Issue Brief

Will AI Make Cyber Swords or Shields?

Authors
Andrew J. Lohn
Krystal A. Jackson

CSET CENTER *for* SECURITY *and* EMERGING TECHNOLOGY

Aug 2022

Executive Summary

Cybersecurity is a constant battle between attackers and defenders who try to leverage advances in technology to gain an advantage. Progress in those technologies can tip the scales in favor of either offense or defense, and it is not always clear beforehand which side will benefit more. This report illustrates how mathematical modeling can provide insights into how advances in technology might affect a few areas of cybersecurity: 1) phishing, 2) vulnerability discovery, and 3) the race between patching and exploitation. We demonstrate the approach and show the types of insights that it can provide.

Phishing

Phishing is already a popular and effective technique for attackers. With little effort, attackers can send a generic message to many recipients, tricking a small percentage of them into cracking the door to the victim's organization. Attackers can work harder to tailor their message to individual targets and increase the probability of success. Today, automated systems that can collect private data and write convincingly threaten to combine the scale of those spray-and-pray phishing campaigns with the effectiveness of spear phishing.

For this report, we consider how sending many emails to an organization increases the chance of a breach but also increases the chances of being discovered by the organization's defenders. We find that even if applying artificial intelligence (AI) to the phishing process increases the odds of an employee falling for a phishing email, the attacker may choose to send few emails to avoid being detected—so few that humans could write them for themselves. That is especially true if phishing detection technologies improve as well. This means that organizations that have been targeted in the past may not experience drastic changes from automated phishing. Organizations that have so far been too low-profile to garner much interest from attackers may not be as fortunate, and may experience an increase in high-quality phishing attacks due to the availability of automated writing systems. If phishing campaigns target a larger number of these organizations, then sharing threat information about these offensive campaigns may become even more beneficial than it already is.

Vulnerability Discovery

In examining the vulnerability discovery process, we found that both computers and humans discover vulnerabilities at a rate that starts high then decreases over time.

When modeling this mathematically there are just two components: one that describes the initial rate of vulnerability discovery, and a second that describes how quickly that rate decreases. Modeling this activity suggests that testing quickly can uncover most or all of those vulnerabilities so they can be fixed, perhaps even before the software is released. However, for testers that continue to discover vulnerabilities with only modest decreases in the rate of discovery, the vulnerability discovery process continues for much longer. Testing faster may just mean that there are more vulnerabilities for attackers to use and for defenders to remediate. This suggests that techniques that simply accelerate vulnerability discovery are a benefit to defense, but techniques that are more creative and sustain the vulnerability discovery process over time—something more akin to current human processes—may actually hurt more than they help.

Race Between Patching and Exploitation

Once a new vulnerability is discovered, attackers and defenders are in a race to either patch the systems or exploit them. Patches need to be developed and also distributed to computers around the world before the exploits are developed and distributed. For this report, we draw on models that match the historical delays in these separate stages of the race.

Defenders usually get a head start, about 80 percent of the vulnerabilities have a patch ready on the day the vulnerability is disclosed. Even when they do not have that head start, patch development tends to be faster than exploit development. This head start and rapid development mean that there are only limited benefits to gain from further advances in patch development. On the other hand, deploying those patches is often much slower in practice for a variety of reasons, so advances that help users incorporate patches can significantly decrease the fraction of computers that are vulnerable to a given vulnerability at a given time. Our estimates suggest that a five times speedup in patch adoption would reduce the peak number of exposed vulnerabilities by about 25 percent and would decrease the number of exposed computers at the one-hundredth-day mark by 400 percent.

Conclusions Regarding Modeling Technology Development

We believe that models like these can help guide investment and research decisions in ways that prioritize technologies that are likely to have the most beneficial impact. As a few examples, AI for phishing may increase the need for information sharing, vulnerability discovery technologies should aim to increase speed of discovery but not creativity, and patch-deployment technology should be prioritized over patch development. However, these models are highly uncertain and we expect that their main benefit may not be in any specific recommendation, but rather in concretizing assumptions. We hope that being mathematically explicit can accelerate debate over which assumptions are most appropriate.

