



Security

Updates, Threats, and Risk Management

Revisiting a recent column considering security updates.

THE PREVIOUS *COMMUNICATIONS* Security column (January 2023), by Fabio Massacci and Giorgio di Tizio,⁹ used an evaluation of data about “Advanced Persistent Threats” (APTs) to defend the proposition that rapid deployment of security updates is largely ineffective and probably unnecessary as a security measure for most organizations. The data and analysis supporting those claims are drawn from the authors’ longer paper in the *IEEE Transactions on Software Engineering*.² The authors also claim security updating would be entirely unnecessary if software vendors and development organizations could be held liable for the consequences of any security vulnerabilities included in their products.

We believe the authors reported on research that is challenging and has received little rigorous analysis over the years. The paper and column raise questions that are relevant and difficult to answer quantitatively. However, given the current state of security updating and secure development, we found the column could be read as advocating and justifying decisions that would increase real-world risk to IT systems. This column addresses these issues by reviewing the definition and application of the term APT, the authors’ data and position on updating, and advocates a different path before a discussion of liability.

Advanced Persistent Threats

Massacci and di Tizio use a very broad definition of Advanced Persistent



Threat: “A sophisticated group involved in malicious cyber activities.” There is obviously no measurable definition of “sophisticated”—in practice, it is often used to mean “attackers that are part of a nation-state or appear to be sponsored by a nation-state,” or sometimes groups that use a group name or have been given a group name (or number) by government agencies or commercial security services firms.

While references to APT are often associated with nation-state attackers, the MITRE ATT&CK framework on which the authors depend lists a variety

of attackers, some clearly nation-state (usually designated APTnn), some financially motivated such as FINnn and Carbanak, and, some unknown. More importantly, the APT designation focuses on the threat actor, rather than how or why the threat succeeded. As the Microsoft 2021 Digital Defense Report pointed out, the tools used by nation-states to compromise victim networks are most frequently the same tools used by other malicious actors.¹⁰

The qualification for being considered an APT group included in the authors’ dataset is far from precise. NIST

acm

Advertise with ACM!

Reach the innovators
and thought leaders
working at the
cutting edge
of computing
and information
technology through
ACM's magazines,
websites
and newsletters.



Request a media kit
with specifications
and pricing:

Ilia Rodriguez

+1 212-626-0686

acmm mediasales@acm.org

acm

media

Security updating is neither a perfect security measure nor a free one. But it is affordable and often effective.

has a very different and broader definition⁵ of APT: “The advanced persistent threat: pursues its objectives repeatedly over an extended period of time; adapts to defenders’ efforts to resist it; and is determined to maintain the level of interaction needed to execute its objectives.”

The NIST definition is much more meaningful. In the real world:

- ▶ Advanced: it got through our defenses
- ▶ Persistent: it took a long time for us to notice
- ▶ Threat: because it got through and because it took a long time for us to notice, it caused a lot of damage.

Since the goal of cybersecurity is always to reduce damage from future attacks, the most important issue is what vulnerability was exploited not what type of group launched the attack.

Updating as a High-Value Defense

The paper points out that many APTs do not exploit vulnerabilities to gain initial access to a target system. Neither the *Communications* column nor the TSE paper makes it clear whether the set of APT attacks described as not having used a vulnerability to gain initial access did or did not exploit vulnerabilities to persist access, to escalate privilege, to evade defenses, or to access credentials. However, the authors’ dataset makes it clear the APTs studied make plentiful use of product vulnerabilities. While blocking initial access is the ideal, blocking any of these later steps by installing an update is still a “win” for the defender and a loss for the APT. Organizations that are committed to security recognize this fact and are diligent about deploying updates.

Since the same attacks have different results against different organizations that have the same software and vulnerabilities, differences in IT and security operations are meaningful—not all attacks succeed, and many that gain initial access cause meaningful damage to some and not to others. Updating is part of an organization’s “defense in depth” that makes the attacker’s job harder.

The TSE paper starts out by saying: “A recent study⁹ shows that it takes more than 200 days for an enterprise to align 90% of their machines with the latest (not known to be vulnerable) software version given the need to perform regression testing.”²

There are several problems with this claim:

- ▶ There are dozens of other studies showing much shorter average time to patch. For example, the Infosec Institute cites between 60 and 150 days.¹¹

- ▶ As companies increasingly use software applications as a service, SaaS providers patch those applications much faster than the paper claims. The recent exploits of Exchange vulnerabilities impacted only customer premise, self-managed Exchange use, not Exchange as a Service offered by Microsoft.

- ▶ The assumption that regression testing takes a long time is mostly limited to an increasingly smaller amount of legacy, custom-built business applications. For example, the Google Chrome browser updates constantly and very few enterprises try to do regression testing before browser updates. Mobile devices and consumer desktop operating systems routinely install updates automatically on hundreds of millions or billions of devices without end-user regression testing.

The authors’ citation seems to have caused them to only consider patching

The APT designation focuses on the threat actor, rather than how or why the threat succeeded.

“immediately” versus monthly or longer. Organizations and SaaS providers are using DevOps approaches that are often rolling out bi-weekly, weekly, or even daily updates of applications—just because Microsoft still releases monthly Windows patches does not mean weekly updates cannot be done.

For example, the Shellshock BASH Shell vulnerability^{4,12} was disclosed in 2015 with a base CVSS score of 9.8 out of 10—highly critical. Ikea’s CIO reacted quickly⁷ and Ikea tested the patch and deployed it to 3,500 Linux servers in less than three hours. Ikea’s practice of requiring all applications to conform to the Red Hat Linux Application Binary Interface provided high isolation from kernel updates and their use of automated tools and accurate asset inventory made such rapid updating easy.

