Security Hygiene:
The First Line of Security

Version 1.3
Released: May 2021

Securosis, L.L.C. 515 E. Carefree Highway Suite #766 Phoenix, AZ 85085 **T** 602-412-3051 info@securosis.com www.securosis.com

Author's Note

The content in this report was developed independently of any sponsors. It is based on material originally posted on the Securosis blog, but has been enhanced, reviewed, and professionally edited.

Special thanks to Chris Pepper for editing and content support.

Copyright

Security Hygiene

Table of Contents

Why Security Hygiene is Critical for Protection

After many decades as security professionals, it's depressing to keep seeing the same issues and mistakes. It feels like we're stuck in hacker Groundhog Day. Get up, clean up the mistakes made by users or administrators, handle a new attack, and fill out compliance reports, only to have to do it all over again the next day. Of course, we live in an asymmetric world when it comes to security. Attackers only have to be right once, and they might be able to get right into your environment. Defenders just have to be wrong once — a fact that attackers exploit to gain significant footholds. It's not fair, but nobody said life was fair.

The most basic advice we give anyone building a security program is to make sure you handle the fundamentals well.

The most basic advice we give anyone building a security program is to make sure you handle the fundamentals well. You remember security fundamentals, right? Visibility for every asset. Maintain a strong security configuration and posture for those assets. Patch systems efficiently and effectively when vendors issue updates. Most practitioners nod their heads about the fundamentals and then spend all day figuring out how the latest malware off the adversary assembly line works, burning a couple of days threat hunting in their environment, or just blissfully assuming that it won't happen to them. You know, the fun stuff. The fundamentals are just… boring.

But the fact is that fundamentals work. Maybe not for every attack but a lot of them — which is how they got to be fundamental in the first place. We'll offer a reminder in this paper. We can't eliminate all the risks, but shame on us if we aren't making it harder for adversaries to gain a foothold in your environment. It's about closing the paths of least resistance, ensuring that unworthy opponents like script kiddies and run-of-the-mill hackers do not bring you down. If we can make adversaries work to compromise your environment, they will likely mess up, triggering detection or leaving behind evidence for a future investigation.

Many Bad Days

As we review the lists of thousands of breaches over the years, quite a few resulted from misconfigurations, not fixing known vulnerabilities, or failure to install vendor patches. Let's dig into three specific breaches to get a good flavor for the downside of failing security hygiene.

- **Microsoft Exchange:** The most recent high-profile breach involved an attack on Exchange servers installed on-premises and resulted in attackers gaining full access to the servers. A scourge of ransomware attacks followed, highlighting the need to keep these critical components up to date.
- **Equifax:** This company left Internet-facing servers vulnerable to the Apache Struts attack unpatched, allowing remote code execution. The patch was available from Apache, but the company didn't apply it to all systems. Even worse, their Ops team checked for unpatched systems and didn't find any, even though they still had vulnerable systems. It was a textbook hygiene fail, resulting in hundreds of millions of user identities stolen. Equifax ended up paying hundreds of millions of dollars to settle their liability. That's a lousy day.
- **Citrix:** When a major technology component is updated, you should apply the patch. It's not like attackers don't reverse-engineer patches to find the vulnerabilities they address. This situation was particularly problematic in the Citrix hack in early 2020 because the attackers could perform automated searches to find vulnerable devices. So, of course they did. The initial mitigations suggested by Citrix, instead of a patch, were neither reliable nor implemented widely within their customer base, leaving many organizations exposed. At the same time, widely distributed exploit code made exploitation easy. Once Citrix did issue patches, customers quickly adopted them and essentially shut down the attack. Patching works, but only if you actually do it.

We don't like picking on companies who have suffered breaches, but we need to learn from their mistakes. And infrastructure hygiene is only getting more complicated. The SolarWinds attack in late 2020 was an example where even doing the right thing and patching the tool ended up providing access to attackers. If you looked at that situation in isolation you might ask, "Why bother patching?"

That question indicates you learned the wrong lesson. Go back up a few paragraphs, where it says, "You can't eliminate every risk." Supply chain attacks happen, and candidly, you can't do much about them besides focusing on detection and monitoring. But not patching a component opens up your systems to anyone with the exploit.

