Security Posture: A Systematic Review of Cyber Threats and Proactive Security

Amanda Jones

A Senior Thesis submitted in partial fulfillment
of the requirements for graduation
in the Honors Program
Liberty University
Spring 2022

Acceptance of Senior Honors Thesis

This Senior Honors Thesis is accepted in partial
fulfillment of the requirements for graduation from the
Honors Program of Liberty University.

_____

Michael Lehrfeld, Ph.D.
Thesis Chair

_____

Philip D. Schall, Ph.D., CISSP, RDRP, FITSP-D
Committee Member

_____

Emily C. Knowles, D.B.A.
Assistant Honors Director

_____

Date

Abstract

In the last decade, several high-profile cyber threats have occurred with global impact and devastating consequences. The tools, techniques, and procedures used to prevent cyber threats from occurring fall under the category of proactive security. Proactive security methodologies, however, vary among professionals where differing tactics have proved situationally effective. To determine the most effective tactics for preventing exploitation of vulnerabilities, the author examines the attack vector of three incidents from the last five years in a systematic review format: the WannaCry incident, the 2020 SolarWinds SUNBURST exploit, and the recently discovered Log4j vulnerability. From the three cases and existing literature, the author determined that inventory management, auditing, and patching are essential proactive security measures which may have prevented the incidents altogether. Then, the author discusses obstacles inherent to these solutions, such as time, talent, and resource restrictions, and proposes the use of user-friendly, open-source tools as a solution. The author intends through this research to improve the security posture of the Internet by encouraging further research into proactive cyber threat intelligence measures and motivating business executives to prioritize cybersecurity.

Keywords: Proactive threat intelligence, security controls, security posture, cybersecurity, WannaCry, Log4j, SolarWinds

Security Posture: A Systematic Review of Cyber Threats and Proactive Security

In recent years, data and security incidents have affected approximately 15% of all computer users globally (AMR, 2021). With the proliferation of Internet-connected smart devices comes the inevitable security vulnerabilities introduced by the native software, firmware, and hardware components installed. To combat security vulnerabilities, cybersecurity professionals develop a proactive security posture by developing current threat intelligence tradecraft to identify the wide array of tools, techniques, and procedures (TTPs) used by specific hacker entities. The methods and opinions of security professionals vary, so the goal of this research is to identify the common factors of catastrophic incidents and determine the best methods to prevent them. Considering the largest security incidents that have taken place in the last five years, Wannacry, Solarwinds, Log4j, this researcher understands the best approach security professionals take to mitigate risk of malware exploitation is to implement inventory management, auditing, & patching. As such, it is important first to establish the methodology for the research.

Research Method

There are many valid research and data collection methods that could be used to explore the proposed topic: surveys, interviews, simulated intrusion detection experiments, case studies, and many more. Because research is used extensively for real-world applications, the goals of this research are to assist decision-makers in designing and enforcing appropriate security policy and strategies. Therefore, the empirical will be almost exclusively favored over the theoretical to demonstrate practical application of recommendations from this research. Given that security best practices are subjective in nature, the research method chosen to pursue the best strategies in proactive cybersecurity is a blend of qualitative and empirical analysis. Moreover, the topic is

rooted in the computational sciences, which is interrupt-based and therefore causal, as well as in behavioral science, strategic intelligence, and criminal psychology. The latter fields are comprised of philosophies which can differ greatly from case to case. Therefore, the design method chosen must maintain a balance between the two seemingly contrasting fields and methodologies of research.

A systematic review is "a review of a clearly formulated question that uses systematic and reproducible methods to identify, select and critically appraise all relevant research, and to collect and [analyze] data from the studies that are included in the review" (Curtin University, 2022, para. 1). These studies contain an in-depth evaluation of a broader research topic using a few particular, well-documented examples. Furthermore, the "approach excels at bringing…an understanding of a complex issue through detailed contextual analysis of a limited number of events or conditions and their relationships" (USC, 2021, para. 2). Most importantly, systematic reviews may be instrumental in encouraging additional research on an emerging problem and may be a starting point for the development and testing of other hypotheses. Due to the many factors involved in cyber attacks, this thesis lends itself systematic review design to evaluate a few particularly challenging scenarios. Then, it is possible to assemble from primary and secondary sources those factors which may have caused or prevented the cyber attack from happening. A systematic review allows for consideration of contextual factors characterizing the chosen cases. Because the purpose of the hypothesis is to gain insight into the prevention of emerging cyber threats as a whole, it follows that multiple real-world analyses would contribute to a broader understanding of the topic.

Systematic Review

The systematic review will investigate three high-profile cybersecurity incidents as cases. Moreover, to maintain currency of research, the incidents chosen have occurred in the last five years, so that the resulting remediation strategies and technologies discussed remain relevant to current standards. In so doing, it is possible to extract common themes from each incident, determine the best practices and lessons learned for a proactive security posture, and provide recommendations for overcoming the obstacles to implementing these practices. The cases selected for this research endeavor are the 2017 WannaCry incident, the 2020 SolarWinds SUNBURST supply-chain attack, and the Log4j vulnerability, discovered late in 2021.

WannaCry affected thousands of organizations globally with a self-propagating ransomware virus, or worm, that rendered the operations of the victim systems futile when proprietary data was completely encrypted. Only the discovery of a deactivation key by a security researcher slowed its spread to even more systems. The SolarWinds supply-chain attack affected around 18,000 well-known vendors by sabotaging a widely used enterprise software suite. Affected customer bases of this attack include Microsoft, FireEye, and CrowdStrike (MITRE, 2020). Fortunately, the customers affected were some house names in cutting-edge technology, cybersecurity, and incident response; therefore, the investigations launched were well-documented, though the investigation continues as of writing. Lastly, Log4j, and its subsequent privilege escalation Log4Shell was recently discovered to have affected most of the Internet's infrastructure. A vast majority of web services publicly available today were found to be vulnerable to this exploit, and many are still vulnerable due to the difficulty of implementing patches. Because of the high impact of the three incidents, a thorough analysis will reveal the universally applicable lessons learned to prevent future incidents.

