Computer and Information Science; Vol. 17, No. 2; 2024
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Proactively Defending Enterprise Computer Systems Against Threats and Vulnerabilities

Faris Sharaf[1], Abdullah Alhayajneh[2], & Thaier Hayajneh[1]

[1] Fordham Center for Cybersecurity, Fordham University, New York, USA

[2] Department of Professional Security Studies, New Jersey City University, New Jersey, USA

Correspondence: Thaier Hayajneh, Fordham Center for Cybersecurity, Fordham University, New York, USA. Tel: 1-212-636-7785 E-mail: thayajneh@fordham.edu

Received: May 23, 2024 Accepted: July 19, 2024 Online Published: September 6, 2024

doi:10.5539/cis.v17n2p28 URL: https://doi.org/10.5539/cis.v17n2p28

Abstract

Cybersecurity remains a critical concern, even amidst global events like the COVID-19 pandemic. The rise of COVID-19 has deepened cybersecurity threats, with phishing emails and phone scams attempting to exploit the situation. This paper focuses on proactive strategies to protect enterprise environments against threats and vulnerabilities, specifically in Windows-based systems. Tracing threats and vulnerabilities to their source aims to address them in their early stages rather than after an attack has occurred. This research tackles three prevalent issues: phishing emails, vulnerability patching, and industrial internet-connected devices. Through analyzing various cyber defense models and vulnerability databases, this paper proposes frameworks to mitigate these issues effectively. The study includes a detailed examination of sources of threats and vulnerabilities, aiming to develop methodologies for practical implementation. Ultimately, the goal is to summarize best practices to enhance tool utilization and process improvement and propose new proactive defense methods. The research emphasizes the shift from reactive to proactive defense strategies to better protect enterprise networks.

Keywords: proactive, enterprise, threats, vulnerabilities, computer systems

1. Introduction

Today's technology has become an indispensable part of our daily lives, comparable to essential needs such as food and water. While technology offers numerous benefits, it also introduces vulnerabilities, especially in large-scale environments like Microsoft Windows enterprise systems. Protecting these environments from potential threats that could exploit existing vulnerabilities is crucial. Effective protection strategies must address internal and external threats to maintain the integrity and security of the enterprise network. This protection can range from internal safeguards, such as a phone passcode, to comprehensive measures that protect a company's network inside and out. The fundamental principle is to proactively safeguard against threats that could exploit vulnerabilities in the system.

Enterprise networks predominantly consist of computing machines running Microsoft Windows operating systems. While these systems are widely used, they are not without flaws, resulting in internal vulnerabilities and potential external and internal threats. A significant aspect of IT operations involves safeguarding these assets to ensure uninterrupted operations. When an intrusion occurs, alerts are triggered, and respective teams work to contain the damage and restore normal operations. However, there are strategies to mitigate threats at earlier stages, which can prevent significant damage and reduce the need for extensive recovery efforts. This approach, known as proactive defense, is constantly evolving and improving. By expanding our understanding of the threat landscape, its origins, and its industry-specific nature, we can better manage risks in advance and effectively control threats and vulnerabilities.

Organizations of all sizes should strive to exceed minimum security requirements. The Target data breach in 2013, despite the company's PCI-DSS compliance, revealed deficiencies in its network defense capabilities (Arbaugh, Fithen, & McHugh, 2000). Threat actors are relentless and indiscriminate, continuously seeking entry points to exploit, regardless of the target's size. Therefore, the notion that any business is completely safe is a misconception. As people are often the source of threats and vulnerabilities, it is impossible to eliminate them, but we can reduce their effectiveness. This paper presents an approach to mitigate these human-induced threats and vulnerabilities.

The paper is organized as follows. Section II covers related works to introduce the motivation behind this research. Section III provides background information, focusing on two main ideas: enterprise IT threats and vulnerabilities and proactive defense strategies. Section IV outlines the goals and objectives and delves into corporate environment threats and vulnerabilities. This section also analyzes the sources and databases of information and the tools and methodologies used. The results and expected outcomes of this research are presented in Section VI, and finally, the paper concludes in Section VII.

