organizations—especially smaller ones—have proven willing to make.

Being asked to square the IT security circle is daunting enough if you're confident in your knowledge and work for a company investing in it. It's perfectly understandable, then, if the rest of us are terrified.

## Don't Panic

In reading articles about IT security online, I often read some variation of “in order to protect a network, defenders have to defend against every possible attack, forever, while attackers only have to succeed once.” I am even guilty of adding this horrible trope to more than one of my own articles. It's an incorrect approach to thinking about security.

IT security is not a zero-sum game. It's not something where you either get everything right or your efforts are useless. IT security is cumulative: Every small effort builds on those that have gone before to ultimately create a solution that's far more secure than that which existed before the efforts began.

And as each of us increases the security of our own individual networks, we also are collectively raising the bar for attackers. As more of us prevent attackers from compromising our networks with

<sup>4</sup> [https://www.theregister.co.uk/2017/10/10/equifax\\_uk\\_records\\_update/](https://www.theregister.co.uk/2017/10/10/equifax_uk_records_update/)

simple, inexpensive attacks, then together we are raising the costs of compromising IT security, and diminishing the rewards.

It doesn't matter how awful the organization we work for is, there's always something we can do, even if we don't have the authority or funding to do everything we know we should. Consider for a moment the Quad9<sup>5</sup> DNS service.

Quad9 is a DNS service offered by the Global Cyber Alliance,<sup>6</sup> which was co-founded by the City of London Police, the Center for Internet Security, and the District Attorney of New York County. DNS requests made to the Quad9 DNS service are checked against the IBM X-Force threat intelligence database, as well as lists from Abuse.ch, F-Secure, ThreatSTOP, and many others.

Malware is unlikely to be able to contact its command and control (C&C) server using DNS if that malware uses the native DNS resolver of an operating system environment (OSE) whose DNS runs through Quad9. By configuring your network to forward DNS requests to 9.9.9.9 for IPv4 and 2620:fe::fe for IPv6, you'll have made a small step toward hobbling potential malware for virtually no effort.

Yes, malware authors can get around this by encoding their own DNS servers into their malware, or by hardcoding IP addresses into their malware. But both of these behavioral changes cost malware authors time and resources, and both changes have consequences. Hardcoding the IP addresses of C&C servers into malware would make finding those servers and shutting them down very easy for security researchers and law enforcement. Having malware make DNS calls to servers other than Quad9 would make those DNS calls stand out like neon beacons to properly configured monitoring solutions, alerting administrators to a compromised system.

<sup>5</sup> <https://www.quad9.net/>

<sup>6</sup> <https://www.globalcyberalliance.org/>

In other words, we don't have to be security wizards for our IT security efforts to be worthwhile. Nor must we work for the most functional of organizations, or see all of our efforts supported as we know we should be. Even the smallest, simplest changes have an effect, no matter how much of an imposter we might feel like in making the attempt.

## **Network Effects**

Implementing every IT security concept mentioned in this book would be expensive to the point of ruinous for small businesses. It would be overwhelmingly difficult to implement for most midsize businesses, and it doesn't cover enough of the quirky little edges to solve all enterprise IT security woes.

Reading this book won't make you a security guru. Also, sadly, your humble scribe possesses no novel insights that will pry open the wallets of the frugal, nor cause end users nor management to obey.

What I hope this book does provide is something of a check-box list of security basics. More importantly, I hope to be able to demonstrate to those who read all the way through exactly how all of these various security approaches tie together.

# CHAPTER 1

## Basic Principles

There are certain basic IT security concepts that need to be understood before conversing about more domain-specific approaches will be of any use. These basic principles transcend specialties and impact technologies across all of IT.

One myth that needs to be killed off early in this book is the “password rules thing.” Password history, complexity rules and regular expiration are bad.<sup>7</sup> They don’t actually help with security and just cause frustration.<sup>8</sup> Administrators only get so many frustration points before users revolt, and it’s increasingly considered a good plan to spend one’s limited frustration points on two-factor authentication (2FA), or simply on higher password length.

## Threats

The first security principle to discuss is that of security threats. Everyone and everything is a security threat, each to a greater or lesser degree. Unfortunately, all of us have some blinders about who and what is or isn’t a threat.

Many IT practitioners, for example, blame users for everything. It’s true that the overwhelming majority of security incidents are caused by end users, but it benefits no one to casually dismiss user threats as “not an IT problem.”

<sup>7</sup> <https://venturebeat.com/2017/04/18/new-password-guidelines-say-everything-we-thought-about-passwords-is-wrong/>

<sup>8</sup> <https://www.semperis.com/microsoft-upends-traditional-password-recommendations-with-significant-new-guidance/>

Too often, IT practitioners treat user ignorance of security as a moral failing: something to be met with chastisement and shaming. This approach doesn't solve the problem. Neither does endless end-user training.

You can't fight human nature, and no human can be alert to all threats, all of the time. All it takes is one person to get taken in by a phishing email, and anyone—even experienced IT practitioners—can become a victim.

Some social engineering experts put months of research into a spear phishing campaign. I'd be vulnerable to such an effort. So would anyone reading this book. That's a bitter truth, but accepting it is the very beginning of developing a useful approach to IT security.

## Human Nature

Humans have a natural tendency toward siege mentality. It's more prevalent in some cultures than others—it's particularly notable<sup>9</sup> in my home country of Canada, for example—but it occurs all over. From an IT security perspective, this tendency is a problem, in large part because it narrows our focus on whom we consider threats, and how we evaluate the likely risk those threats represent.

We all find it easy to talk about defending our network from the boogymen on the other side of the internet connection. Hackers and other internet miscreants are a readily identifiable “other.” They're easy to paint as an out-group; thus, finding resources and political will to defend against them is comparatively easy.

Partially because defending the castle walls is where the resources tend to go, insider threats are reported again<sup>10</sup> and again<sup>11</sup> as being

<sup>9</sup> [https://en.wikipedia.org/wiki/Survival:\\_A\\_Thematic\\_Guide\\_to\\_Canadian\\_Literature](https://en.wikipedia.org/wiki/Survival:_A_Thematic_Guide_to_Canadian_Literature)

<sup>10</sup> <https://hbr.org/2016/09/the-biggest-cybersecurity-threats-are-inside-your-company>

<sup>11</sup> <https://www.isdecisions.com/insider-threat/statistics.htm>

## Beware the Insider Threat

Within government agencies, the second highest threat perceived by IT professionals is that of careless and untrained insiders.

Source: Unintentional Insider Threats: A Review of Phishing and Malware Incidents by Economic Sector.

July 2014

By Jeremy R. Strozer, Matthew L. Collins, Tracy Cassidy

<https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=296268>



the most frequent threats to our networks. In some cases, insiders are reported to be responsible for nearly 75%<sup>12</sup> of security breaches.

These insider threats aren't all evildoers twirling their mustachios or cooking up elaborate plots to take down the internet. They're also you, and me, our bosses, and even the cleaners that come by every Thursday night.

How many times has Oopsie McFumblefingers CCed instead of BCCed?<sup>13</sup> I'll bet that the UK's National Health Services (NHS) investigates the GPO<sup>14</sup> to remove "Reply All" from Outlook after it nearly brought its IT systems to its knees.<sup>15</sup> And I could bring to the table a story about the cleaners unplugging the Intrusion Detection System (IDS) to plug in the vacuum, resulting in a ransomware compromise event and a lot of sleepless nights.