WannaCry

WannaCry was a global cyber incident that was discovered in May of 2017. Affecting large corporations and healthcare facilities in over 50 countries, the attack spread through a ransomware worm which exploited vulnerabilities in the Windows Server Message Block (SMB) protocol (Chen & Bridges, 2017). The worm, once spread, would encrypt the host computer's data, rendering it unusable and compromising the integrity and availability of the data permanently unless a method of decryption was discovered. The SMB vulnerability that allowed the WannaCry attack is identified by the Common Vulnerabilities and Exposures (CVE) database as CVE-2017-0145 (MITRE, 2017). More specifically, SMB version 1 allowed for remote code execution, or running executables from another machine or interface, in the following Windows operating systems:

- Microsoft Windows Vista SP2
- Windows Server 2008 SP2 and R2 SP1
- Windows 7 SP1
- Windows 8.1
- Windows Server 2012 Gold and R2
- Windows RT 8.1
- Windows 10 Gold, 1511, and 1607
- Windows Server 2016

The vulnerability is a result of improper input validation in the SMB protocol. Because Microsoft does not provide the open-source code for SMB version 1, the technical vulnerability details showing the problematic code have not been provided. However, it is known that the

attack vector which was deployed in the ransomware is the leaked exploit EternalBlue, alleged by the Russian research group Kapersky Lab (2015) to have been designed by United States National Security Agency (NSA) researchers. The scope of systems affected by the attack is best summarized by C.E.R.Team EU (2017):

> The exploit used – EternalBlue – has been made available on the Internet through the ShadowBrokers dump on April 14th, 2017, but already earlier patched by Microsoft on March 14th, 2017 as part of MS17-010 for the supported versions of the Microsoft Windows operating system. Unfortunately, the patch was not available at that time for legacy Windows XP, Windows 8, as well as for Windows Server 2003 systems. Even in case of systems where the patch was available, it appears that many organizations have not installed it. (p. 1)

According to Sumo Logic (2020), the three fields of interest that demonstrate the exploitability of a system to the EternalBlue exploit are the name, status, and subcommand artifacts of the SMB commands field. Feeding a crafted ping to the system can show whether SMBv1 is vulnerable to the EternalBlue exploit. The EternalBlue tool has been publicly available through the Metasploit penetration testing framework and deployed on Kali Linux systems for several years as of the date of this publication. Relying on a backdoor named DoublePulsar, an attacker may use EternalBlue to exploit the SMBv1 input validation vulnerability. The first protection against being attacked in this way is practicing due diligence in installing the latest versions and patches of operating systems on the operational network. This is particularly important with software for which critical CVEs have been previously identified. Inability to maintain system updates was the main reason why WannaCry was able to spread so rapidly to so many systems globally. Even as of writing, one node that is vulnerable with CVE-2017-0145 can lead to the compromise of data and systems on the entire network. However, one

additional security failing in this scenario worth investigating is how the leak of EternalBlue occurred.

The tools used by the group behind WannaCry were leaked by a hacking group called TheShadowBrokers in April of 2017, one month before the attack. Allegedly developed by a group called Equation Group, purportedly linked to the NSA, the EternalBlue tools, among others, were commandeered to quickly develop and drop the worm. Despite the amount of controversy surrounding the identities of the groups involved, the NSA and other intelligence agencies have not made statements to confirm their involvement in developing the tools at the time of writing. Furthermore, the Threat Hunter Team at Symantec (2019) reported that a hacking group identified by the name Buckeye made use of the DoublePulsar backdoor even before TheShadowBrokers' leak of the Equation Group tools, which may potentially link Buckeye to one of the two groups. Considering the difficulty of identifying the identities of the Equation Group and TheShadowBrokers, and since APTs typically make operational security a priority to remain undetected (Ghafir & Prenosil, 2014), it is no surprise that threat intelligence and attribution for the WannaCry ransomware proved difficult for investigators. Luckily, identifying SMB-related threat activity is simple according to Sumo Logic (2020), as they allege the protocol is "ripe for behavior-based detection" (para. 3). The United States Government (USG) has attributed the WannaCry worm to nation-state hackers affiliated with North Korea, unilaterally referred to as the Hidden Cobra APT (CISA, 2021a). The government analyzed the indicators of compromise (IOCs) and partnered with other nations such as Canada, the United Kingdom, Japan, Australia and New Zealand in the attribution, and the tools, techniques, and procedures used aligned with MITRE's (2021a) database on the Lazarus Group, another name for the North Korean APT. However, despite the scale of the attack and the mass efforts to put an

end to the worm, the remediation of the WannaCry ransomware surprisingly originated with a single security researcher.

Marcus Hutchins, referred to online by the alias MalwareTech, is a security and malware researcher from the United Kingdom who found a solution to the rapid spread of the WannaCry worm. As a part of his job responsibilities, Hutchins was tasked with continuously monitoring for unregistered malware command and control (C2) domains to identify botnets. They are best described by Hutchins (2019):

1. Look for unregistered or expired C2 domains belonging to active botnets and point it to our sinkhole (a sinkhole is a server designed to capture malicious traffic and prevent control of infected computers by the criminals who infected them).
2. Gather data on the geographical distribution and scale of the infections, including IP addresses, which can be used to notify victims that they're infected and assist law enforcement.
3. Reverse engineer the malware and see if there are any vulnerabilities in the code which would allow us to take-over the malware/botnet and prevent the spread or malicious use, via the domain we registered. (paras. 8-10)

After registering one such identified domain and pointing it towards a malware sinkhole server, Hutchins discovered that execution of the propagation payload of WannaCry failed. Hutchins corroborated his findings with other security researchers on Twitter who discovered the worm dropper no longer executed. Because the domain was hard-coded in the propagation snippet, it would be simple for the original engineers of the worm to change the domain and continue infecting other systems; however, with this kill switch enabled, no further computers were

infected with the original WannaCry worm that was spreading through the Internet. By the time the kill switch was discovered, the effort to patch SMB on a broader scale had already been initiated. Nonetheless, systems that are unpatched and have the vulnerable SMB version may still be infected with the exact same exploit.