Introduction

Since the day a stray spark ignited a village rather than warm the night, humanity has had to reckon with the awful powers of its most beneficial creations. Today, the pace of progress leaves little time to weigh a technology's merits, and the role for new tools is rarely as clear-cut as it was for swords and shields. Society as a whole, and policymakers in particular, must buy time by anticipating what is coming before it arrives in order to prepare and to steer innovators toward applications that maximize benefits while minimizing harm.

One way to weigh the positives and negatives of specific technological advances makes use of a large body of mathematical and empirical models of cybersecurity. Those models are usually built to reflect the current status quo but they can also be used to give clues about what to expect in the future by adjusting their variables to match possible advances in AI. We illustrate this process for three aspects of cybersecurity: phishing, vulnerability discovery, and the race between patching and exploitation. Many detailed and complicated models exist to draw from, but we focus on very simple models to illustrate the process and for clarity. This report discusses that process and shows the types of insights that can be drawn from it. Further technical details and the underlying math are available in a companion report.[1] We also provide the programming code and data in an associated GitHub repository.[*]

The mathematical approach is not meant to give precise answers because the parameters (and even the mathematical structure of the models) are too uncertain. Rather, the approach requires assumptions to be stated clearly, which we believe can accelerate debates over which types of progress and outcomes are most reasonable. We hope that adding additional rigor to discussions that are usually primarily qualitative will help decision makers and security researchers narrow in on which scenarios are likely, and to allocate resources most appropriately.

[*] https://github.com/georgetown-cset/modeling_ai_for_cyber

Phishing

Autonomously-Generated Phishing Email

Tom,

I hope you're doing well! I wanted to remind you about the upcoming event we have at the food bank. We could really use your help!

The event is on Saturday, December 12th from 9am-12pm. We'll be helping sort and distribute food to people in need. It's a great way to give back to the community and make a difference in people's lives.

If you're interested in volunteering, please add your information to the list on our website. I hope you can join us!

Thanks,

Sara

A cautious recipient would not fall victim to this email unless they are this fictional Tom who knows this fictional Sara, in that case it could be very enticing. Learning to mimic a real Sara's writing style, learning that a real Tom volunteers at this food bank, and also that Sara works there may be hard for humans, but it is simple for computers. Computers are optimized for pulling elements out of the mountains of public, leaked, or stolen data, and the email above cost less than half a cent to write. It is just a matter of filling the underlined sections of the prompt below with data from the database, then AI text generators can do the rest.*

* This particular example was written by GPT-3 with only the prompt that is shown.

Prompt for Phishing Email

Tom has been volunteering with Capital Area Food Bank for two years. Sara works for Capital Area Food Bank and needs to convince him to add his info to the list at her website for their upcoming event. Her email described the event:

Tom,

This technology is particularly worrisome because historically only a select few high-profile victims have been targeted with this quality and specificity, but soon it may be unleashed on the masses.[2] A flood of targeted emails like these provide attackers with many advantages but it also introduces risks. More malicious emails provide the defender with a greater opportunity to notice the attack and start quarantine and cleanup. Therefore, the attacker may try to send just enough emails to maximize their chance of gaining access to the network without alerting the victim's defenders.

To model this scenario, we assume that the attacker's goal is to compromise at least one account anywhere in the targeted organization without alerting defenders. We also assume that if the defenders are tipped off to any of the phishing messages, they can use the information in that message (such as the payload or header or outbound links) to find all the other phishing messages in the campaign. With those assumptions in place, the model is very simple.[*] The inputs are 1) the click rate among possible victims, 2) the probability that automated methods will report a given phishing email to the defense team, and 3) the probability that a human will report a given phishing email to the defense team. We then make two calculations: 1) the probability that a campaign will successfully phish at least one victim in the organization without alerting defenders, and 2) the optimal number of messages to maximize that probability.