<sup>12</sup> <https://securityintelligence.com/news/insider-threats-account-for-nearly-75-percent-of-security-breach-incidents/>

<sup>13</sup> <https://www.howtogeek.com/128028/htg-explains-whats-the-difference-between-cc-and-bcc-when-sending-an-email/>

<sup>14</sup> <https://blogs.technet.microsoft.com/exchange/2009/09/29/removing-reply-all-functionality-for-outlook-users-who-participate-in-reply-all-storms-via-group-policy-what-to-do/>

<sup>15</sup> [https://www.theregister.co.uk/2017/01/31/nhs\\_reply\\_all\\_email\\_fail\\_half\\_billion\\_messages/](https://www.theregister.co.uk/2017/01/31/nhs_reply_all_email_fail_half_billion_messages/)

A threat isn't always a malefactor. Inattention to one's duties is a threat. End-user apathy is a threat. Burning your sysadmins out like spent candles because you nickel-and-dime everything is a threat.

And, yes, even foreign governments are a threat. Especially if you're charged with implementing regulatory compliant IT, and you find yourself subject to next-generation privacy regulations such as the European Union's General Data Protection Regulation (GDPR).

The first step in IT security is defining your threats, and continually re-weighting their importance. As the most common compromise vectors are mitigated, the relative importance of individual threats changes.

Oopsie McFumblefingers makes a common appearance in all organizations; however, the threat he presents might be negligible if the organization has great data protection and a well-maintained Security Information and Event Management (SIEM) system. Conversely, a foreign government using a silent subpoena to poke through the data you store in a public cloud provider might seem highly unlikely to occur, but the GDPR's fine of "4% of annual global turnover, or €20 Million (whichever is greater)" can make protecting against even such seemingly unlikely events seem important in a hurry.

Keeping an internal list of threats, the weighting of those threats, and the rationale behind that weighting is an important part of securing one's IT. This list should be regularly revised, and IT security priorities adjusted accordingly.

## Physical Security

The next step in IT security is physical security. If you put your data on a single hard drive, and someone copies that hard drive, they have all of your data. If they steal that hard drive, they not only have your data, they deny you access to that data.

In the worst-case scenario, if an organization has put all of its data on that single hard drive, and that organization doesn't have a backup of that data, then a single theft may just have ended that organization. In this case, the loss of that single hard drive—to theft, hardware failure, environmental disaster, and so on—is an existential threat to the organization. It's not hard to make an argument for a cloud backup solution, a padlock, and maybe the use of a self-encrypting hard drive to such an organization.

At the scale of a data center, physical security is more difficult. When one starts talking about physical security for an organization with multiple sites, a mobile workforce, and where workloads exist on-premises, in a services provider cloud and in a hyperscale cloud, physical security can quickly become a nightmare.

Physical security has three basic approaches. The first is access denial: Locking things up offers a basic prevention against people taking your stuff. If you have locks, add a fence. If you have a fence, add guards. If you have guards, randomly fill your data center with deeply disconcerting pictures of clowns whose cold, dead eyes are hooked up to motion sensors and follow intruders everywhere they go. Get creative! Every little bit helps.

## Potential Security Tactic: Scarecrow Clowns?

The word coulrophobia means a persistent and irrational fear of clowns. While there is a lack of official data on the prevalence of coulrophobia, some estimates state 12% of the US population suffers from it.





The second basic approach to physical is data locality: Don't allow data to physically exist where it's likely to be vulnerable to theft. This is typically viewed as an argument for virtual desktop infrastructure (VDI) or other remote application delivery methods, but it can just as easily be an argument for solutions that only unlock data or applications based on the GPS coordinates of the user. Yes, these are a thing. There are multiple vendors.

The third basic approach to physical security is active denial: even if someone does make it past the nightmarish zombie clown army to grab your disks, make sure that data is unreadable. Encryption is typically the front-line defense here, but remote-wipe technologies are also helpful.

## Unified Authentication

Unified authentication is far more important to security than most administrators will acknowledge. We spend our careers memorizing lists of credentials. As a result, we don't find it remotely odd to have to enter one username and password combination for this system, another for that one, and umpteen more besides. All of which expire on different schedules.

IT practitioners all know we shouldn't reuse passwords, but—surprise—everyone does it.<sup>16</sup> Even us. Given this, it's probably irrational to expect end users to play credential whack-a-mole, especially as the number of workloads—and workload providers—continues to increase.

Unified authentication has other security benefits as well. Unified authentication makes it easier to apply Access Control Lists (ACLs) in a consistent manner, as well as to track access and usage, even across infrastructures.

<sup>16</sup> <https://keepersecurity.com/assets/pdf/Keeper-Mobile-Survey-Infographic.pdf>

Consider an organization that uses multiple cloud providers. A multi-cloud management solution with integrated unified authentication would allow administrators to gain visibility and control that would be extremely difficult if that organization were restricted to the native cloud management tools of the various providers.

Let's use as an example an unfortunate developer who has a virtual machine (VM) on each hyperscale cloud provider. This developer's credentials have been compromised, and the attacker has installed ransomware into all of the developer's VMs.

If the SIEM solution used by the developer's employer has to look at each cloud independently, it may notice that a VM saw increased disk activity shortly after a login by the developer's user account, or it may not. SIEM-alerting thresholds are a tricky thing.

With unified authentication, however, that same SIEM solution wouldn't be looking at each cloud provider individually. The SIEM solution's alerting could be triggered in a number of places.

Perhaps the credentials compromise was automated, and the SIEM solution notices logins to all clouds simultaneously with the same developer credentials. If the developer doesn't normally automate his deployments in this fashion, that could be a behavioral red flag. Similarly, the SIEM solution could be triggered by having all VMs that a given user account has access to suddenly start pinning their storage.

Unified authentication is more than just a convenience. Unified authentication makes applying ACLs easier, as well as helping both humans and applications correlate activity. Ultimately, unified authentication allows us to make sense of authentication activity that otherwise is just too diverse to get a handle on.

# Principle of Least Privilege

No discussion about IT security has really begun until the principle of least privilege has been discussed. Least privilege dictates that no user—whether human or bot—be granted more access than is absolutely required to perform their duty.

Hypothetically, least privilege is easy. If Sally the sales wizard doesn't need access to anything except sales resources, don't give them to her. In practice, this is difficult.



Of all the basic IT security concepts, least privilege is the most fiendishly difficult to actually implement in practice.

Sally's user—or the groups to which that user belongs—may have access to all sorts of things by default. Defanging those defaults is one problem, and ensuring that changes to the default reflect Sally's existing user is another.

Some applications may not have granular enough controls to properly satisfy least privilege. Perhaps the point-of-sales application is ancient, so in order to make a sale in which inventory records are accurately altered to reflect the goods sold in a sale, Sally's user also has the rights to go into the inventory records directly and just start entering random numbers. Bad Sally. Do not do.

Modern privacy regulations have a bearing on this situation, as well. Least privilege is baked into GDPR, for example, and many expert interpretations of the GDPR greatly complicate life for everyone.

Let's say that Sally, bored with her career as a sales superstar, took up nursing. In order to do her job, Sally needs to be able to pull up patient records on patients for which she is responsible. Under

the GDPR, however, Sally shouldn't be able to rummage around in records of patients for which she is not responsible. Nor should she be able to access those records when she is not on shift, or if she takes her work tablet off-premises. Stick to Angry Birds, Sally, not angry auditors.