Not an Option

After all the downsides above, let's say you are still resistant to practicing good infrastructure hygiene. Don't take it from us — listen to your auditor, who will find (and report) all sorts of deficiencies if you can't keep things configured strongly and patched. Let's highlight a few regulatory mandates which call for patching.

- **PCI:** Requirements 2, 6, and 11 mention patching.
- **ISO 27001:** Control A.12.6.1 deals with remediating vulnerabilities (patching).
- **GDPR Article 25:** Data Protection by Design and Default and Article 32: Security of Processing allude to the need to have systems that protect customer data, and if the systems don't have good hygiene, they cannot protect said customer data.

> We don't like picking on companies who have suffered breaches, but we need to learn from their mistakes.

- **NIST SP 800-53 R3:** Configuration Management (CM), Risk Assessment (RA), and System and Information Integrity (SI) all highlight the need to patch infrastructure.

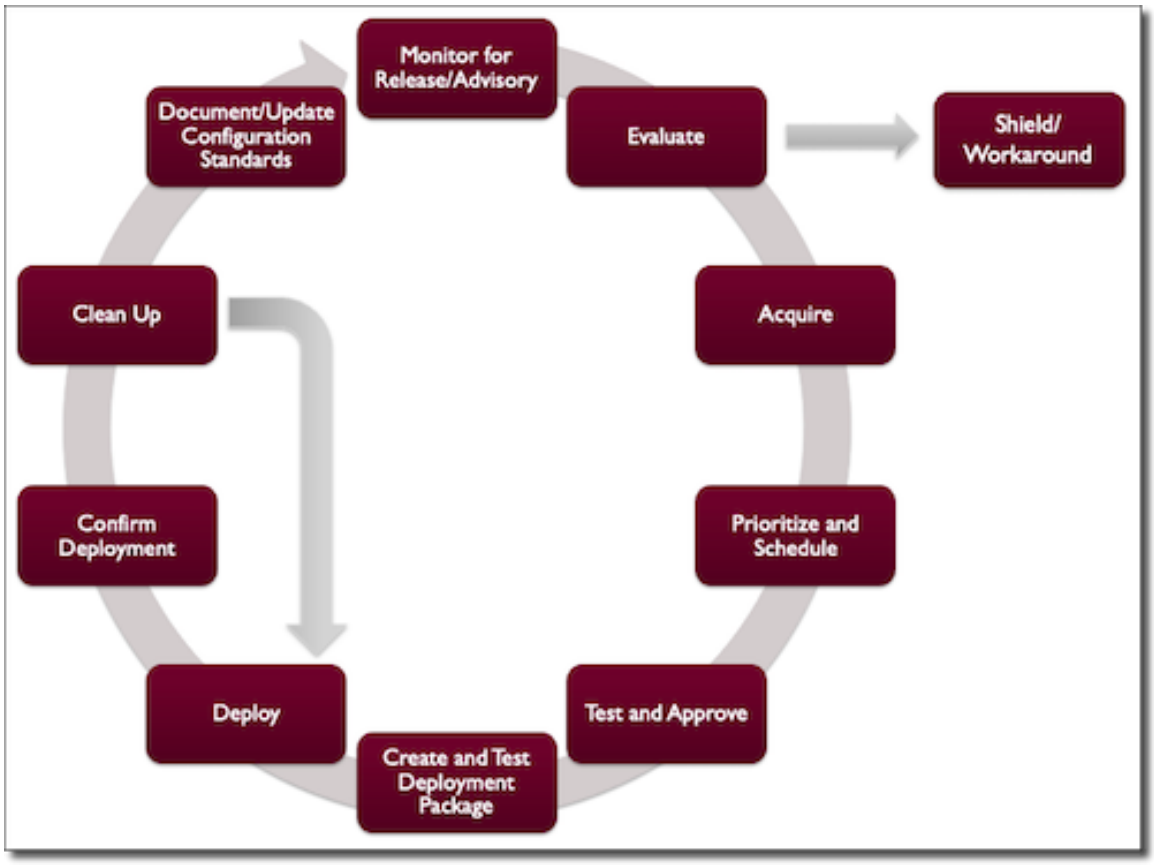
You don't have an option, do you?