As the authors suggest, some organizations conduct risk assessments to decide whether to deploy an update or how quickly to do so. But risk evaluation can overlook an attack vector and monitoring can be both costly and ineffective. As the Ikea experience illustrates, deploying updates is affordable and effective. And the reduced attack surface reduces the cost and increases the effectiveness of monitoring for attacks that do not exploit vulnerabilities.

While rapid updating is not a cure-all that protects organizations from APT (or other) attacks, we believe updating is often effective, as demonstrated by the paper’s statistics on attackers’ use of vulnerabilities. And the experience of SaaS vendors, consumers, and enterprises such as Ikea demonstrates that rapid updating is not as difficult or as risky as the authors suggest. In fact, we believe rapid updating is an important best practice for all technology users.

Liability

The January Security column does not defend its argument for developer liability beyond a claim that liability would eliminate the need for patching, and a reference to an article by Poul Henning-Kamp that does not defend liability either. Henning-Kamp’s article advocating product liability for software appeared in 2011.⁶

If the introduction of product liability for software were likely to eliminate the need for patching by eliminating vulnerabilities, it would be well worth consid-

We believe rapid updating is an important best practice for all technology users.

ering. Unfortunately, there is no reason to believe that would be the result.

For a selling party to assume liability for a defect found in a product, in the U.S. Uniform Commercial Code the product has to be considered “tangible”—and the UCC says software is still considered to be “a general intangible.”³ In an odd way, this actually makes sense. Engineering disciplines only exist in tangible areas, such as civil, chemical, mechanical, and electrical engineering where tables of materials strengths and properties can be created and regulations can be created around acceptable safety margins based on intended use. No such tables exist for software, and none have shown a sign of emerging over the past 20 years.

So, regulations to support developer liability are not on the horizon and even if they ever do emerge, it will not change the fact that software will continue to be no more tangible than speech, where to prove liability an intent to harm has to be proven.


Just as newspapers issue retractions every day for erroneous sentences, software will forever be issuing retractions for miscoded lines of code. Better fact checking and editing processes reduce the problem in the news industry, but retractions are still required. Better testing and development processes can do the same for software, but patching will still be required because bad guys will take advantage of any and all vulnerabilities.

The picture for product security is not bleak: there are best practices that all vendors should be applying, and while not perfect, they do make attackers’ jobs more difficult. The U.S. government’s Executive Order 14028¹ and the NIST Secure Software Development Framework¹⁴ provide guidance for using such practices, and put the force of

the government’s procurement power behind their implementation.

The Biden Administration in the U.S. released its National Cybersecurity Strategy¹³ in March, 2023. The strategy includes a proposal for a form of developer liability but also acknowledges that even the most advanced software security programs cannot prevent all vulnerabilities: even if liability is assumed by software vendors at some level, users will still be required to update their software.

Conclusion

Security updating is neither a perfect security measure nor a free one. But it is affordable and often effective. Organizations should not allow the Massacci and di Tizio column to deter them from continuing to apply security updates rapidly. And organizations should seek to adopt software products that have been developed using best practices as specified in the recent Executive Order and SSDF. 

References

1. Biden, J.R. Executive Order on Improving the Nation’s Cybersecurity. (May 12, 2021); <https://bit.ly/3FfDITM>
2. Di Tizio, G. A quantitative evaluation against advanced persistent threats. *IEEE Transactions on Software Engineering* (2022).
3. Drug and Device Law. New Decision Directly Addresses the “Is Software a Product” Question. (May 2, 2022); <https://bit.ly/3JrKzNE>
4. Information Technology Laboratory. National Vulnerability Database: Vulnerability Metrics. National Institute of Standards and Technology; <https://bit.ly/2IzbEfp>
5. Joint Task Force Transformation Initiative. Managing Information Security Risk, Organization, Mission, and Information System View. National Institute of Standards and Technology, Gaithersburg, MD, 2011.
6. Kamp, P.H. The software industry is the problem. *Commun. ACM* 54, 11 (Nov. 2011), 44–47.
7. Kerner, S.M. Ikea pPatched for Shellshock by methodically upgrading all servers. *eWeek* (June 28, 2015); <https://bit.ly/3ZE7b3p>
8. Klein, G. et al. seL4: Formal verification of an operating system kernel. *Commun. ACM* 53, 6 (June 2010), 107–115.
9. Massacci, F. and di Tizio, G. Are software updates useless against advanced persistent threats? *Commun. ACM* 66, 1 (Jan. 2023), 31–33.
10. Microsoft Corporation. Microsoft Digital Defense Report. (Oct. 2021); <https://bit.ly/3mBaaLn>
11. Morrow, S. Time to patch: Vulnerabilities exploited in under five minutes? (Aug. 2, 2021); <https://bit.ly/3Fdk4cP>
12. Souppaya, M. Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities. National Institute of Standards and Technology, Gaithersburg, MD, 2022.
13. U.S. National Cybersecurity Strategy. March 2023; <https://bit.ly/40d1biD>
14. Verizon. 2022 Data Breach Investigations Report. (May 24, 2022); <https://vz.to/3JufDNb>

Steve Lipner (lipner@outlook.com) is Executive Director of SAFECode in Wakefield, MA, USA.

John Pescatore (jpescatore@sans.org) is Director of Emerging Security Trends at the SANS Institute in Rockville, MD, USA.

Copyright held by authors.