None of these examples are even touching the live wire that is needs assessment. Even a small business can easily have dozens of workloads, each with their own storage, their own OSE and application ACLs, and so forth. Large enterprises can have to deal with hundreds of thousands of employees and millions of workloads. Determining who needs to access what can be—and is—a full-time job for entire departments of individuals in the largest organizations, and even in these cases the best efforts frequently fall short.

Getting a handle on least privilege often requires a change in approach across the organization. Denying access by default is critical for all but the smallest organizations. Access rights should be defined both within infrastructure and workloads, as well as within monitoring solutions. This allows monitoring solutions to raise a flag on changes to rights, as well as access attempts—or successes—that don't agree with the rights defined in the monitoring applications.

Maintaining a copy of ACLs in one's monitoring application requires strict change controls to ensure that drift doesn't occur between infrastructure in use and the monitor solutions; but it's an increasingly vital form of automated alerting as the ratio of workloads to administrators continues to grow.

## **Defense in Depth**

Defense in depth is another critical concept for any technologist. The defense in depth approach is to layer defenses one atop another so that even if an attacker compromises one security layer, they must still penetrate additional layers in order to achieve their goal.

## Passwords Suffer From Conflicting Requirements

Passwords have been built into systems and used effectively as the sole authentication mechanism for many years. It is a technically mature technology, although passwordss suffer from two conflicting requirements: the passwordss must be sufficiently ‘random’ to prevent them being guessed by an attacker, yet simultaneously not too difficult for the user to remember. The security of a password-based authentication system relies on achieving the right balance between the two.



Source: Investigation on Vulnerabilities of Preboot and Post-boot Authentication

Xinzhi Liu and Lijun Wang

<http://cradpdf.drdc.gc.ca/PDFS/unc56/p525778.pdf>

Implementing defense in depth requires that administrators *always* assume compromise. Having anti-malware on your mobile employee endpoints, for example, is a good first step. But we all know that none of the anti-malware solutions catch all the creepy crawlies.

A defense in depth response to this reality might be to perform data flow introspection on all SMTP, SMTPS, HTTP, and HTTPS transiting the organization’s edge router. This should catch most of the creepy crawlies before they even make it to the endpoint.

Administrators might also employ the previously discussed Quad9 DNS to both help prevent users from ever being able to access sites that contain malware, as well as mitigate the damage malware can do if it does make it past both in-flight and at-rest anti-malware scans. Most organizations also have some form of configuration management to lock the system down, and prevent users from

doing silly things like running as local administrator, limiting the damage most malware can do even if the user does execute it.

Similarly, behavioral analysis of the endpoint's processes and outbound data flows could also catch malware in action. Because our hypothetical administrators are delightfully paranoid, we'll also say that all workloads with access to personally identifiable information are also delivered to endpoints via VDI or other forms of over-the-wire application virtualization. Finally, applications executing on the device itself are fully containerized with Bromium.<sup>17</sup>

In the scenario I just described, multiple layers of defense against contracting creepy crawlies were deployed. The assumption was also made that, eventually, something would find its way through and be able to execute. That something would have to then break out of its container. Assuming it did so, the changes it caused in the endpoint's behavior would be noticed, and administrators alerted.

Throw disk encryption and automated incident response (AIR) into the mix—such as auto-quarantining or even auto-wiping an endpoint whose behavior deviates too far from baseline—and we're leaning toward being able to claim that every reasonable step to defend that endpoint has been taken.

## Protocol and Standards Creep

Backward compatibility is great for convenience, but terrible for security. From a vendor standpoint, it's difficult to strike the right balance, and this requires awareness on the part of systems administrators.

A classic example is LM and NTLMv1 authentication. These older authentication methods are horribly broken, and thankfully easily disabled.<sup>18</sup> From a security standpoint, disabling these protocols

<sup>17</sup> <https://www.bromium.com/>

<sup>18</sup> <https://www.techrepublic.com/article/tech-tip-lock-down-systems-by-disabling-lm-authentication/>

is pretty important, but from a usability standpoint, having these protocols off by default was an important part of the initial negative reactions to Windows Vista.

Similar tales can be told about SSL 3.0,<sup>19</sup> which absolutely must be dropped in favor of TLS 1.2 or TLS 1.3 as soon as possible. In fact, all SSL versions, as well as TLS 1.0 and 1.1, need to be dropped<sup>20</sup> immediately.

Next to the principle of least privilege, staying on top of protocols and standards that need to be retired—and actually doing so—might be the greatest challenge in IT security. This is not something where administrators can simply trust that vendors will solve the problem for us.

Many vendors will issue guidance in their blogs, but the defaults in the actual products they ship won't necessarily be aligned with their own published best practices guidance. In addition, vendors will often only make changes to protocol and standards support on the very latest versions of their product, using it as one more lever to convince organizations to upgrade from older products.

<sup>19</sup> <https://blog.mozilla.org/security/2014/10/14/the-poodle-attack-and-the-end-of-ssl-3-0/>

<sup>20</sup> <https://blogs.technet.microsoft.com/askpfplat/2018/02/12/retire-those-old-legacy-protocols/>

## CHAPTER 2

# Network Security

Like all aspects of IT security, network security comes in flavors. There's security for preventing others from spying on your network traffic. There's security for ensuring that anyone who does spy on your network can't extract meaningful insights. And then there's the use of networks as part of securing workloads, which is typically accomplished by spying on the network traffic of those workloads.

Attempts to prevent people from spying on your network traffic are, for the most part, pointless. For good or ill, passive network taps have always been ridiculously easy to create. From the venerable vampire tap<sup>21</sup> to today's Cat-6 passive taps,<sup>22</sup> and even into fiber,<sup>23</sup> if you can get access to the physical network, you can figure out what's transiting it.

We'll leave quantum communications out of this for now. If you can afford quantum communications, you can afford helicopters full of identically-suited security experts to advise you. So I feel OK with leaving that particular niche out of this Gorilla Guide.

Wireless is no better. The Wi-Fi WEP encryption was famously broken<sup>24</sup> almost as soon as it came out, and was replaced with WPA, then WPA2, both of which have also been cracked.<sup>25</sup> WPA3 certification has only just begun,<sup>26</sup> which means that for the next few months at

<sup>21</sup> [https://en.wikipedia.org/wiki/Vampire\\_tap](https://en.wikipedia.org/wiki/Vampire_tap)

<sup>22</sup> <https://www.securityforrealpeople.com/2014/09/how-to-build-10-network-tap.html>

<sup>23</sup> [https://en.wikipedia.org/wiki/Fiber\\_tapping](https://en.wikipedia.org/wiki/Fiber_tapping)

<sup>24</sup> [https://en.wikipedia.org/wiki/Fluhrer\\_Mantin\\_and\\_Shamir\\_attack](https://en.wikipedia.org/wiki/Fluhrer_Mantin_and_Shamir_attack)

<sup>25</sup> <https://www.gizmodo.com.au/2017/10/wi-fis-most-popular-security-method-might-be-broken/>

<sup>26</sup> <https://www.theverge.com/circuitbreaker/2018/6/26/17501594/wpa3-wifi-security-certification>



## Network Security Is Increasingly Critical

As of Dec 31, 2017, there were an estimated 4,156,932,140 Internet users in the world. This is approximately 54.4% of the world population. An estimated 121.7 exabytes per month of traffic transited the Internet in 2017. Internet traffic is expected to more than double by 2021, reaching over 278.1 exabytes per month.