Although the exploit may still be executed on vulnerable systems, awareness of the vulnerability has been spread such that many systems have been patched or updated. Mohamed of Microsoft (2017) supplied the patches for even end-of-life Windows systems. However, many dated infrastructures continue to rely on SMB version 1 and unpatched Windows operating systems (United State Federal Bureau of Investigation et al., 2022). Regardless of a vulnerability's criticality rating and the thousands of computer systems affected, the cost to maintain systems often outweighs the risk of cyber attacks to higher level decision-makers, particularly those industries which depend on very low downtime guarantees such as critical infrastructure and health services. Nonetheless, the United Kingdom's National Cyber Security Centre (NCSC) (2021) has developed a guide for proactively mitigating ransomware attacks in the wake of WannaCry, including practicing good asset management and keeping software and systems up to date with patches. In a perfect world, all systems would already have implemented the patches for the vulnerability when they were released before the attack, and no systems would have been compromised by the WannaCry ransomware.

SolarWinds

SolarWinds is the colloquial name for the 2020 supply chain attack on the SolarWinds software company which affected all customers using the Orion network management system (NMS). Attributed by the USG to advanced persistent threat group APT29, or as they call themselves, CozyBear, the exploit was intended to siphon intelligence and establish persistence

in the infrastructure of the companies using the product (Alshamrani et al., 2019). The exploit marked a rare occurrence in malicious cyber activity, as supply-chain attacks are purportedly much more difficult to execute than dropping viruses or ransomware (Mandia, 2020). This is due to the extensive quality assurance and auditing of larger vendors wherein the risk of vulnerabilities deployed to customers would have global, catastrophic impact. Such was the case in the supply chain attack in December of 2020, when over 300,000 customers of SolarWinds, including Microsoft, FireEye, CrowdStrike, and Oracle, found their systems compromised via the SolarWinds product (Williams, 2021).

As an incident response and cyber threat intelligence powerhouse, FireEye was the first to disclose the news of the breach, though they were not initially certain of the attack vector of the suspected nation state actor. However, they operated with utmost transparency during the investigation of the attack, sharing useful threat intelligence information with the rest of the community and with customers. They revealed within days of discovering the zero-day that the attack, called SUNBUSRT, originated within the supply-chain as a trojan horse update to the Orion NMS. Because the supply chain of the Orion software updates was exploited, the malware contained the digital signature of SolarWinds products, which is intended to affirm the authenticity of the product. The behavior of SUNBURST is described by FireEye (2020):

> After an initial dormant period of up to two weeks, [SUNBURST] retrieves and executes commands, called “Jobs”, that include the ability to transfer files, execute files, profile the system, reboot the machine, and disable system services. The malware masquerades its network traffic as the Orion Improvement Program (OIP) protocol and stores reconnaissance results within legitimate plugin configuration files allowing it to blend in with legitimate

> SolarWinds activity. The backdoor uses multiple obfuscated blocklists to identify forensic and anti-virus tools running as processes, services, and drivers. (para. 3)

In essence, the malware evades any antivirus and intrusion detection or prevention systems (IDS/IPS) to execute complete control over the system using the legitimate infrastructure of the network management system. Because of this sophisticated manipulation of the SolarWinds product, the nation-state actors were able to stay hidden within consumer networks running the SolarWinds Orion NMS, even making lateral movements to other systems and stealing data after the exploit was disclosed to the public.

Due to the post-compromise activity exhibited by the threat actors, a myriad of indicators of compromise were uncovered that assisted in attributing the attack. The sophistication of the attack, including the delayed malware execution awaiting the joining of the infected system to the domain, as well as the feat of infiltrating SolarWinds' product update pipeline itself, suggested a nation-state adversary was behind the attack. One of many names associated with the Russia's Foreign Intelligence Service, CozyBear is also linked with the intrusion of the United States Democratic National Committee as well as Operation Ghost targeting the European Ministry of Affairs (Faou et al., 2019). The main attribution indicator for the SolarWinds attack is the strong code similarities between the SolarWinds campaign and previous attacks by the CozyBear group, including the use of MISPRINT/SIBOT malware (MITRE, 2021b). In remediation, according to SolarWinds CEO and President Sudhakar Ramakrishna, "[SolarWinds has] obtained new digital code-signing certificates and have rebuilt the versions signed with the certificate to be revoked, have re-signed [the] code, and have re-released all of the products previously signed with the certificate to be revoked" (SolarWinds, 2021, para. 7). Furthermore, the security community has collaborated on a list of all known detections and signatures

associated with the faulty Orion releases in order to protect the community as they seek to install legitimate product patches to this malware.

The SUNBURST backdoor allegedly exploited by CozyBear subverts the advisements given by most security professionals; in this case, patching systems with the latest Orion NMS update was the vector for the malicious payload to establish command and control over approximately 18,000 organizational and public-facing domains across the world (CISA, 2021b). Thus, the objective of the attack was twofold: the group needed to design malware that would first be undetectable and insert the malicious code into the Orion software, while they ensured the code itself would remain undetected to all consumers of the NMS product. The initial breach is described by CrowdStrike (2021):

- SUNSPOT is … malware used to insert the SUNBURST backdoor into software builds of the SolarWinds Orion IT management product.
- SUNSPOT monitors running processes for those involved in compilation of the Orion product and replaces one of the source files to include the SUNBURST backdoor code.
- Several safeguards were added to SUNSPOT to avoid the Orion builds from failing, potentially alerting developers to the adversary's presence. (paras. 2-4)

A supply-chain breach like the one described above violates the inherent trust consumers place in purchased and open-source software. Therefore, it is incumbent upon concerned consumers to implement another industry standard in cybersecurity: zero-trust architecture. This philosophy proposes that vulnerabilities can and will be present along every node in an organization's architecture, be it the human element (employees), physical security controls, administrative controls, or technical controls. Zero-trust argues for auditing and non-repudiation at each level to

secure trust (Rose et al., 2020). The occurrence of side-channel and supply-chain attacks indicate that it is imperative to proactively detect and defend against attacks through a thorough knowledge of the interactions between system components (Hatfield, 2021). Furthermore, it is also recommended that routine audits of system activity and processes be automated through logging and machine learning data analytics solutions which can detect and predict anomalies (Barker, 2020).