As a baseline for the absence of AI, we use a click rate of 3 percent, which is the current estimated median click rate across organizations.[3] Click rates grow closer to 30 percent for carefully tailored emails like those used in spear-phishing campaigns.[4] For this study we imagine AI systems that can be as capable as human writers, so we use that 30 percent for the AI-generated click rate. Table 1 provides the rest of the parameters as well as the outputs of the model.

[*] Specifically, the equation is $P_{\text{undetected infection}} = P_{\text{infection}}P_{\text{no alert}}$ where $P_{\text{infection}} = 1-(1-P_{\text{click}})^N$ and $P_{\text{no alert}} = (1 - P_{\text{human alert}} - P_{\text{machine alert}} + P_{\text{human alert}}P_{\text{machine alert}})^N$.

Table 1: Parameters and Outcomes for Phishing Campaigns

Model Configuration	Click Rate	Computer Alert Rate	Human Alert Rate	Undetected Intrusion Probability	Optimal Message Count
Human-Generated Emails	3%	1%	1.5%	28%	26
AI-Generated Emails	30%	1%	1.5%	84%	9
AI-Generated Emails vs. Improved Detectors	30%	25%	0.5%	28%	2

Source: Andrew J Lohn and Krystal Alex Jackson, "Will AI Make Cyber Swords or Shields: A few mathematical models of technological progress."

The baseline without AI enhancements works out to a 28 percent chance of a successful and undetected phishing campaign, and the attackers should send 26 messages to maximize their probability. If the AI can write as convincingly as some spear-phishing campaigns, this raises the click rate to 30 percent and the probability of a successful and undetected campaign jumps to 84 percent with only nine messages. That probability of undetected success drops back to 28 percent if the automated alert system also improves enough to detect phishing 25 percent of the time. In that case, the optimal number of messages is only two.

Given those results, the threat of computers that can automate phishing increases an attacker's odds substantially. On the other hand, the optimal number of messages in the automated campaigns is low enough that humans could write them. Automated phishing increases the odds of success but decreases the number of messages that attackers need to write. If AI-assisted automated detection improves, it will decrease the optimal number of attacker messages even further and bring down their odds of success.

For defenders at organizations that are already frequently targeted, this means that there is reason to think automated phishing will not change the threat significantly. AI-enabled systems that can write many messages will in turn provide more opportunities to catch the intruders. For defenders at organizations that were previously less targeted though, it may have a bigger impact. Attackers will likely be able to automate campaigns against many organizations rather than just a few, expanding the number of targets that are economically feasible to attack. The prospect that one attacker may target many defenders creates additional incentives for those defenders to work together. Defenders have an advantage in numbers if they can rapidly share details like the addresses of malicious websites or information from the email headers such that all organizations benefit from phishing detection at one.

There may be little hope for stopping attackers from leveraging AI-generated phishing capabilities. This suggests that, in addition to improved information sharing, the biggest gains for defenders can be found by improving automated detection capabilities. Along these lines, the model indicates that even modest advances in automated phishing-detection technology could be a double benefit to defenders. It reduces an attacker's odds of success for a campaign and it may also reduce the number of emails in a campaign. If an attacker is concerned about their emails being flagged as phishing, then improvements in detection could drive them to send fewer emails, perhaps even to such an extent that they could just as easily write them manually.

Vulnerability Discovery

As an alternative way to access a network, or to advance the attack beyond the initial intrusion, attackers may use a vulnerability in the victim's system to advance their attack. These vulnerabilities are bugs that attackers may exploit to allow them to access and control a target's computers and networks. Many of the best and brightest on both offense and defense spend their days searching for these flaws to either exploit or patch. And most of the biggest headlines and hand-wringing happens when they uncover a new one, but they are not alone in the search—automated methods are already an important component of that search. In this section, we consider how that search might be different if those automated methods could better mimic human intelligence, and fortunately there is data to help understand those differences.