The continued phenomenal growth in both Internet users and Internet traffic are reasons to consider networking knowledge – and especially network security knowledge – one of the most important knowledge domains for any organization.

The Internet connects nearly all computer systems on the planet to one another. This allows individuals, bots, and compromised systems to attack computers anywhere in the world, regardless of geographic proximity.

The potential threat landscape of network security is “everything and everyone connected to the Internet”. At the end of 2017, that was over half of the Earth’s human population, and that percentage is growing every year.

Sources: [1] <https://www.internetworldstats.com/stats.htm>

[2] [https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html#\\_Toc484813982](https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html#_Toc484813982)

least, Wi-Fi cannot be secured. Also: all civilian cellular networks and protocols are insecure,<sup>27</sup> including LTE.<sup>28</sup> So there's that.

This leaves network security to focus on securing data flows so as to prevent the bad guys from seeing what's going on, while at the same time ensuring that the good guys can pry those same data flows open to see what's going on.

## Networking 101

To understand network security, one must first understand a little bit about how networking works. To keep things simple we'll stick with the TCP/IP stack and Ethernet, and this will be a super-quick review that assumes you know all the basics, but might need them freshly called to mind.

Data center networking has two network address types that concern administrators: physical (or MAC) addresses, and logical (or IP) addresses. A given physical network interface card (NIC) can have multiple addresses of each type, as can virtual NICs. Computers can have multiple physical NICs and OSEs can also have multiple virtual NICs.

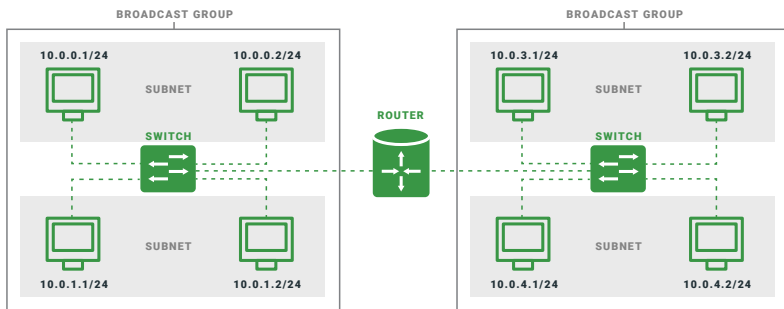
A subnet is a logical network. All workloads or devices that share a subnet can communicate with one another by addressing the IP address of other workloads or devices on that subnet. I won't explain how to calculate subnet masks here because the internet is full of websites happy to do this.<sup>29</sup>

Also, thankfully, the details of subnetting are actually completely irrelevant to network security. All we need to know about subnets is that for a workload or device on one subnet to talk to a workload or device on another subnet, packets must transit a router. Routers

<sup>27</sup> <http://thehill.com/policy/cybersecurity/277329-serious-weaknesses-seen-in-cell-phone-networks>

<sup>28</sup> <https://arxiv.org/pdf/1607.05171.pdf>

<sup>29</sup> <https://www.techrepublic.com/blog/data-center/ip-subnetting-made-easy-125343/>



**Figure 1:** This diagram shows a network with 4 subnets and two broadcast groups.

are the network magic that allow subnets to communicate with one another.

Broadcast domains, though not a thing in IPv6, are another important concept, because IPv4 isn't going away anytime soon. All devices and workloads located in a single broadcast domain can see broadcasts from all other devices and workloads located in that broadcast domain, regardless of subnet.

Consider four devices physically wired into a switch: 10.0.0.1/24, 10.0.0.2/24, 10.0.1.1/24, and 10.0.1.2/24. Because they're on the same subnet, 10.0.0.1/24 and 10.0.0.2/24 can talk to one another, as can 10.0.1.1/24 and 10.0.1.2/24, but 10.0.0.1/24 and 10.0.1.1/24 cannot talk to one another, because they're on different subnets.

When 10.0.1.1/24 wants to speak to 10.0.1.2/24, it sends out a broadcast to determine what 10.0.1.2/24's MAC address is. This broadcast can be seen by 10.0.0.1/24, which means that even if 10.0.0.1/24 can't communicate with 10.0.1.1/24, 10.0.0.1/24 can see with whom 10.0.1.1/24 is trying to communicate.

VLANs are used to break up broadcast domains. In the device/switch scenario, this means preventing 10.0.0.1/24 from gaining insight into 10.0.1.1/24's communications. Combined, VLANs and subnets create network barriers that have useful properties for IT security purposes.

# Edge Security and Introspection

For workloads or devices on one subnet to talk to workloads or devices on another subnet, they must transit a router. The router then forms what's called the network's edge. All traffic that leaves the subnet transits the router, which makes the router the perfect place to start prying open data flows to see what's inside, and even manipulate them.

Observing and manipulating the contents of a network flow is called introspection when done by the good guys, and a man-in-the-middle attack when done by the bad guys. For unencrypted network flows such as HTTP or SMTP, introspection is simple.

When performing introspection, routers manipulate the data packets to cause the TCP sessions to terminate at the router, or at the address of a third-party network security solution. The router or third-party security solution then manipulates the contents of data flow, for example, by stripping out all references to websites contained on a blacklist, or stripping infected attachments from an email.

The router then creates a new TCP session between the router or third-party security solution and the workload or device for which the original TCP data flow was intended. The modified data flow is sent instead of the original.

Encrypted data streams—such as HTTPS—require more effort. For routers to successfully introspect an encrypted data stream, the protected workload in question must trust the router. In this case, the router intercepts the request for an encrypted communication with a protected workload and terminates that workload, as described earlier.

When creating the replacement data flow to the protected workload, however, the router must use its own encryption certificate. As a result, the protected workload must trust the router's certificate.

While unencrypted data flows can be introspected without the introspection requiring the active participation of protected workloads, the same isn't true of encrypted data flows. This makes securing encrypted data flows more work for administrators, but it also means that any attempt by the bad guys to perform a man-in-the-middle attack on an encrypted data flow will be really obvious.

## Firewalls

Firewalls are a key component of network security. Whether used as part of an individual workload, or as part of a router, the purpose of a firewall is to determine which data flows may pass.

In addition to source and destination MAC and IP addresses, TCP/IP packets are addressed to a specific port. In their most basic configuration, this combination of address and port is used by firewalls to determine if traffic should be allowed to pass.

If a workload only presents its resources on port 80, for example, then all traffic attempting to contact that workload on ports other than 80 should be denied by the firewall, as well as logged to determine if those connection attempts represent a threat. The more narrow the range of acceptable address and port combinations can be made, the more effective that firewall is.

A workload that operates as part of a tightly coupled service, for example, may be expected to only communicate with one other workload that forms its service. Here, our workload may present its resources on port 80 as before, but any attempts to connect to it from an IP address other than that of the partner workload in its service should probably be considered an attack.

# Microsegmentation

Network segmentation, at its most basic, uses a combination of subnetting and VLANs (or, more frequently, VXLANs) to isolate individual workloads and groups of workloads tightly coupled into services from one another. The purpose of this is to create as many network edges as possible, separating workloads from one another while allowing for the maximum possible data flow introspection points.