Log4j

Log4j 2 is an Apache tool written in Java that provides logging services for other software on the system. Recently discovered in November of 2021, a zero-day vulnerability in Log4j 2 allowed for remote code execution and near-unfettered access to services such as Cloudflare (an entity providing much of the backbone of the Internet), iCloud, Minecraft, and Twitter, among many others (Mott, 2021). This vulnerability allows an attacker to send a single command string to the logging service and execute code remotely, potentially establishing control over the server and all nodes on the network. Because of the popularity of Apache services and Log4j, it is difficult to quantify just how many servers and vendors are affected by the vulnerability. However, the criticality and suspected scope of the vulnerability caused the U.S. Cybersecurity & Infrastructure Security Agency (CISA) to issue Emergency Directive (ED) 22-02: Mitigate Apache Log4j Vulnerability for all federal civilian agencies. The directive required administrators to complete all steps necessary to root out the vulnerability on government systems (CISA, 2022). While the situation is continuing to develop and the extent of the vulnerability's impact is not fully known, the literature as of publication suggests that the Log4j vulnerability is highly critical and poses a great threat to the security of the Internet of Things.

The web vulnerability originates within the implementation of the Java Naming and Directory Interface in Log4j (Apache, 2021). The Java Naming and Directory Interface (JNDI) is responsible for lookups within Java programs associated with the Domain Name Service (DNS), Lightweight Directory Access Protocol (LDAP), Remote Method Invocation (RMI) and more. Within Log4j 2, this application programming interface (API), was configured by default to allow lookups within log messages sent to the server via any protocol, which would establish a connection between the API and that domain. Therefore, an attacker is able to point the server to a malicious site or a command and control (C2) domain simply by sending data to the server utilizing the vulnerable version of Log4j. Wortley et al. (2021) describe the steps required to exploit the vulnerability, demonstrating the ease with which the steps may be completed:

- Data from the User gets sent to the server (via any protocol).
- [The server] logs the data containing the malicious payload from the request ${jndi:ldap://some-attacker.com/a}, where some-attacker.com is an attacker controlled server.
- The [Log4j] vulnerability is triggered by this payload and the server makes a request to some-attacker.com via "Java Naming and Directory Interface" (JNDI).
- This response contains a path to a remote Java class file (ex. http://second-stage.some-attacker.com/Exploit.class), which is injected into the server process.
- This injected payload triggers a second stage, and allows an attacker to execute arbitrary code. (para. 7)

Patches for Log4j 2 and its dependencies have been released to remediate the vulnerability, but threat actors have developed sophisticated exploits to maintain persistence on

servers and networks once patched. Discovered in December by CrowdStrike, a new threat group called Aquatic Panda launched an attack utilizing Log4Shell on an academic institution (Wiley, 2021). While not as sophisticated in operational security as the groups behind the WannaCry and SolarWinds incidents, Aquatic Panda were on track to take down the target's endpoint monitoring and response service. While the victim was quickly alerted of the intrusion by Aquatic Panda and patched the vulnerability, the shell access provided by the exploit would have allowed the attacker to remotely perform tasks at any privilege level provided the malicious command and control server was configured properly. Moreover, Yan et al. (2021) of Palo Alto Networks have identified through diagnostic data of their network devices across the Internet over "60,476,284 hits that had the associated packet capture that triggered the signature" for the Log4j exploit (para. 20). Mass scans are being conducted by employees, researchers, and attackers alike in order to determine what systems are vulnerable to the exploit, with many of the latter even using the scan as a dropper for a shell on those systems. It is suspected that the Log4j vulnerability could have devastating effects if targeted by more advanced threat groups. But, whether perpetrated by a script kiddie or an organized crime group, shell access by any threat actor can compromise the confidentiality, integrity, and availability of the data and systems affected.

All new versions of Apache's Log4j 2 have mitigated the vulnerability by hardening the JNDI API and removing compatibility with LDAP, but the question remains: How does a vulnerability so detrimental infiltrate the infrastructure of the Internet through several iterations of the product? Unfortunately, even rigorous testing of the software cannot predict every outcome, though one testing method may have proved helpful in this case if used proactively. Automated fuzzing provides a potential solution to code analysis and security testing: this

solution creates millions of permutations of input strings for a given executable and can be customized according to the accepted parameters of the program. Applied to Log4j, it is possible that fuzzing the JNDI implementation with log messages containing test domains could have revealed the vulnerability ahead of time. The most important countermeasure to the vulnerability, however, is patching current systems. As of writing, no high-profile attacks with extreme consequences have been discovered, or at least publicly disclosed. However, as patches have been provided for several Java versions of Log4j 2, it is even more imperative that administrators updated all web services and vulnerable applications to the new version. This way, sophisticated self-propagating attacks like WannaCry, which preyed upon systems that were out-of-date when the appropriate patch was months since released, may be mitigated, or avoided altogether. Should organizations fail to implement the patches provided, it is suspected that high-dollar organizations, such as tech companies, critical infrastructure, government and government contractors, and healthcare industries will be targeted for attack with catastrophic effects in its wake.

Existing Security Measures

To round out the systematic review, it is important to consider additional security recommendations and best practices. The literature regarding proactive security measures, or security best practices, is vast; experts point to several unique—and often costly—solutions to security vulnerabilities. Nonetheless, the articles reviewed contain significant findings and implications for the effectiveness of various cybersecurity countermeasures. Firstly, the National Initiative for Cybersecurity Education promotes extensive personnel training to create a secure cybersecurity architecture (Paulsen et al., 2012). Additionally, Rose et al. (2020) advocated for a zero-trust architecture in which authentication and authorization actions are performed before

each enterprise session, which improves security at each layer in the network. Logging and auditing solutions, particularly those that automate big data processing through machine learning, also allow for a proactive cybersecurity architecture (Muggler et al., 2017). Version control and inventory management give essential insight into the vulnerability status of an organizational network and allow for proactive countermeasures to be systematically performed (Knorr, 2013). Proactive measures allow for organizations to prevent attacks on low-hanging fruit and identify vulnerabilities before they are exploited.