Vulnerability researchers have some automated techniques for finding vulnerabilities such as symbolic execution or fuzzing.[5] In general terms, symbolic execution analyzes the code in an attempt to determine which inputs will cause a crash while fuzzing simply floods a system or program with various combinations of inputs in an attempt to crash it. They are standard and indispensable techniques, but machines today are no match for humans. The Defense Advanced Research Projects Agency's Cyber Grand

Challenge aspired to close that gap, and China's vulnerability discovery competitions, the Robot Hacking Games, have continued trying to push that frontier.[6]

From a modeling perspective, machines are similar to groups of humans—their discovery rates have both been shown to follow what is called a power-law distribution.[7] That means that their ability to find new vulnerabilities decreases over time but that some vulnerabilities can still be discovered after long periods of time. This is true for both the basic fuzzing approach and for groups of humans, but to different degrees. Precisely how many vulnerabilities they find, and how quickly their discovery rate decays, depend on the power law's two parameters: the coefficient and the exponent.

The coefficient sets the initial rate of vulnerability discoveries, and the exponent sets how rapidly that rate decreases. These parameters depend both on which software is being inspected and the capability of the inspector. If the software is full of vulnerabilities or if the discoverer is very effective, then it will have a large coefficient and vulnerabilities will be discovered quickly. If the vulnerabilities quickly become hard to find or if the discoverer only knows a few techniques for finding them, then the power law would have a large exponent and the rate of discovery decays quickly. Put differently, a large coefficient means the tester finds vulnerabilities quickly, and a small exponent means that the tester can continue finding new ones for a long time.

Based upon prior research, basic fuzzing's ability to find new vulnerabilities declines quickly. That corresponds to power laws that have high exponents in the range of two to four—we use the value of three in this analysis.[8] Humans, who continue to find new vulnerabilities for a longer time, have a smaller exponent in their power laws. For a collection of humans all working separately but grouping their findings together, the exponent appears to be about 0.4.[9] This is a big difference because power laws have peculiar behavior when the exponent crosses the value of one. For exponents greater than one, we can calculate the total number of vulnerabilities that the technique is able to find. Running the technique faster means that all of those vulnerabilities can be found and fixed faster. When the exponent gets below one, that is not true. Below that threshold, the tester can keep finding vulnerabilities, so running it faster could create a constant flow of new vulnerabilities. Mathematically, there is no way of knowing when, or if, it will stop finding vulnerabilities for a sufficiently complex computer program.

There are cases where attacks on computer systems can be beneficial, but presuming those are the exception rather than the rule, funders and researchers should be cautious about developing technologies that are better at vulnerability discovery. If the new systems simply operate faster, then they are likely to be beneficial to society. Faster discovery rates can help software vendors discover and fix more of the vulnerabilities before the product goes live. If, on the other hand, discovery systems get better at continuing to find new vulnerabilities, then they might cause more harm than good. Those systems might continue finding vulnerabilities long after the software has gone live, creating a larger pool of vulnerabilities for attackers to use and for defenders to remediate. Whether these vulnerabilities can be used to exploit vulnerable systems is partly addressed in the next section.

Patching vs. Exploitation

A flood of new vulnerabilities would strain an already overworked cyber defense workforce, and technology can either help it or make matters worse. Defenders are in a race every time a new vulnerability is discovered. Sometimes they have a head start and sometimes they have to catch up. However, the head starts are often short-lived because providing a patch can bring attackers into the race by allowing them to compare old and new versions to find vulnerabilities.

Log4shell is a particularly high stakes example of this race because of its ubiquity and ease of use.[10] Fortunately, defenders were given a head start when a security researcher informed Apache of the flaw on November 24, 2021. That meant Apache already had a patch ready when they announced it to the world on December 9th.[11] That head start, along with security teams working around the clock, has helped to keep the damages below the worst initial fears.[12] Still, four months later, the internet was littered with vulnerable systems and Log4shell is likely to be a problem for years to come.[13]

In general, systems are at risk when defenders are slower to develop and distribute patches than attackers are to develop and distribute exploits. For defenders, both developing and distributing can cause delays. Though an important difference is that while the development delay for any given vulnerability is the same for all potential victims, their distribution delays can be different. Many organizations and software force automatic updates that can be very prompt, but some organizations cannot afford the downtime that it would take to incorporate the new safer code, or they cannot risk buggy patches that might hamper performance or affect other systems. So, there will almost always be some fraction of computers that are vulnerable to some fraction of vulnerabilities.