When combined with firewall automation and orchestration, network segmentation becomes microsegmentation, and is an important tool for preventing the spread of compromise throughout a network. Because workloads and services are individually isolated from one another—with all their traffic passing through a router-controlled firewall, as well as potentially being inspected by monitoring solutions or even introspected—a compromised workload cannot start launching attacks against other workloads without being noticed.

In practice, the most effective microsegmentation solutions also incorporate layer 2 extensibility. Layer 2 extensibility allows connecting networks located on multiple infrastructures together as though they were a single network. This means that microsegmentation can both allow workloads running on multiple infrastructures to work together as part of a service, as well as isolate that service from the rest of the network.

Currently, microsegmentation is the only network strategy that allows for a unified approach to workload and service segmentation across multiple infrastructures. As a result, microsegmentation is an increasingly important IT security solution for organizations engaging infrastructures from multiple providers, in addition to their own on-premises solutions.

## Security Principle: Microsegmentation

Microsegmentation is an emerging network security market. This area of IT endeavor is still young enough in its lifecycle that technical and marketing definitions are somewhat fluid, with different vendors having different definitions terms.



Microsegmentation vendors that focus exclusively on firewall automation and orchestration, for example, would take issue with the inclusion of network segmentation technologies (such as VXLANs) in the definition. This is in part because these vendors do not include these technologies in their products.

Other vendors, which do combine both network segmentation and firewall automation/orchestration in their products insist that both technologies are necessary for true microsegmentation. This argument is partly based on marketing, and partly on technology.

No firewall can prevent all workload compromises, and eventually, something on the network will be compromised. The practical difference between the two approaches to microsegmentation comes down to what happens when a workload is compromised.

Microsegmentation products that rely entirely on firewall automation and orchestration can protect a workload from begin attacked, but they have limited means to prevent a compromised workload from attacking the rest of the network.

Microsegmentation products that incorporate network segmentation, however, can protect the rest of the network by preventing a compromised workload from communicating with the rest of the network. This is done by removing that workload's ability to traverse the network segment edge.

The difference between the two approaches is the presumption of compromise. A microsegmentation solution that incorporates network segmentation presumes that one or more workloads will eventually be compromised, even if it is defended by firewall automation and orchestration. The network segmentation is another layer of defense.

Remember: it takes only seconds to compromise a workload or device, and most networks are filled with devices – from printers to IoT devices – that cannot participate in (and thus cannot be protected by) firewall automation and orchestration solutions.

In highly virtualized environments, microsegmentation is combined with IT automation to enable the implementation of network-based security solutions on a scale that would be infeasible if done manually. Microsegmentation's use of VLANs allows a VM's network security context to travel with that VM as it migrates from host to host within a cluster, or even as it's moved to completely different infrastructures, such as those operated by a cloud provider.

## **Port-Based Network Management**

Not all network environments are rapidly-changing, highly-dynamic environments. While microsegmentation may be the current bleeding-edge technology of network security, there's still life left in security approaches that are now decades old. One example of this is port-based network management.

Where administrators know which devices and workloads are expected to be present on a particular switch port, they can prevent anything other than the expected devices and workloads from communicating on that switch port. Any attempt by an unauthorized device or workload to communicate on that port can be logged, and administrators alerted.



## CHAPTER 3

# Endpoint Security

Securing endpoints arguably occupies the majority of an organization's security effort. Some of this is driven by vendors: much of the commercial focus of IT security is on securing endpoints. But a great deal of the effort in security endpoints is due to the perception of the end user as a security threat.

To properly consider endpoint security, one must first define what an endpoint is. For many—especially vendors—an endpoint is any device used by an end user. A desktop PC, notebook, mobile phone or tablet would be universally considered an endpoint. Beyond this, however, definitions fray, depending on which vendor is selling what solution.

Most servers and embedded systems run the same (or very similar) OSEs as end-user operated systems. They have very similar security concerns, and are managed by the same management solutions. Systems administrators tend to lump servers and embedded systems into the term “endpoint,” though many vendors license their management solutions differently based on the version of the OSE, or its intended use.

Frequently forgotten in discussions about securing endpoints—and often forgotten during security discussions in general—are all the other endpoints in an organization. The term *endpoint* comes from networking, and basically means “anything that can communicate on a network.” From a security standpoint, this definition of endpoint is the one that matters.

Endpoints, then, are not just servers, embedded systems, desktops, notebooks, mobiles and tablets. Endpoints are also printers, IoT sensors, switches, routers, and much, much more. All endpoints are the target<sup>30</sup> of modern attackers, and to the extent that it's possible, all endpoints must be secured.

## Best Effort

Offline attacks range from the famous-but-dated Van Eck Phreaking,<sup>31</sup> to various takes<sup>32</sup> on Near Sound Data Transfer (NSDT),<sup>33</sup> including the now-famous TEMPEST<sup>34</sup> attack vectors. While one is exceedingly unlikely to ever find themselves the target of NSDT or TEMPEST attacks, these extreme examples of vulnerability should serve to illustrate the futility of relying on endpoint security alone.



Unfortunately, in reality, no endpoint can ever be fully secured. Any device that's connected to a network is vulnerable. In fact, even removing a device from a network doesn't render it immune to attack.

Any device connected to a network exposes vulnerabilities. The network stack of the OSE, the firmware of the network card, as well as potentially the firmware of the device's LAN-on-motherboard (LOM) are all attack surfaces, even with firewalls firmly in place and denying all traffic.

<sup>30</sup> [https://www.theregister.co.uk/2018/03/09/slingshot\\_malware\\_uses\\_cunning\\_plan\\_to\\_find\\_a\\_route\\_to\\_sysadmins/](https://www.theregister.co.uk/2018/03/09/slingshot_malware_uses_cunning_plan_to_find_a_route_to_sysadmins/)

<sup>31</sup> [https://en.wikipedia.org/wiki/Van\\_Eck\\_phreaking](https://en.wikipedia.org/wiki/Van_Eck_phreaking)

<sup>32</sup> [https://www.theregister.co.uk/2018/03/12/turning\\_speakers\\_into\\_covert\\_listening\\_devices/](https://www.theregister.co.uk/2018/03/12/turning_speakers_into_covert_listening_devices/)

<sup>33</sup> [https://en.wikipedia.org/wiki/Near\\_sound\\_data\\_transfer](https://en.wikipedia.org/wiki/Near_sound_data_transfer)

<sup>34</sup> [https://en.wikipedia.org/wiki/Tempest\\_\(codename\)](https://en.wikipedia.org/wiki/Tempest_(codename))

Despite the seeming futility, endpoint security is an important part of defense in depth, and critical to an organization's overall security posture. There are three basic approaches to endpoint security: baselining, segmentation, and hardening.

## Presumption of Compromise

Baselining and segmentation are important for endpoints that cannot be secured. An IoT light bulb, for example, probably isn't securable. IoT vendors tend to offer limited support, and the security of most IoT devices—especially<sup>35</sup> the consumer broadband routers given out by most ISPs—is notoriously terrible to begin with.

Once infected, IoT devices become a platform from which attacks can be launched against other systems. This is where baselining and segmentation come in. All devices and workloads should be baselined. Ideally, baselining should be done in an automated and ongoing fashion. Deviations from baseline should not only trigger alerts, but should trigger automated responses, such as quarantine.