Furthermore, the literature contains evidence that proper incident response measures assist in preventing future exploitation. In their evaluation of cyber attack attribution on industrial control systems, Cook et al. (2016) determined that honeypots and malware analysis provided the most reliability and the least performance detractors for threat intelligence gathering, and network forensics and intelligence-led attribution detailed the most accurate metrics of identification and intent. Extensive logging and data analysis may be used to assist in the attribution process to gather actionable threat intelligence and prevent offenders from covering their tracks. Skopik and Pahi (2020) affirmed in their study that analysis of TTPs reveals patterns which can be attributed to one hacker or coalition. Additionally, the organization and broader scope of the attack can be much more telling than the technical artifacts such as IP addresses or network traffic, which can be simple to spoof (Maglaras et al., 2018). Finally, Miller and Davenport (2020) concluded that the large quantities of data analyzed in the intelligence process were much more easily processed using machine learning; attribution teams may compare the results of the machine analysis with normal activity to identify the origin of attacks. Forensic information can then be categorized, analyzed at large, and checked for quality.

Independent research is not the only source of proactive security recommendations. Authoritative organizations in cybersecurity such as the National Institute of Standards and Technology (NIST) have developed hundreds of resources containing recommendations for industries such as healthcare, critical infrastructure, and government to implement for a comprehensive security posture. The 800 series of special publications relate to Computer and Information Security, while the 1800 series contain Cybersecurity Practice Guides (NIST, 2022). The special publications contain myriad recommendations that provide an ample starting point for stakeholders looking to improve the security posture of their organization. All things considered, the literature contains evidence which supports the important role of adherence to security standards in proactive cybersecurity through administrative, technical, and physical controls.

Despite the myriad security measures available, cyber threats affect hundreds, if not thousands of systems daily (AMR, 2021). Ultimately, many of the most optimal or comprehensive practices are simply not realistic for many organizations due to resource restrictions (Lidster & Rahman, 2018). Moreover, many emerging cyber threats exploit unknown (zero-day) vulnerabilities for which no security controls exist. This was made clear by the ransomware infiltration of the Fortune 500 Colonial Pipeline (Keary, 2022), and the espionage and persistence established within cutting-edge tech powerhouses Microsoft, Oracle, and FireEye (Peisert et al., 2021). Nonetheless, determining the impediments to proper security posture and solutions to these problems may assist the community in prioritizing adequate cyber infrastructure with solutions suitable to available resources.

Discussion

The three incidents revealed several key takeaways to maintain a proactive security posture. It is hoped that, by expounding on the takeaways from WannaCry, SolarWinds, and Log4j, the last of which is a known and ongoing vulnerability affecting many systems on the Internet, more awareness is brought to these security practices which may prevent future cyber incidents. In turn, individuals and organizations may be saved from serious personal, financial, or professional losses due to the exploitation or breach by threat actors. Following is a discussion of the major lessons learned for security measures from the systematic review, the obstacles to implementation of the security measures, and suggested solutions to these obstacles. The author acknowledges that many of the proposed solutions will not solve issues with resources or implementation universally. Solutions provided are generated in light of the existing literature, industry standards and best practices. The three most critical practices for maintaining a proactive security posture are inventory management, auditing, and patching.

The WannaCry incident demonstrated the criticality of inventory management, or keeping an accurate record of all systems and their associated services, configurations, and vulnerabilities. The self-propagating virus determined which systems adjacent to its current host contained the same SMBv1 vulnerability through intelligence collection and scans, and thus was able to pivot throughout the Internet at lightning speed. If the virus, however, encountered systems which were not vulnerable by any means, it was likely they would not be infected, unless there existed another means of lateral movement to that system to drop the payload. Inventory management allows for a quick response to newly identified vulnerabilities (Brykczynski & Small, 2003). Automated solutions make inventory management simple; if all systems within the network topology are accurately recorded within the management database, it

is simple to receive alerts when new vulnerabilities are discovered, or patches are released for all affected systems and services found the management system. Furthermore, remote monitoring and management (RMM) solutions allow configuration changes and updates to be conducted remotely, meaning that larger organizations with several physical locations do not require technicians to travel and work on the physical console of that system. This reduces the burden on employees and the human element in general to simply remember the topology of the network. "Live" documentation, which is routinely updated when configuration or major changes to systems are made, helps with knowledge sharing and consistency in procedures when it comes to securing and maintaining the organizational network.

Furthermore, end-of-life (EOL) systems and software must be documented within the inventory management system, and the organization should put forth every effort to upgrade to supported systems (Townsend et al., 2020). This is extremely pertinent to industrial control and critical infrastructure systems, where updates typically require downtime, and every second down can mean major losses to the organization (Knorr, 2013). I recommend attributing ample resources to designing a test system to mimic the operational system, then updating and troubleshooting the test system before performing actions on the operational systems on a wide scale. Additionally, organizations should inform customers of scheduled downtime for industrial control systems, conducting updates in segments to keep facilities operational, albeit with some delays in service. Although there are barriers to implementing new solutions, assist in creating a comprehensive and proactive inventory management schema that helps protect the organization from catastrophic attacks.

The SolarWinds incident demonstrated the importance of auditing, adopting a zero-trust architecture to the fullest extent possible. Auditing is a critical practice in all fields, as it ensures

that the policies, procedures, and behaviors of an organization adhere to known ethics and standards. Applied to computer security, auditing ensures that all network activity within the purview of an organization is legitimate, non-repudiated, and aligned with the organization's use of the technology. The technical aspect of auditing can be implemented in a variety of ways, but the most common methods are network monitoring and logging solutions, intrusion detection or prevention systems, and machine learning threat intelligence solutions. Logging all foreign or critical activities on the network allows for a paper trail for abnormal activities or those that may cause major changes. IDS/IPS solutions take this network traffic data and analyze it further against known rules, alerting the network administrator or incident response team of the traffic and taking action such as quarantining the process or IP address. Finally, data scientists have created user-friendly modules for machine learning threat intelligence solutions that may be trained to detect and predict abnormal activity on the network in advance, or learn to investigate the source, vector, or surface of the attack (Qamar et al., 2017).