2. Material Studied

Research into the threats and vulnerabilities of computer systems continues to grow due to its evolving nature and significant economic impact on organizations (Paller, 2006). Exposure to enterprise environments, along with understanding IT operations, issues, and experiences with patching, has contributed to a comprehensive knowledge base. This accumulated expertise enables analyzing existing methods, identifying their shortcomings, and proposing improvements or new solutions using current tools. For instance, emails from SCCM administrators regarding proposed patches, deployment schedules, and related vulnerabilities have been instrumental in identifying the most critical flaws affecting Windows enterprise systems.

Building on previous work, this research incorporates additional background expertise to provide a solid understanding of contemporary issues and the underlying problems. Advanced studies in Cyber Security have facilitated hands-on practice with exploits and payloads targeting operating systems like Microsoft Windows. Experience in penetration testing, intrusion detection, and wireless security has further reinforced the knowledge required to address these challenges. This endeavor highlights current limitations in combating threats and vulnerabilities and underscores the necessity for new methodologies, detailing potential innovations and implementations.

This research utilizes several methods, including SCCM, Sandbox environments, and Oracle VirtualBox VMs with Kali Linux and Windows operating systems. Sources such as SANS, CVE details, OWASP Top 10, and exploit- db were employed to address threats and vulnerabilities. Cross-operating systems and penetration testing were conducted within a controlled virtual machine environment, incorporating Kali Linux, Metasploit, and Windows operating systems. Although cross-operating systems testing holds significant promise and is expected to yield positive results, the primary focus remains on conducting thorough analyses, comparisons, and deriving meaningful findings. The intended tool may not be fully developed, making it challenging to prove its reliability and effectiveness at this stage definitively.

Another method involves running a Nmap scanner against various versions of Windows 10 software to identify potential exploits, particularly in outdated software. The feasibility of these exploits is then assessed to determine their practicality, execution, and potential to cause damage. In this context, feasibility refers to whether the exploit is authentic and practical, if it can be successfully executed, and if it can indeed inflict damage, given the necessary extent and skill level. A vulnerability scanner will also identify exploitable vulnerabilities in programs or web applications and corresponding recommended countermeasures. Many vulnerability scanning tools target web-based applications and internet information services, detecting issues such as invalid or outdated certificates. For example, OpenVas is a vulnerability scanner that identifies security issues in servers and network devices. This free tool, available in Kali Linux, detects vulnerabilities and provides recommended safeguards. This study will run a vulnerability scanner against Windows within a virtual machine to analyze the results and identify potential improvements or limitations in the scan test (Allodi, Luca, & Fabio 2012). Another method involves examining SCCM patching to identify existing patches for specific machines. The computer name is entered in the machine/IP address field to conduct a test, and connectivity is checked to determine if the machine is online. If the machine is online, applicable patches are identified and installed. Machines that have been recently imaged or offline for an extended period will require multiple patches to be installed.

Sandbox manipulation may be limited due to restricted access and the inability to create or configure one according to specific requirements during testing. In this case, Cuckoo Sandbox will be utilized and studied to gather as much information as possible about our company's Sandbox system. The goal is to determine if it can be deployed near a subnet for groups such as Accounts Receivable, who need to receive multiple vendor attachments as part of their daily routines. Another use case is to open attachments and links within the Sandbox for triage purposes rather than on the user's computer. Additionally, an alternative approach could involve deactivating attachments and links and displaying their details or the actual URL behind the link, similar to hovering over it. This ensures that users log into the Sandbox environment to check attachments or manually visit the site rather than accessing the information through email shortcuts.

In this paragraph, several tools are listed to address information gaps. The Cyber Kill Chain (CKC) model will be

analyzed to develop an improved version. Additionally, various vulnerability and threat database sources will be compared and contrasted, including OWASP Top Ten, SANS AtRisk, exploit-db, and CVE details. These sources will aid in generating a web-based engine to scan the internet for organizational devices connected to the network (Wichers & Dave, 2013). This engine will monitor, alert the security team, and provide a report with details such as site name, statistics, discovery time, and method. Furthermore, it will generate device information within the network, including usage status, purpose, and owner, similar to the Shodan monitor but with enhanced functionalities.