Microsegmentation is also useful when defending the inherently indefensible. Isolating known-vulnerable workloads reduces their chance of compromise. More importantly, if the microsegmentation is applied such that it secures packet egress from the vulnerable network, as well as packet ingress, then microsegmentation greatly reduces the chance that an infected endpoint can compromise anything else on the network.

The presumption of compromise underlies defense in depth, and it's critical to how one approaches endpoint security. Everything related to endpoint security—whether we're talking about an IoT light bulb, or someone's notebook—is about adding layer upon layer of security and hoping that holds.

<sup>35</sup> [https://www.theregister.co.uk/2018/06/07/vpnfilter\\_is\\_much\\_worse\\_than\\_everyone\\_thought/](https://www.theregister.co.uk/2018/06/07/vpnfilter_is_much_worse_than_everyone_thought/)

At the same time, because one is presuming compromise, one must be prepared to act once the endpoint is ultimately compromised, so knowing how to wipe the device and restore it to expected functionality is just as important a part of endpoint security as is throwing up roadblocks against potential attackers.

## Endpoint Management Solutions

Endpoint management solutions typically focus on securing devices used by end users. There are numerous enterprise endpoint management solutions available. While each has a different set of features and supports a different collection of endpoints, each operates in a similar fashion.

Traditionally, endpoint management solutions focused on configuration management. One of the most basic and pervasive forms of endpoint management is Microsoft's Active Directory (AD). Rough equivalents of AD to Linux are Spacewalk,<sup>36</sup> Sattelite,<sup>37</sup> and similar solutions.

AD-style configuration management solutions allow systems administrators to create policies that apply configurations to OSEs and applications. These configurations may or may not have a security element, such as preventing Internet Explorer in Windows OSEs from executing ActiveX components.

Endpoint management solutions in the first decade of this millennium focused on ensuring that AD-style configurations applied to all devices, regardless of where they were. Notebooks, mobile phones, and tablets spent considerable amounts of time disconnected from an organization's network, and systems administrators needed a way to ensure that policies they crafted were being applied.

<sup>36</sup> <https://spacewalkproject.github.io/>

<sup>37</sup> [https://en.wikipedia.org/wiki/Satellite\\_\(software\)](https://en.wikipedia.org/wiki/Satellite_(software))

Modern endpoint management solutions focus more on securing information than they do on securing devices. Diverse solutions are used, from application virtualization, remote delivery, and containerization, to encryption, theft detection, and remote wiping.

Endpoint management is a fundamental component of IT security and can't be neglected, even if most endpoint management solutions only cover a fraction of the endpoints in use in any organization.

## Patch Management

Patch management of endpoints is typically included in configuration management solutions, and is a core feature of most endpoint management solutions. Patch management is essential to IT security.

Patches fix known security bugs, reducing one's attack surface; but patches can also break applications and OSEs. Patch management not only allows administrators to force devices and workloads to apply updates, but also allows for staged rollouts of patches.

*Canary groups* are an important concept in patch management. Canary groups are users and workloads that receive patches before the rest of the organization. Canary group members should be selected to represent a diverse cross-section of devices, execution environments, and applications.

The purpose of canary groups is to test patches on a limited subset of users and workloads to see if the patches break anything. If they do, patch distribution to the rest of the organization can be halted until vendors are consulted and workarounds, hotfixes, or new patches are issued.

While delaying patches doesn't at first seem important or relevant from a security point of view, it's important to remember that IT security must not be about IT by fiat. IT exists to serve a purpose.

That purpose is often making a human being's life easier, generating profit, or even keeping life-saving machinery working.

If the humans involved become wary of patches—or IT security in general—then those patches will be avoided. In extreme cases, people will start to actively attempt to circumvent security to do their jobs.

Active circumvention attempts by authorized users is one of the worst possible outcomes of any IT security endeavor, and one of the most frequent triggers for such undertakings is inadequate patch management. Few things make an entire workforce loathe IT more than a bad patch bricking everyone's computers.

## Anti-Malware

Anti-malware solutions are the Hail Mary<sup>38</sup> of IT security. If these come into play, then it means all other security measures have failed. Anti-malware solutions are supposed to detect malware as it attempts to execute on an endpoint, and either prevent it from doing so, or clean it up after the fact.

The effectiveness of anti-malware solutions is questionable at best. Signature-based anti-malware solutions can only protect against known malware, and malware authors train their malware against heuristic anti-malware solutions.

The latest iteration of anti-malware offerings are bulk data computational analysis (BDCA)-based “next-gen” anti-malware solutions. These solutions range in effectiveness from surprisingly effective to roughly as effective as prayer, depending on the malware they're confronted with, and the attack vector it uses.

While even the best of next-gen anti-malware is by no means remotely enough by itself to secure anything, anti-malware remains

<sup>38</sup> <https://www.consumeraffairs.com/news/just-how-effective-is-antivirus-software-070816.html>

an important final hurdle, especially on endpoints where end users regularly open documents or use web browsers.

## Host-Based Intrusion Detection (HID)

When all the other layers of security have failed, IT security turns from prevention to detection. Network-based intrusion detection has been previously discussed and ranges from firewalls and edge security, to data flow introspection and behavioral profiling.

Intrusion detection can—and should—live on endpoints as well. While network-based intrusion detection examines data flows, HID examines running processes. Both forms of intrusion detection look for irregularities in behavior.

HID solutions typically use an agent, which may execute at the OSE level, or the execution environment level. The rise of containerization makes determining the observation capabilities of the HID solution important: some containerization solutions provide more isolation for applications than others, meaning OSE-level HID agents may not be able to accurately observe containerized workloads.

By the same token, HID agents included in a containerized environment in order to observe a specific workload are unlikely to be able to fully observe the host OSE. Fully instrumenting an execution environment when using containerization may require agents both at the OSE and the container level, depending on configuration.

The operation of HID solutions is deceptively simple: observe processes, look for odd behavior, and generate an alert if odd behavior is seen. Determining what qualifies as “odd” is where the magic lies.

Baselining can help, but only if the application behaves in a reasonably consistent fashion. Chrome, for example, would be functionally impossible to baseline, because it has almost evolved

## Malware and Organized Crime

Anti-malware may be a Hail Mary, but we can't ignore it. SonicWall Capture Labs researchers recorded 5.99 billion malware attacks in the first half of 2018, and that's just one organization's statistics.



Cryptominers and ransomware have been duking it out for top malware type for at least the past two years, and this is, of itself instructive: the majority of malware that any organization is likely to encounter is malware aimed at directly generating revenue for the malware author.

This malware is often tied to organized crime. Both cryptominer and ransomware malware types are developed by expert malware authors, but the attacks come from others who buy the software from the malware authors. There are sophisticated technical support, marketing and sales structures that resemble commercial software development, including telemetry to assist malware developers in building more effective products.

This business-like approach to malware extends beyond headline malware categories. There are, for example, emerging malware markets for industrial espionage and sextortion toolkits, among many other categories.

Source: <https://sensorstechforum.com/5-99-billion-malware-attacks-2018-ransomware/>

into its own OSE, complete with its own multimedia subsystems, storage solutions, network stack and more. Chrome's behavior can amount to everything an end user might possibly wish to do with their computer, hence the existence of Chromebooks.



The inverse of attempting to profile Chrome would be profiling a cron job. A cron job runs at predetermined times, performs a pre-determined activity, and that activity should look more or less the same every time. CPU usage, network activity, and disk activity of a regularly scheduled cron job should fall within some reasonably narrow boundaries with each execution, and HID solutions should have no problem spotting deviations from the norm.