Whether attackers have the code to exploit those vulnerable systems depends on how quickly attackers write their own exploits. In principle it also depends on how quickly attackers can share and spread proven exploits, but for this work we assume that malware can spread fast enough that it does not add a meaningful delay.

There have been a few studies of the delays for each of these stages.[14] Whether the exploits are written by a private development company, or just collected in a public repository, does not seem to make a big difference.[15] In both cases, the same mathematical function matches their development timelines, and about half of the vulnerabilities had corresponding exploits within 50 days for both cases.

Turning to defense, about 80 percent of vulnerabilities have a patch before they are disclosed.[16] Even when patch developers do not have a head start, they appear to move a little quicker than attackers. Of the remaining bugs, about 80 percent have a patch within the next two months.[17] Deploying those patches on the other hand, is a bit slower. After one hundred days, only about half of all machines are patched against half of all vulnerabilities.[18] Patch adoption has been getting faster over the years—a testament to the value of approaches such as automatic updates—but it is clearly still the long pole in this tent for many organizations and systems.

In the technical companion to this paper, we fit equations to this real-world data then perform the math to combine them together to calculate the probability that a computer is both vulnerable and that attackers have the code to exploit it.[19] We can then explore how much benefit or harm might come from dialing up patch-deployment efficiency or accelerating development for patches or exploits.

We find that accelerating patch writing provides only small benefits, even when developers do not have a head start. On the other hand, accelerating patch adoption technology, once a patch is available, has a profound effect. Our estimates suggest that a five times speedup in patch adoption would reduce the peak number of exposed vulnerabilities by about 25 percent and would decrease the number of exposed computers at the one-hundredth-day mark by 400 percent. For comparison, even completely eliminating the patch development delay would only reduce the peak exposed fraction by about 13 percent (half as much) and would only decrease exposure at the one-hundredth-day mark by about 20 percent (twenty times less).

Immediate exploit development also has the potential to dramatically change the security landscape, at least among those vulnerabilities where defenders do not have a head start. Naturally, if exploits are developed very quickly but patching remains slow, then the fraction of exposed computers starts high. Over time, patching catches up, and the exposed fraction becomes more similar to what it would have been without automated exploit writers. Even after one hundred days though, the exposed fraction is still 60 percent higher than the baseline case without automated exploit writers. Frustratingly, the story does not improve much if patch writing advances just as quickly as exploit writing such that they are both effectively instantaneous. That is because deploying the patches is still slow. The models suggest that by the one-hundredth-day mark, 30 percent more systems would be exposed if both exploits and patches were written instantaneously than would be exposed today without those technologies.

It may seem natural for patch writing and exploit writing to be comparable technologies but they are different in important ways. System administrators need to be sure that a patch will not adversely affect their systems.[20] An unreliable patch can be as bad as an attack on the system, so defenders are hesitant and slow to deploy them. Even if technology improves at the same rate for writing both patches and exploits, the advantage goes to attackers because patch deployment is currently the long pole. As a result of being the primary bottleneck, even comparably modest advancements in patch-deployment technology stand to make substantial contributions to security. Fortunately, that can include a wide range of technologies, not just automated means for incorporating new code. For instance, advances in methods for testing patches to verify their reliability could help give defenders the confidence to update faster. Many systems already patch quickly, such as through automatic updates or in managed cloud environments, and the average timelines are already shrinking, but there is still plenty of opportunity for further acceleration.