3. Background

3.1 Proactive Defense

Reactive network defense and protection methods are still widely used but are now considered traditional due to advancements in the threat landscape and the surge in hacks and breaches reported daily. It is important to note that when referring to enterprise environments, we predominantly discuss Microsoft Windows operating systems, including user machines and servers. Hence, Windows will be assumed and not explicitly mentioned unless necessary. Reactive defense primarily involves dealing with and containing incidents after they occur. In contrast, proactive models, such as the Lockheed Martin Cyber Kill Chain (CKC), focus on defending against attacks before they happen. This research article emphasizes proactive defense methods. The CKC model extends beyond reactive measures to study and understand threats like Advanced Persistent Threats (APTs), enabling early-stage responses. This model comprises an eight-phase attack lifecycle, where stopping an attacker at any stage prevents further penetration by disrupting the normal attack cycle. The CKC is fundamentally an attack model (https://www.cvedetails.com/cve/CVE-2019-1222/)

3.2 Threats and Vulnerabilities

Threats and vulnerabilities have long been the bane of IT security teams and organizations. Risk management aims to identify, evaluate, prioritize, and control risk to an acceptable level. This paper proposes the hypothesis that vulnerabilities attract threats rather than the reverse. Threat actors and agents may exist but typically do not activate or tailor an attack unless there is an exploitable vulnerability. These actors constantly seek flaws and weaknesses to exploit, regardless of the organization's size. The threat landscape is extensive. Organizations must look beyond internal techniques to mitigate risks effectively and collaborate with others, especially within the same industry, as threats can be industry-specific. Thinking innovatively, staying updated with the latest cybersecurity news, and going beyond merely meeting compliance standards is essential. Vulnerabilities must be closely monitored and promptly addressed (Schmidt & White, 2017). If security flaws are not embedded into the software design, it should become a mandatory requirement within the company, if not a government regulation. One of the primary ideas of this research is that vulnerabilities attract threats. Threat actors must find a flaw to exploit, and a vulnerability provides that opportunity. Threat actors or agents may already exist but remain dormant until they identify a vulnerability. Once they find an exploitable weakness, they become active. Figure 1 below proposes a model for the relationship lifecycle between threats and vulnerabilities.

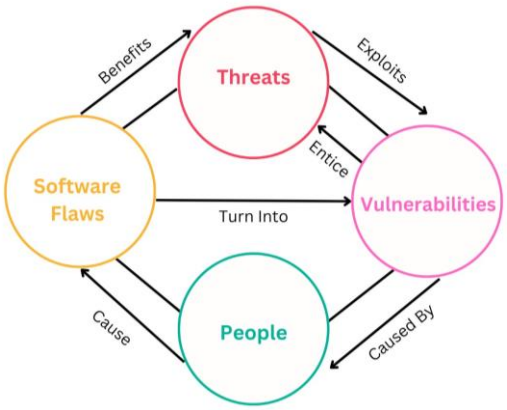

Figure 1. Threat and Vulnerability Relationship Cycle

3.3 Goals

Current models and trends in organizational protection are not promising. With rapid advancements and pervasive methods in the threat landscape, enterprise defense systems continue to struggle with network breaches.

As breaches frequently make headlines, it is clear that we are falling behind and need to enhance our approaches. It is imperative to shift from reactive to proactive defense. This project aims to develop proactive defense models and frameworks based on current challenges and other risk management methods. Our proposed model relies on root cause analysis, identifying the source of the problem to prevent it in the future. This approach advocates for proactive network defense strategies rather than adhering to existing reactive defense methodologies. Addressing breaches and incidents after they occur is insufficient. Once a threat agent infiltrates the system, there is no guarantee it can always be stopped. Therefore, it is crucial to prevent attacks in enterprise networks at early stages or before they happen.