OSSEC<sup>39</sup> is the standard HID solution to which all others are compared. In part this is because it's open source, although another reason for its use as a standard in the HID space is its ubiquity. OSSEC is not only frequently deployed on its own, it's also included in popular IT security offerings.

## Infrastructure as Code

Once an endpoint is known to be infected, it must be wiped and restored to a known good operating condition. There are three ways to go about this. The first is to rely on a firmware-based “restore to factory defaults” function. This is typical on IoT devices, and may or may not restore the device to a state prior to the latest patches having been applied.

Factory resets will clear all configurations, including connections to configuration management solutions. The second approach to endpoint restoration is imaging, which works in a similar fashion: An image is taken at a point where the workload is considered to be “known good,” and this image is re-applied when the device is compromised.

Both factory reset and imaging represent a period of significant vulnerability before devices can be brought into compliance with the latest configurations. This often includes a period of time during which workloads are not fully patched.

<sup>39</sup> <https://ossec.github.io/>

Unfortunately, both factory reset and imaging solutions rarely include any mechanism to inform administrators when an endpoint is fully up-to-date, compliant, and ready to resume operation. In addition, neither approach accounts for the possibility that the firmware or images themselves may be compromised.

Infrastructure-as-code solutions aim to resolve these issues, as well as take over from where traditional AD-like configuration management solutions leave off, especially in the Linux ecosystem. Puppet,<sup>40</sup> Chef,<sup>41</sup> Saltstack,<sup>42</sup> and Ansible<sup>43</sup> are the most popular configuration management solutions in the OSE management portion of the infrastructure-as-code space, with Terraform<sup>44</sup> being among the most popular solutions for addressing infrastructure components below the OSE.

The purpose of infrastructure as code is twofold. The first is to define, as much as possible, the exact details of an execution environment, from the bare metal through to the libraries and frameworks made available to applications. OSE and application configuration would be defined, as would patch levels, agents to be deployed, and more.

Infrastructure as code often includes defining monitoring parameters, sometimes including behavior profiles and baselines. An infrastructure-as-code approach aims to, as much as possible, separate a workload's data and configuration from the underlying execution environment.

Infrastructure as code makes rebuilding an execution environment extremely simple; a single line of code can instruct infrastructure to instantiate a workload based on a given configuration, attach storage, test that the workload complies with the configuration, and

<sup>40</sup> <https://puppet.com/>

<sup>41</sup> <https://www.chef.io/>

<sup>42</sup> <https://saltstack.com/>

<sup>43</sup> <https://www.ansible.com/>

<sup>44</sup> [https://www.theregister.co.uk/2017/12/06/what\\_is\\_terraform/](https://www.theregister.co.uk/2017/12/06/what_is_terraform/)

then enter that workload into service. Crucially, because all aspects of the workload are defined in code, compliance of the workload with its assigned configuration can be assessed in an ongoing manner.

Though frequently associated with DevOps and containerization, infrastructure as code can be applied to most endpoints. Routers, switches, servers, desktops, and phones are all examples of devices that can be subject to most (or all) of the process of wiping the device, injecting a new OS or hypervisor, and then applying configurations, patches, agents, and so forth until the device is in compliance.

Infrastructure as code may use any number of approaches to instantiate the workload's OS. It may be installed from a vendor-provided release-to-manufacturing .iso image, and patched up from there. The infrastructure-as-code solution may use an organization-maintained image, a template that slipstreams the latest patches in before installation, or anything in between.

As with factory resetting devices, or rebuilding from an image, infrastructure-as-code solutions should be kept as isolated as possible until they're confirmed to be in compliance. The one possible exception to this are infrastructure-as-code solutions that incorporate patch slipstreaming into initial OS deployment; though in a perfect world, even these would build in a controlled environment before being entered onto a potential production network.

## **Separate Management VM**

An often-overlooked element of endpoint security is the segregated management VM. Any endpoint can be compromised, including that of the systems administrator. A backup plan should exist for situations when the administrator's endpoint is compromised, and there's still management to do, but there isn't time to rebuild the systems administrator's endpoint.

Creating a VM that has all of the tools necessary to manage relevant infrastructure is a highly recommended check against this possibility. This VM need not ever be brought online, except for patching and diagnostics to ensure that it remains in compliance for current configurations; but it should exist just in case.

Remember: always presume compromise, and plan for as many compromises as are possible.

## CHAPTER 4

# Process, Auditing And Compliance

While technological approaches to IT security have thus far been the focus of this book, in reality, IT security is just as reliant (if not more so) on processes than it is on technology. IT security could reasonably be described as the rigorous definition of how everything should behave, combined with constantly monitoring everything to see if it deviates from that behavior.

At a bare minimum, then, IT security would need standards that allow the description (and reading) of definitions, a place to store definitions, a means to separate false positives (and false negatives) from relevant information, as well as change-management processes. That's a lot of red tape.

Modern regulatory regimes tend to focus on processes rather than technologies. Regulators 25 years ago would barely have been able to conceive of facial recognition as a real-world technology they might have to worry about. Today, not only is facial recognition widespread, applications exist that can convincingly replace a person's face in a video,<sup>45</sup> opening the door to automated defeat of even the most advanced facial recognition solutions sometime in the next decade.

Strictly defining one's IT security processes—especially change management—is important for a number of reasons. Efficiency, predictability, and user morale are all reasons to be meticulous about process.

<sup>45</sup> [https://www.theregister.co.uk/2018/01/25/ai\\_fake\\_skin\\_flicks/](https://www.theregister.co.uk/2018/01/25/ai_fake_skin_flicks/)

# Versioning of Configurations

In a perfect world, everything in IT configurations, including devices, OSEs, applications—all of it—would be deployed using the principles of infrastructure as code. Code can be put into a versioning system, and versioning systems provide huge advantages to security teams.

The theory goes something like this: when all infrastructure is defined as code, then every change made to the code that defines infrastructure should be recorded in and tracked by a versioning system. This means that every change to one's infrastructure is stored. The entire history of one's infrastructure, from its initial creation to the present, can be examined.

The versioning system should track who makes those changes, who tested those changes, and who authorized releasing those changes to production. If authority delegation is used at any point, the individuals delegating authority should also be tracked.

Configuration versioning has many uses. It allows (or at least assists with) rolling workloads back to a known good state. It allows tracking user authorizations that were used to imitate changes, which can help understand either why changes were made, or highlight compromised accounts.

Configuration versioning may also help catch temporary compromises, helping to identify malicious actors who are covering their tracks. Configuration versioning also plays a role in automating auditing processes.

# Security Information and Event Management (SIEM)

SIEM is an important part of compromise detection. SIEM solutions gather event log and performance data into a central repository and then attempt to extract insights from this data.

The extent to which SIEM solutions are useful to security teams depends greatly on the quality of the solution's insight functionality. A SIEM solution that simply forwards every alert from the entirety of an organization's IT to an administrator's email is completely useless. Those emails will quickly be ignored, and real issues will go unnoticed amid the flood of alerts.

Support for the totality of an organization's IT is also crucial. Much like purported endpoint management solutions, SIEM solutions can be quite limited in out-of-box support for infrastructure components, OSEs and applications.