Conclusion

Cyber defenders seek to arm themselves with the best technology available. However, their resources (time, money and expertise) are limited. Therefore, it is important to prioritize the technologies and the practices that are most likely to maximize success. For their part, developers and members of the scientific community should seek to pursue technology that is most beneficial. Policymakers and organization leaders also play a role by creating incentives that guide researchers. None of this is easy and the job is filled with unknowns and uncertainty, but we hope that the mathematical treatment summarized here can shed some light on a few areas of cybersecurity to help steer investment and development efforts in a positive direction.

For one, it is likely that automated text generation will improve phishing campaigns but it is unlikely to be the scourge that the most pessimistic futurists envision. One of the main reasons for this is because human-written campaigns are already so effective and efficient. Vulnerability discovery tools on the other hand are double edged: tools that are simply faster will likely improve security, but tools that can continue to discover vulnerabilities for a longer amount of time may flood the market with them, causing more harm than good. The ability to write patches autonomously would be a beneficial defensive technology, but it is unlikely to reshape the defensive landscape, even if it is completely successful. That is because deploying those patches remains a bottleneck. Patch-deployment tools on the other hand are a clear security boon, so research should prioritize finding ways to shorten patch-application times. Though none of these conclusions are certain, debates over these harms and benefits need to look faster and farther to give cyber defenders and policymakers time to set a course and weigh trade-offs. We hope that this approach can help accelerate this discussion, even among those who might disagree with aspects of these models.

Authors

Andrew Lohn is a senior fellow with the CyberAI Project at CSET, where Krystal Jackson is a visiting junior fellow.

Acknowledgments

For feedback and assistance, we would like to thank John Bansemer, James Dunham, Micah Musser, Chris Rohlf, and Shelton Fitch. We would also like to thank Justin Grana and Alexei Bulazel for their insightful comments.

Document Identifier: doi: 10.51593/2022CA002

Endnotes

[1] Andrew J Lohn and Krystal Alex Jackson, "Will AI Make Cyber Swords or Shields: A few mathematical models of technological progress," arXiv:2207.13825 [cs.CR], 2022, http://arxiv.org/abs/2207.13825

[2] Eugene Lim et al., "Turing in a Box: Applying Artificial Intelligence as a Service to Targeted Phishing and Defending against AI-Generated Attacks," in *Black Hat*, GovTech Singapore (2021), https://i.blackhat.com/USA21/Wednesday-Handouts/US-21-Lim-Turing-in-a-Box-wp.pdf.

[3] "Verizon: 2021 Data Breach Investigations Report," *Computer Fraud & Security* 2021, no. 6 (2021): 4.

[4] Pavlo Burda et al., "Testing the Effectiveness of Tailored Phishing Techniques in Industry and Academia," *ARES '20: Proceedings of the 15th International Conference on Availability, Reliability and Security*, no. 3 (August 2020): 1–10, https://doi.org/10.1145/3407023.3409178; Tian Lin et al., "Susceptibility to Spear-Phishing Emails: Effects of Internet User Demographics and Email Content," *ACM Transactions on Computer-Human Interaction: A Publication of the Association for Computing Machinery* 26, no. 5 (October 2019), https://doi.org/10.1145/3336141.

[5] David Brumley, "Open Source Security Podcast EP. 151 - The DARPA Cyber Grand Challenge With David Brumley," August 12, 2019, https://forallsecure.com/blog/open-source-security-podcast-ep-151-the-darpa-cyber-grand-challenge-with-david-brumley.

[6] Micah Musser and Ashton Garriott, "Machine Learning and Cybersecurity" (Center for Security and Emerging Technology, June 2021), https://cset.georgetown.edu/publication/machine-learning-and-cybersecurity/; Ben Buchanan et al., "Automating Cyber Attacks" (Center for Security and Emerging Technology, November 2020), https://cset.georgetown.edu/publication/automating-cyber-attacks/; Dakota Cary, "Robot Hacking Games" (Center for Security and Emerging Technology, September 2021), https://cset.georgetown.edu/publication/robot-hacking-games/.