Knowledgeable human auditors validate the inputs and outputs of technical controls and provide additional recommendations to decision-makers at an organization. Verifying alerts and activity reinforces the notion of zero trust in the organization's architecture by operating on the assumption that any component of the organization may be exploited. As regards SolarWinds, auditing the activity of the SolarWinds network manager is what led to its initial discovery on FireEye's organizational network. Exhaustive auditing was likely not conducted at an earlier date because of the inherent trust in the SolarWinds product and infrastructure itself. From observation, the main impediment to implementing these solutions is a lack of resources. The affected organization may lack expertise (as may often be the case with smaller institutions or those less reliant on technology), or the organization may not have the funds necessary to invest

in proactive technical auditing measures. While many solutions are too costly for some entities, it is important to note that free, open-source technical solutions exist, allowing for malicious code review, and use of the solution comes at no cost to the organization.

Finally, the Log4j situation teaches a simple lesson: administrators must install system patches immediately, particularly those that provide essential security remediations. Update management, or patching, depends on proper inventory management, (Dissanayake, 2021). As it now stands, if every system dependent on the vulnerable version of Log4j 2 were patched today, the incident would be resolved entirely. However, the main issue with patching is that the systems that depend on the vulnerable software, particularly custom code solutions that are hard coded to function with the current version, may not function the same or at all with patches. While modularity is a property adopted by most knowledgeable software developers, patches may still break functionality (Mockus & Weiss, 2002). Administrators should rely upon version control software and snapshots when applying patches, such that, if a patch does break or modify functionality of the service, the damage is easily undone. Furthermore, for systems where downtime service-level agreement standards are measured in seconds, a mockup system that mirrors the operational network can be created to implement the patch, testing the functionality and patching sequence before rollout. These solutions may provide a remedy to the obstacle of functionality issues with patching, and it is hoped that they will encourage organizations with the vulnerable version of Log4j to patch affected systems immediately.

Future Threat Research

Though inventory management, auditing, and patching comprise proactive security posture, researchers and professionals continue to develop ways to provide cyber threat data and analytics to users real-time. One major advancement in proactive security is the concept of

unified threat reporting. Just as CVEs are uploaded to a unified and recognized database, indications of compromise and other cyber threat artifacts may be shared through a centralized or subscription-based threat model using the Structured Threat Information Expression™ and Trusted Automated eXchange of Indicator Information™ (STIX-TAXII) protocol suite (OASIS Cyber Threat Intelligence Technical Committee, 2021). STIX-TAXII is a method of sharing threat intelligence information in a standardized language (STIX) and well-known services/protocols (TAXII). This allows for easy processing and dissemination of cyber threat intelligence (CTI) not only to stakeholders within the organization, but to other organizations and industries which may be affected by the threat. According to ThreatConnect (2021), "The eight indicator constructs include: Observable (activity), Indicator (what to watch), Incident (where), TTP, Exploit Target, Campaign (why), Threat actor – (who), and Course of action" (para. 6). The indicator constructs allow for standardized sharing of forensic information which can then be processed by the desired analytics or machine learning solution of each organization. The National Council of ISACs (NCI) acts as a central knowledge base for sector-based Information Sharing and Analysis Centers (ISACs) to share STIX-TAXII data with other members in that sector. The intercommunication between industries of threat intelligence data fosters a community-wide threat intelligence effort to prevent widespread exploitation by emerging threats. The benefit of the ISACs is clear: sharing of CTI and prevention methods improves the security of each industry and of the global computer infrastructure.

Not only is real-time sharing of CTI critical, but IOC analysis must also be conducted in real-time. Machine learning provides a way for threat analysts to automate data processing and export critical CTI recommendations synchronously. There are myriad applications and current uses of machine learning in threat intelligence today. One basic function is filtering and

classification of data (natural language processing)—a machine learning algorithm can classify based on certain keywords and add words to its word bank if they are frequently associated with the existing ones (Rodriguez & Okamura, 2020). However, to gather intelligence on technical vulnerabilities and the origins of an attack, more sophisticated algorithms may need to be used. Zhou et al. (2019) proposed using ensemble learning, or multiple machine learning algorithms or models to process large data sets and gather more relevant information and check for accuracy. These methods, however, have proven useful for detecting threats in a reactive posture, while Rodriguez and Okamura (2020) promoted a proactive posture by processing up-and-coming CTI topics online. Machine learning algorithms may also take as inputs STIX-TAXII information to provide anomaly detection, predictive forecasting, pattern recognition, and many more capabilities.

Conclusion

Proactive cybersecurity is a multidisciplinary objective that integrates a myriad of human, physical, administrative, and technical factors. Forming a strategy of proactive cybersecurity requires knowledge of several specialties from system administration to criminal psychology; however, adopting security best practices is as simple as learning from past incidents: the WannaCry attack, the SolarWinds SUNBURST supply-chain attack, and the Log4j 2 vulnerability. Through this systematic review, the author determined correlations between the attack vectors employed in each scenario and the security best-practices which may have prevented exploitation. These best practices are summarized as proper inventory management, auditing and automated analysis of logs and digital artifacts, and system and software patching. Having a thorough knowledge of the interactions on an organization’s network allows for detection and prevention of cyber threats before they compromise the system. Furthermore,

unified threat reporting and machine learning are promising developments that can benefit the global community through real-time threat analytics. Though many impediments exist to implementing these practices, such as financial and knowledge resources, there are many open-source technical solutions available. Regardless of the obstacles, stakeholders would benefit from adopting a proactive cybersecurity posture to protect their company vision, integrity, and livelihood.

References

Alshamrani, A., Myneni, S., Chowdhary, A., & Huang, D. (2019). A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials*, *21*(2), 1851-1877. doi:10.1109/comst.2019.2891891.