Fixing Vulnerabilities

As mentioned above, the most basic advice we can give on security is to do the fundamentals well. That doesn't insulate you from determined and well-funded adversaries, but it will eliminate the paths of least resistance most attackers take.

As if that weren't enough, now you have essentially zero margin for error because attackers have automated many attacks' reconnaissance. Thus, if you leave anything exposed, they will find it. They have bots and scripts constantly searching for weak links.

All that said, you aren't reading this paper to keep hearing about the challenges of doing security, are you? Let's shift our focus to how to fix issues.



Fix It Fast and Completely

The vendors themselves issue updates and patches for vulnerabilities discovered in their products. Customers then patch their systems so they're up to date and secure. We've been patching as an industry for a long time. And we at Securosis have been researching patching for almost as long. Feel free to jump in the time machine and check out our seminal work on patching in the original Project Quant.

The picture above shows the detailed patching process we defined back in 2009. You need a reliable and consistent process to patch effectively. We'll point specifically to the importance of the Test and Approve step due to the severe downside of deploying a patch that takes down a component.

Yet working through a robust patching process can take anywhere from a couple of days to a month. Many larger enterprises expect to have patches deployed within a month of release. But in reality, a few weeks may be far too long to apply a high-profile patch for an issue that is being actively exploited. You need a high-priority patching process for patches that address very high-risk vulnerabilities. An essential requirement is to establish and agree on criteria for triggering the high priority (out-of-cycle) patching effort and which parts of the normal patching process you will skip.

Alternatively, you could use a temporary virtual patch, which attempts to protect against exploitation of an unpatched vulnerability based on the attack's signature. This is only possible if the attack has an established identifiable pattern to build the signature, and even then, signatures provide inadequate attack surface protection. On the bright side, the deployment of virtual patches can be rapid. However, given that information about new high-risk vulnerabilities is often thin and vague, it is difficult, if not impossible, to ensure that any workaround has correctly keyed into the optimal identifiable pattern or that the attack won't morph to change its signature. It all comes down to how perfectly the vendor providing the temporary virtual patch can identify the pattern. If that falls short, a large percentage of exploits may still get through. And if the vendor overshoots the pattern, you will break your application by blocking legitimate transactions. Further, you still need to run numerous tests to ensure nothing breaks, and you may not save much time over patching anyway. That's a lot of considerations, which amount to real risks.

One of the other downsides to temporary virtual patches is that all traffic destined for the vulnerable component needs to run through the inspection point or mitigation logic. If traffic can reach the component directly, the temporary virtual patch is useless. For instance, if a virtual patch is deployed on a perimeter security device to protect a database, an insider with direct access to the database can bypass the patch to exploit the unpatched database. In this context, an 'insider' might simply be an adversary with a foothold inside the perimeter or with a set of administrator credentials.

You also need to clearly understand what to do after an attack pattern is recognized. If you kill the entire connection, that could impact many other users who use the same connection pool or affect application behavior in different unanticipated ways.

For high-priority vulnerabilities that you cannot patch immediately, either because the patch isn't available yet or due to downtime or other maintenance challenges, a temporary virtual patch can provide an important short-term alternative. But remember that you aren't fixing the component — you're hiding it behind the virtual patch. With 30 years of experience under our belts, we can definitively tell you that hoping an attacker doesn't find your vulnerable systems is not a path to success.

Given the ease with which adversaries can change their attack signature to evade detection and temporary virtual patches, the fact that it's impossible to come up with a perfect detection signature, and the difficulty in ensuring that all traffic goes through an inspection point, deploying a vendor patch is the only long-term solution. Speaking of long-term solutions…

Take Full Advantage of Shared Responsibilities

One of the most compelling things about the cloud revolution is the idea of replacing some infrastructure components with platform services (PaaS). We alluded to this above, so let's dig a bit deeper into how shared responsibility can improve infrastructure hygiene. First, shared responsibility is foundational to cloud computing; each cloud provider takes on specific responsibilities. The cloud consumer (you) has interlocking security responsibilities. The combination is shared responsibility.

Ultimately moving some responsibility to a service provider reduces both your attack and operational surface area, and that's a good thing. Over the long term, strategically using PaaS services is one of the better ways to reduce technology stack risk.