[7] Thomas Maillart et al., "Given Enough Eyeballs, All Bugs Are Shallow? Revisiting Eric Raymond with Bug Bounty Programs," *Journal of Cybersecurity* (2017), http://arxiv.org/abs/1608.03445; Marcel Böhme and Brandon Falk, "Fuzzing: On the Exponential Cost of Vulnerability Discovery," in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (New York: Association for Computing Machinery, 2020), 713–24.

[8] Mingyi Zhao and Peng Liu, "Empirical Analysis and Modeling of Black-Box Mutational Fuzzing," in *Engineering Secure Software and Systems* (Springer International Publishing, 2016), 173–89.

[9] Maillart et al., "Given Enough Eyeballs, All Bugs Are Shallow? Revisiting Eric Raymond with Bug Bounty Programs."

[10] Raphael Hiesgen et al., "The Race to the Vulnerable: Measuring the Log4j Shell Incident," arXiv:2205.02544 [cs.CR], 2022, http://arxiv.org/abs/2205.02544; Tatum Hunter and Gerrit De Vynck,

“The ‘Most Serious’ Security Breach Ever Is Unfolding Right Now. Here’s What You Need to Know,” The Washington Post, December 20, 2021, https://www.washingtonpost.com/technology/2021/12/20/log4j-hack-vulnerability-java/.

[11] Steve Povolny, “Log4Shell Vulnerability Is the Coal in Our Stocking for 2021,” McAfee Blog. December 20, 2021, https://www.mcafee.com/blogs/enterprise/mcafee-enterprise-atr/log4shell-vulnerability-is-the-coal-in-our-stocking-for-2021/.

[12] Chester Wisniewski, “Log4Shell: No Mass Abuse, But No Respite, What Happened?,” Sophos News, January 24, 2022, https://news.sophos.com/en-us/2022/01/24/log4shell-no-mass-abuse-but-no-respite-what-happened/.

[13] Danny Palmer, “Log4j Flaw: Thousands of Applications Are Still Vulnerable, Warn Security Researchers,” ZDNet, April 28, 2022, https://www.zdnet.com/article/log4j-flaw-thousands-of-applications-are-still-vulnerable-warn-security-researchers/.

[14] Giorgio Di Tizio, Michele Armellini, and Fabio Massacci, “Software Updates Strategies: A Quantitative Evaluation against Advanced Persistent Threats,” arXiv:2205.07759 [cs.CR], 2022, http://arxiv.org/abs/2205.07759; Kathleen Metrick, Jared Semrau, and Shambavi Sadayappan, “Think Fast: Time Between Disclosure, Patch Release and Vulnerability Exploitation — Intelligence for Vulnerability Management, Part Two,” Mandiant, April 13, 2020, https://www.mandiant.com/resources/time-between-disclosure-patch-release-and-vulnerability-exploitation.

[15] Frank Li and Vern Paxson, “A Large-Scale Empirical Study of Security Patches,” in *CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (New York: Association for Computing Machinery, October 2017), 2201–15; Lillian Ablon and Andy Bogart, “Zero Days, Thousands of Nights: The Life and Times of Zero-Day Vulnerabilities and Their Exploits” (Rand Corporation, 2017), https://www.rand.org/pubs/research_reports/RR1751.html.

[16] Li and Paxson, “A Large-Scale Empirical Study of Security Patches.”

[17] Li and Paxson, “A Large-Scale Empirical Study of Security Patches.”

[18] Antonio Nappa et al., “The Attack of the Clones: A Study of the Impact of Shared Code on Vulnerability Patching,” in *2015 IEEE Symposium on Security and Privacy*, 692–708.

[19] Lohn and Jackson, “Will AI Make Cyber Swords or Shields: A few mathematical models of technological progress.”

[20] Frank Li et al., “Keepers of the Machines: Examining How System Administrators Manage Software Updates,” in *Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security* (USA: USENIX Association, 2019), 273–88.