AMR. (2021). Kaspersky Security Bulletin 2021: Statistics. Securelist English Global securelist.com. https://securelist.com/kaspersky-security-bulletin-2021-statistics/105205/.

Apache. (2021). *Apache Log4j security vulnerabilities*. https://logging.apache.org/log4j/2.x/security.html.

Barker, C. (2020). *Applications of Machine Learning to Threat Intelligence, Intrusion Detection and Malware* (Publication No. 985) [Honors thesis, Liberty University]. Scholars Crossing. https://digitalcommons.liberty.edu/honors/985.

Brykczynski, B., & Small, R. A. (2003). Reducing Internet-based intrusions: Effective security patch management. *IEEE Software*, *20*(1), 50-57. doi:10.1109/MS.2003.1159029.

C. E. R. Team-EU. (2017). *Wannacry ransomware campaign exploiting SMB vulnerability*. (CERT-EU Security Advisory 2017-012). https://cert.europa.eu/static/SecurityAdvisories/2017/CERT-EU-SA2017-012.pdf.

Chen, Q., & Bridges, R. A. (2017). Automated behavioral analysis of malware: A case study of WannaCry ransomware. *16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 454-460. doi:10.1109/ICMLA.2017.0-119 https://ieeexplore.ieee.org/abstract/document/8260673?casa_token=fnk4h5JNZ2QAAAAA:YQgGhsu3EgMNND1gD3WS-98BhGiMblVahkOcxNlkrAlfVxFcO3x9QOPZR6iH1GGtIdTHM2z1bLg.

CISA. (2021a). *North Korea Cyber Threat Overview and advisories*. CISA. https://www.cisa.gov/uscert/northkorea.

CISA. (2021b). *Malware analysis report* (Report no. AR21-105A MAR-10327841-1.v1 – SUNSHUTTLE). CISA. https://www.cisa.gov/uscert/ncas/analysis-reports/ar21-105a.

CISA. (2022). *Apache Log4j vulnerability guidance.* CISA. https://www.cisa.gov/uscert/apache-log4j-vulnerability-guidance.

Cook, A., Nicholson, A., Janicke, H., Maglaras, L., & Smith, R. (2016). Attribution of cyber attacks on industrial control systems. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, 3*(7). http://dx.doi.org/10.4108/eai.21-4-2016.151158.

CrowdStrike Intelligence Team. (2021). *Sunspot malware: A technical analysis.* crowdstrike.com. https://www.crowdstrike.com/blog/sunspot-malware-technical-analysis/.

Curtin University. (2022). *Systematic reviews: What is a systematic review?* LibGuides. https://libguides.library.curtin.edu.au/systematic-reviews.

Dissanayake, N., Jayatilaka, A., Mansooreh, Z., Babar, M. A. (2021). Software security patch management - A systematic literature review of challenges, approaches, tools and practices. *Information and Software Technology, 144.* https://doi.org/10.1016/j.infsof.2021.106771.

Faou, M., Tartare, M., & Dupuy, T. (2019). Operation Ghost. *ESET Research White papers, October 2019.* https://www.welivesecurity.com/wp-content/uploads/2019/10/ESET_Operation_Ghost_Dukes.pdf.

FireEye. (2020). *Highly evasive attacker leverages Solarwinds Supply Chain to compromise multiple global victims with Sunburst Backdoor*. Mandiant.

https://www.mandiant.com/resources/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.

Ghafir, I., & Prenosil, V. (2014). Advanced persistent threat attack detection: an overview. *International Journal of Advanced Computer Networks and Its Security*, *4*(4), 50-54.

Hatfield, N. (2021). *Software-Based Side Channel Attacks and the Future of Hardened Microarchitecture* (Publication No. 1059) [Honors thesis, Liberty University]. Scholars Crossing. https://digitalcommons.liberty.edu/honors/1059.

Hutchins, M. (2019). *How to accidentally stop a global cyber attacks*. MalwareTech. https://www.malwaretech.com/2017/05/how-to-accidentally-stop-a-global-cyber-attacks.html.

Kaspersky Lab. (2015). *Equation Group: The Crown creator of Cyber-Espionage*. https://www.kaspersky.com/about/press-releases/2015_equation-group-the-crown-creator-of-cyber-espionage.

Keary, J. (2022). *Rebuffing Russian Ransomware: How the United States Should Use the Colonial Pipeline and JBS USA Hackings as a Defense Guide for Ransomware* (Publication No. 1274) [Student Scholarship, Seton Hall University]. Law School Student Scholarship. https://scholarship.shu.edu/student_scholarship/1274/

Knorr, K. (2013). Patching our Critical Infrastructure: Towards an Efficient Patch and Update Management for Industrial Control Systems. *Securing Critical Infrastructures and Critical Control Systems: Approaches for Threat Protection*. 190-216. IGI Global.

Lidster, W., & Rahman, S. (2018). Obstacles to implementation of information security governance. *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data*

Science And Engineering (TrustCom/BigDataSE). 1826-1831. doi:10.1109/TrustCom/BigDataSE.2018.00276.

Mandia, K. (2020). *FireEye shares details of recent cyber attack, actions to protect community*. FireEye. https://www.fireeye.com/blog/products-and-services/2020/12/fireeye-shares-details-of-recent-cyber-attack-actions-to-protect-community.html.

Maglaras, L., Ferrag, M., Derhab, A., Mukherjee, M., Janicke, H., & Rallis, S. (2018). Threats, countermeasures and attribution of cyber attacks on critical infrastructures. *EAI Endorsed Transactions on Security and Safety, 5*(16). http://dx.doi.org/10.4108/eai.15-10-2018.155856.

Miller, S., Davenport, T. (2020). *Machine learning support for cyber threat attribution at FireEye*. FireEye. https://www.fireeye.com/blog/products-and-services/2020/06/machine-learning-support-for-cyber-threat-attribution-at-fireeye.html.

MITRE. (2017). CVE-2017-0145 Detail. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0145.