The specific division of responsibility depends on the service and the delivery model (SaaS or PaaS) but suffice it to say that embracing a PaaS service for an infrastructure component gets you out of the operations business for that component. You don't need to worry about scaling or maintenance, including security patches. I'm sure you'll miss the long nights and weekends away from your family, running hot fixes on servers and databases.

Ultimately moving some responsibility to a service provider reduces both your attack and operational surface area, and that's a good thing. Over the long term, strategically using PaaS services is one of the better ways to reduce technology stack risk. Indeed, a service provider can still make a mistake, but the risk is considerably less. Service providers have their reputation and brand equity to worry about and dedicate considerable resources to addressing vulnerabilities and keeping customers safe.

The Supply Chain

If there is anything we've learned from the recent SolarWinds breach and the Target attack (from 2013), which started with a breach of a third-party contractor, it's that hygiene responsibilities don't end at the boundaries of your environment. As mentioned above, you may not be responsible for maintaining your providers' and partners' infrastructure components, but you are accountable for how their weaknesses can impact your environment.

Wait, what? Let's clarify a bit. If an external business partner gets compromised and the attacker moves into your environment and starts wreaking havoc, guess what? You are accountable for that. Sure, you can make the case that the partner was responsible for protecting their environment and failed. But that won't help when you are in front of your organization's audit committee explaining why your third-party risk program wasn't good enough.

Just as we want to leverage the shared responsibility model to get operational help and reduce the attack surface, you need to spend additional resources on risk management to understand the importance of what is at risk to choose an appropriate remediation approach.

Success and Consistency

We should reiterate that none of the approaches to infrastructure security hygiene are mutually exclusive. A patch eliminates vulnerability on a component, but there are cases where a temporary virtual patch can temporarily, reduce an immediate risk. The best long-term solution will always involve patches provided directly from the vendor and may include moving to a PaaS service. You'll need to figure out the best approach on a case-by-case basis, balancing risk, availability, and your ability to refactor the application.

Get a Quick Win

High-priority vulnerabilities happen all the time, and how you deal with them typically determines how people perceive the capability and competence of the security team. In this scenario, let's consider a small financial services organization, maybe a regional bank. They use a home-grown client/server application to handle customer loan data, and it uses stored procedures heavily for back-end processing. The application team periodically updates the front-end web interface, but the back end has largely remained unchanged. It's a common situation: if it's not broken, don't fix it; the application seems modern to customers (who use the web interface), and the back end works well enough. But every so often, they get a vendor alert regarding a high-profile vulnerability impacting the back-end database, warning of the imminent release of a patch. As such, the security team must figure out the best and most secure path forward.

High-priority vulnerabilities happen all the time, and how you deal with them typically determines how people perceive the capability and competence of the security team.

The first step in our process is risk analysis. Based on a quick review of threat data, there is an exploit in the wild, which means doing nothing is not an option, and time is critical. Next, we need a sense of the application's importance. It's described above as holding customer loan data, so it's essential to the business and within the scope of the bank's regulatory oversight. Because application usage typically occurs during business hours, a patch can be applied after hours.

A quick fix is needed because the exploit is in the wild, as security researchers have indicated that specific queries can provide access to the database. The security team implements a temporary virtual patch using a perimeter IPS device, inspecting and blocking the particular dangerous queries. The attack parameters are often too broad for IPS blocking, but in this case, the temporary patch was workable.

As another precaution, the team increases monitoring around the database to alert them to any insider activity, which would evade the temporary virtual patch. Additional monitoring will detect an adversary bypassing the application using an already compromised device to make the dangerous queries directly against the database.

The operations team then needs to apply the vendor patch during their next maintenance window. The temporary virtual patch bought the team some time to test the vendor patch to ensure it doesn't impact the application. The vendor patch test showed no adverse impact, and the operations team successfully applied it during the next window.

The last step involves a strategic review of the process to identify improvement for next time. At some point, the application will be refactored and moved into the bank's cloud presence, but not for 24 months. Does it make sense to increase the priority? Probably not — even if the next vulnerability doesn't lend itself to mitigation using a temporary virtual patch, remediation can occur via an off-hours emergency update without significant impact to application availability. As refactoring the application begins, teams will consider initially moving some stored procedures to an app server tier and later migrating the data to PaaS to reduce both the application's attack and operational surface. They also should consider whether a commercial SaaS offering can replace the application altogether.