Security teams should insist on solutions that perform event correlation, automated root cause analysis, behavioral analytics, and offer rich reporting and analytics. It's also good for SIEM solutions to integrate with authentication infrastructure, as this can lead to more accurate alert prioritization.

Authentication systems often contain information beyond just a username and password. This information can include, for example, the department a user belongs to, the geographic location of their office, and whether or not they're a mobile user. This data can be used to find correlations in events that otherwise aren't possible.

One of the most famous examples of how SIEM solutions can integrate expanded user awareness is login locality awareness. If a user logs into an account from San Francisco, and then 30 minutes later logs into that same account from Los Angeles, then

there's a high probability that either the user account has been compromised, or the user has a sci-fi portal gun. Both possibilities require immediate investigation.

## Encryption

Deploying encryption requires implementing technologies, but successfully deploying encryption requires keeping track of one's encryption keys.

The technological solution to handling encryption keys is a key management server. Think password manager, but for encryption keys. Like a password manager, the primary purpose of a key management server is convenience.

Password managers make multiple usernames and passwords convenient, so when we sign up for things on the internet we're

## Mountains of Logs

In practice, SIEM can very quickly become a Big Data problem. Computers generate a lot of logs, and the proliferations of IoT sensors is magnifying this for many organizations.



To learn more about this, it is worth taking the time to investigate Netflix's adventures adopting the Elastic Stack. The Elastic Stack (formerly ELK) is one of the major open source SIEM solutions.

Netflix's discussions about the challenges around data ingestion and analytics, and how this in turn caused the Elastic Stack to evolve, are highly instructive as regards the real world difficulties that implementing SIEM solutions can present.

Source <https://www.elastic.co/videos/netflix-using-elasticsearch>



less likely to use the same username and the same password every time. Similarly, key managers make keeping track of encryption keys possible, which in turn makes it more likely we'll actually use encryption.

Describing all of the different ways encryption can be deployed would be a Gorilla Guide of its very own. The short version is that for encryption at rest, one can use encryption on a per-application basis, frequently in concert with an endpoint management or VDI solution. Encryption at rest can also be done on a per-OS basis, a per-VM basis, or at the level of the underlying storage.

In each of these cases, the encryption keys for that storage will have to be stored somewhere, or the storage can't be unlocked. That "somewhere" is a key management server. Data protection for the key management server is absolutely critical, because if it's lost, so is all the data it protects. Similarly, whatever access solution is being used to authorize user and administrator access to that key management server needs to be completely invulnerable, or one won't be able to get into the key management server to get at the precious keys.

## **Data Protection**

Given that numerous regulatory regimes already mandate encryption, and that encryption is likely to be a standard requirement of all future regulatory regimes, it's safe to say that most organizations are going to have to learn how to deploy encryption. Security teams are typically held at least partially responsible for encryption implementation and key management.

This makes the ongoing viability of key management servers a serious concern to security teams, and brings data protection under security's scrutiny, as well. There are numerous other reasons, however, that security teams should already be paying attention to an organization's data protection practices.

The General Data Protection Regulation (GDPR), among other regulatory regimes, makes organizations as responsible for their backups as they are for their primary working data. A compromise of a backup is still a compromise, and personally identifiable information can be leaked from that attack vector just as readily as it can be from a production workload.

In addition, the GDPR and other emerging privacy regulations give citizens full control over their data. This includes the rights to be forgotten and to ensure that the data held about them is accurate. If citizens exercise these rights, that data needs to be modified in production systems, as well as in backups.

This can be something of a security nightmare, because yet another regulatory requirement (again of the GDPR) is that organizations keep track of all data accesses. If someone reads, creates, modifies, or deletes data, then what was done and who did it needs to be recorded.

Updating backed-up data to reflect deletion or modification requests can be a security challenge. It's unlikely that backups will be modified in real time. More likely, they'll be modified as part of a batch process. Organizations will have to decide if changes to those records should be reflected as coming from the citizen that requested the change, the user account that authorized the changes (human or otherwise), or the user account context under which the batch process is running.

Each choice may have different regulatory and auditing impacts, and each may affect change attribution for security purposes. As a result, some organizations are having to build separate access authorization structures for data protection access so that they can record different accesses for legal purposes and security purposes.

# Asset Detection and Management

Asset detection and management solutions are typically touted for financial or operational reasons. The purposes of these solutions is to detect when new IT has been added to the network, identify what it is, and potentially generate alerts about the addition.

These solutions can also be very useful to security teams. New things appearing on one's network are always a security concern, and in large enough organizations identification and classification of new arrivals needs to be automated.

Infrastructure as code, asset detection and management, and automated incident response are a potent combination. When combined, asset management systems can be informed ahead of time that new infrastructure is expected. Any infrastructure that appears and isn't expected to exist can be immediately and automatically quarantined. Any infrastructure that's expected to exist, but doesn't conform to expected configuration, can also be immediately and automatically quarantined.

## Vulnerability Scanning

The penultimate security consideration in this book is the humble vulnerability scanner. Even when operating an infrastructure that's fully defined as code, it's worth regularly scanning one's infrastructure to ensure that no known vulnerabilities exist.

Vulnerability scanners come in flavors. There are a number of commercial solutions on the market, as well as a number of open source offerings. OpenVAS<sup>46</sup> is the most prominent open source solution, and is often incorporated into commercial offerings.

Vulnerability scanners are reasonably simple to use. They can be scheduled to run on different groups of targets at different time

<sup>46</sup> <http://www.openvas.org/>

frames, and most integrate with the major SIEM solutions. This allows the SIEM solutions to handle storing the results of regular scans, as well as alerting administrators or triggering an automatic quarantine via an automated incident response solution.

## **Support Calendars**

The final piece of advice in this security book will be to create and maintain a support calendar. The goal of support calendars is to mark end-of-support dates for all hardware and software in use, ensuring that nothing goes out of support without relevant individuals being made aware.

Asset detection and management solutions are often useful, as they typically pull whatever support information is detectable. Unfortunately, not all devices, OSEs, or applications list their support information. In addition, not all vendors actually provide support for the entire stated lifetime of a product, meaning that a great deal of IT can exist within an organization that's supposed to be under support, but which is vulnerable to known attacks, with no expected patch date.

Support calendars are thus something of an industry euphemism, as they often contain not only end-of-support information for individual IT assets, but also an organization's vendor blacklist. In many cases this is also the location of the secret squirrel contact information to get hold of the one human being at a vendor that can actually accomplish things.

## **No Plan Is Perfect**

IT security is a balance between the pragmatic desire to rigidly specify everything, and the very human desire to operate without constraints. IT security that's not accepted by a user base will be ignored, or worse, actively circumvented. As a result, no IT security

plan can ever be perfect, in design or practice; but that isn't reason not to try.

Every layer of IT security we can implement contributes to a defense in depth approach. Every organization that secures something contributes to the security of the whole, raising the cost for attackers that extra little bit. All those little bits count.

None of us can ever know everything, plan for everything, or prevent every attack. We each must do the best that we can, and the rest is up to our backups. And if you haven't tested your backups recently, now would be a good time.

# About the Author



## **Trevor Pott**

Trevor Pott is a hands-on technologist and cofounder of eGeek Consulting Ltd. He is the rare individual who can understand and operate almost any technology, but is also able to communicate the how, what, and why of a product to just about anyone. He uses this unique blend of experiences and talents to help technology vendors fine-tune their message to both business and technical buyers.