MITRE. (2020). CVE-2020-14007 Detail. https://nvd.nist.gov/vuln/detail/CVE-2020-14007#vulnCurrentDescriptionTitle.

MITRE. (2021a). Lazarus Group, HIDDEN COBRA, Guardians of Peace, ZINC, NICKEL ACADEMY, Group G0032 | MITRE ATT&CK®. https://attack.mitre.org/groups/G0032/.

MITRE. (2021b). APT29 | MITRE ATT&CK®. https://attack.mitre.org/groups/G0016/.

Mockus, A., & Weiss, D. (2002). Predicting risk of software changes. *Bell Labs Technical Journal, 5*(2). 169-180. https://doi.org/10.1002/bltj.2229.

Mohamed. (2017). *Wanna cry ransomware : Update 5/21/2017 fix. Wanna Cry Ransomware : Update 5/21/2017 FIX - Microsoft Community*. Microsoft. https://answers.microsoft.com/en-us/windows/forum/windows_10-security-winpc/wanna-cry-ransomware-update-5212017-fix/5afdb045-8f36-4f55-a992-53398d21ed07.

Mott, N. (2021). *Countless servers are vulnerable to apache log4j Zero-Day exploit*. PCMAG. https://www.pcmag.com/news/countless-serves-are-vulnerable-to-apache-log4j-zero-day-exploit.

Muggler, M., Eshwarappa, R., & Cankaya, E. C. (2017). Cybersecurity management through logging analytics. *International Conference on Applied Human Factors and Ergonomics*. 3-15. Springer, Cham.

NCSC. (2021). *Mitigating malware and ransomware attacks*. ncsc.gov.uk. https://www.ncsc.gov.uk/guidance/mitigating-malware-and-ransomware-attacks.

NIST. (2022). *Publications.* Computer Security Resource Center. https://csrc.nist.gov/Publications.

OASIS Cyber Threat Intelligence Technical Committee. (2021). *Introduction to TAXII*. Oasis Open. https://oasis-open.github.io/cti-documentation/taxii/intro.html.

Paulsen C., McDuffie E., Newhouse W., & Toth P. (2012). NICE: Creating a cybersecurity workforce and aware public. *IEEE Security & Privacy, 10*(3), 76-79. doi:10.1109/MSP.2012.73.

Peisert, S., Schneier, B., Okhravi, H., Massacci, F., Benzel, T., Landwehr, C., ... & Michael, J. B. (2021). Perspectives on the SolarWinds incident. *IEEE Security & Privacy, 19*(02), 7-13.

Qamar, S., Anwar, Z., Rahman, M. A., Al-Shaer, E., & Chu, B. (2017). Data-driven analytics for cyber-threat intelligence and information sharing. *Computers & Security, 67*, 35-58. doi:10.1016/j.cose.2017.02.005.

Rose, S., Borchert, O., Mitchell, S. & Connelly, S. (2020). Special Publication 800-207: Zero trust architecture. National Institute of Standards and Technology, Gaithersburg, MD. https://doi.org/10.6028/NIST.SP.800-207.

Rodriguez, A., & Okamura, K. (2020). Enhancing data quality in real-time threat intelligence systems using machine learning. *Social Network Analysis and Mining.*, *10*(1). https://doi.org/10.1007/s13278-020-00707-x.

Skopik, F., & Pahi, T. (2020). Under false flag: Using technical artifacts for cyber attack attribution. *Cybersecurity, 3*(1). doi:10.1186/s42400-020-00048-4

SolarWinds. (2021). *Security Advisory*. SolarWinds. https://www.solarwinds.com/sa-overview/securityadvisory

Sumo Logic. (2020). *Defense in depth: Doublepulsar*. Sumo Logic. https://www.sumologic.com/blog/defense-in-depth-the-equation-group-leak-and-doublepulsar/

ThreatConnect. (2021). *STIX-TAXII*. ThreatConnect, Inc. https://threatconnect.com/stix-taxii/.

Townsend, A., McBride, T., Lusty, L., Sexton, J. & Ekstrom, M. (2020). Special Publication 1800-11: Data integrity recovering from ransomware and other destructive events. National Institute of Standards and Technology, Gaithersburg, MD, https://doi.org/10.6028/NIST.SP.1800-11

United States Federal Bureau of Investigation, United States Cybersecurity and Infrastructure Security Agency, United States National Security Agency, Australian Cybersecurity

Center, & United Kingdom National Cyber Security Centre. (2022). *2021 Trends Show Increased Globalized Threat of Ransomware.* Joint Cybersecurity Advisory. https://www.ic3.gov/Media/News/2022/220209.pdf

USC. (2021). *Organizing your social Sciences research PAPER: Types of research designs*. LibGuides. https://libguides.usc.edu/writingguide/researchdesigns

Wiley, B. (2021). *Aquatic panda in possession of Log4Shell exploit tools*. CrowdStrike. https://www.crowdstrike.com/blog/overwatch-exposes-aquatic-panda-in-possession-of-log-4-shell-exploit-tools/

Williams, J. (2021). *What You Need to Know About the SolarWinds Supply-Chain Attack*. SANS Institute. https://www.sans.org/blog/what-you-need-to-know-about-the-solarwinds-supply-chain-attack/

Wortley, F., Thompson, C., & Allison, F. (2021). *Log4Shell: RCE 0-day exploit found in log4j 2, a popular Java logging package*. LunaSec. https://www.lunasec.io/docs/blog/log4j-zero-day/

Yan, T., Deng, Q., Zhang, H., Fu, Y., Grunzweig, J., Harbison, M., & Falcone, R. (2022). *Apache log4j vulnerability CVE-2021-44228: Analysis and mitigations.* Palo Alto Networks. https://unit42.paloaltonetworks.com/apache-log4j-vulnerability-cve-2021-44228/

Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2019). An efficient intrusion detection system based on feature selection and ensemble classifier. *Journal of LaTex Class Files*, *14*(8). https://www.researchgate.net/publication/332168887_An_Efficient_Intrusion_Detection_System_Based_on_Feature_Selection_and_Ensemble_Classifier.