Organizational Alignment

The scenario above shows how all the options for infrastructure hygiene can play together to mitigate the risk of a high-priority database vulnerability effectively. Several teams were involved in the process; starting when Security identified the issue, worked through the remediation alternatives, and decided on a temporary virtual patch and additional monitoring. The IT Ops team played an essential role in managing the vendor patch testing and application. The architecture team will consider refactoring the application or migrating to a SaaS offering.

To work together effectively, all these teams need to align and collaborate to ensure the desired outcome: application availability with no loss of data. However, we should mention another team with a crucial role in facilitating the process: Finance. They pay for items such as a perimeter device that can provide a workaround and a support/maintenance agreement to ensure access to patches, especially for easily forgotten legacy applications. As critical as technical skills remain to keep infrastructure in top shape, ensuring the technical folks have the resources to do their jobs is just as important.

If you have any questions on this topic, or want to discuss your situation specifically, feel free to send us a note at info@securosis.com.

About the Analyst

Mike Rothman, Analyst and President

Mike's bold perspectives and irreverent style are invaluable as companies determine effective strategies to grapple with the dynamic security threatscape. Mike specializes in the sexy aspects of security — such as protecting networks and endpoints, security management, and compliance. After 20 years in and around security, he's one of the guys who "knows where the bodies are buried" in the space.

Starting his career as a programmer and networking consultant, Mike was an analyst at META Group prior to founding SHYM Technology, and then held executive roles at CipherTrust and TruSecure. Mike then started Security Incite in 2006 to provide a voice of reason in an over-hyped yet underwhelming security industry. After taking a short detour as Senior VP, Strategy at eIQnetworks, Mike joined Securosis with a rejuvenated cynicism about the state of security.

Mike published The Pragmatic CSO <http://www.pragmaticcso.com/> in 2007 to introduce technically oriented security professionals to the nuances of what is required to be a senior security professional. He also possesses a very expensive engineering degree in Operations Research and Industrial Engineering from Cornell University. His folks are overjoyed that he uses literally zero percent of his education on a daily basis.

About Securosis

Securosis, LLC is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services. Our services include:

- **Primary research publishing**: We publish the vast majority of our research for free through our blog, and package the research as papers that can be licensed for distribution on an annual basis. All published materials and presentations meet our strict objectivity requirements, and follow our Totally Transparent Research policy.
- **Cloud Security Project Accelerators**: Securosis Project Accelerators (SPA) are packaged consulting offerings to bring our applied research and battle-tested field experiences to your cloud deployments. These in-depth programs combine assessment, tailored workshops, and ongoing support to ensure you can secure your cloud projects better and faster. They are designed to cut months or years off your projects while integrating leading-edge cloud security practices into your existing operations.
- **Cloud Security Training:** We are the team that built the Cloud Security Alliance CCSK training class and our own Advanced Cloud Security and Applied SecDevOps program. Attend one of our public classes or bring us in for a private, customized experience.
- **Advisory services for vendors**: We offer a number of advisory services to help our vendor clients bring the right product/service to market in the right way to hit on critical market requirements. Securosis is known for telling our clients what they NEED to hear, not what they want to hear. Clients typically start with a strategy day engagement, and then can engage with us on a retainer basis for ongoing support. Services available as part of our advisory services include market and product analysis and strategy, technology roadmap guidance, competitive strategies, etc. Though keep in mind, we maintain our strict objectivity and confidentiality requirements on all engagements.
- **Custom Research, Speaking and Advisory**: Need a custom research report on a new technology or security issue? A highly-rated speaker for an internal or public security event? An outside expert for a merger or acquisition due diligence? An expert to evaluate your security strategy, identify gaps, and build a roadmap forward? These defined projects bridge the gap when you need more than a strategy day but less than a long-term consulting engagement.

Our clients range from stealth startups to some of the best known technology vendors and end users. Clients include large financial institutions, institutional investors, mid-sized enterprises, and major security vendors. For more information about Securosis, visit our website: <http://securosis.com/>.