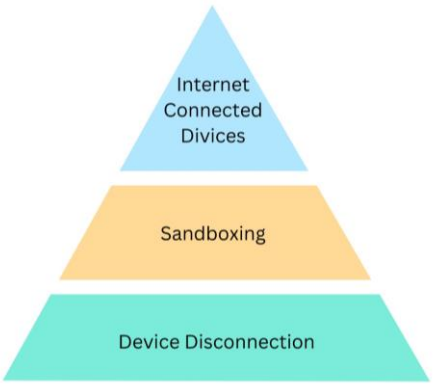

Figure 2. Proactive Defense Levels

3.4 Proactive Against Threats & Vulnerabilities

Many organizations are lagging in terms of incident response time and the detection of suspicious network activity. According to the "Detecting the Deceivers through Deception" webinar hosted by Kevin Fiscus and Tony Cole on April 8, 2020, it takes approximately 279 days to detect or respond to a breach (Brickell, Hall, Cihula, & Uhlig, 2006). Additionally, organizations have an average of 9 hours to stop a breach before a sophisticated attacker moves laterally, establishes more beachheads, and reaches their objective. The average global dwell time before an adversary is detected in an environment is 56 days. The speakers also highlighted that 40 percent of vulnerabilities have available fixes but remain unpatched. The defense methods described here follow a hierarchical format, becoming more proactive as the levels progress. For instance, patching vulnerabilities effectively protects against attacks exploiting software vulnerabilities. However, even before reaching that level, we could have prevented the intrusion by stopping the execution of a malicious email link or attachment. To be even more proactive and extend beyond the enterprise perimeter, we could have limited the information available on the internet that attackers use to initially infiltrate our network. Passive reconnaissance leads to active reconnaissance. A monitoring tool could be set up to block internet scanners, monitor internet-connected devices, and send reports.

Some organizations have IT staff with various security certifications, yet they regularly experience network breaches. Software executing many of these attacks is readily available online (Prevelakis & Spinellis, 2001). This situation indicates the need for alternative approaches to information security. Beyond certifications, what is truly needed are security-savvy, experienced, and versatile IT security personnel combined with a practical and enforced set of policies, processes, and procedures (Brinkley et al., 2013). The importance of proficient IT security staff is akin to having skilled doctors; mistakes in this field are costly. Security is not merely about writing a few lines of access control code on a firewall and then neglecting it—this approach is called "set and forget." Firewalls and IDS alone are insufficient; 99% of intrusions result from exploiting known vulnerabilities or configuration errors where countermeasures were available (Onwubiko, Cyril, & Lenaghan, 2007).

4. Methods

Several sources provide reliable information on various operating system vulnerabilities, including CVE details, exploit-db, and the SANS AtRisk newsletter. The latter offers a weekly summary of newly discovered attack vectors, vulnerabilities with active exploits, and explanations of recent attacks. SANS, which stands for Sysadmin, Audit, Network, Security, keeps users informed about new attacks, regulations, and the most critical security vulnerabilities discovered each week (Harris, Brendon, & Hunt, 1999). The newsletter includes two key sections: "Top Vulnerability This Week" and "Recent Vulnerabilities for Which Exploits Are Available." In this study, the SANS newsletter is used to confirm the presence of vulnerabilities in Microsoft Windows operating system enterprise environments. A commonality among these vulnerabilities is remote code execution (Gerhard, 2005).

Another valuable source of free CVE security vulnerability information is CVEdetails.com. CVE stands for Common Vulnerability and Exposures. The site lists vulnerability details, exploits, references, and Metasploit modules. Analysis of this resource reveals that a significant percentage of Microsoft vulnerabilities involve code execution, and the trend indicates an exponential growth in the number of vulnerabilities over time. For example, in 1999, there were 172 vulnerabilities, 42 related to code execution; by 2019, there were 668 vulnerabilities, with 42 being code execution vulnerabilities. Code execution attacks are the most prevalent among other vulnerabilities, such as denial of service and overflow. In 2019, while no exploits were available for the 270 vulnerabilities found, many had a criticality score exceeding 9, indicating high severity. The data on this site originates from the National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) and exploit-db, another source to be discussed. Threat actors, adversaries that send malicious emails, exploit these vulnerabilities, starting with vulnerable users and extending to unpatched software. This study uses CVEdetails.com for statistical analysis of vulnerability trends. The exponential growth of vulnerabilities (VVV) over time (ttt) is modeled by V(t)=V0ektV(t) = V_0 e^{kt}V(t)=V0ekt, where V0V_0V0 is the initial number and kkk is the growth rate. A prioritization algorithm is proposed, involving data collection, categorization, scoring, and mitigation strategies, focusing on the most critical vulnerabilities to enhance security.

Similar to NVD, exploit-db is another essential source for vulnerability research that includes a risk score as a standard measure. It lists vulnerabilities across multiple platforms, such as Windows, hardware, and Linux, indicating whether exploits are downloadable, verified, and if the application is vulnerable. For instance, filtering by the Windows platform yields 9,722 exploits out of 42,558. Further filtering by "Verified" and "Has Application" reduces this to 2,032. Downloading options include Python files for exploits and actual application downloads. However, only about a quarter of Windows exploits are downloadable with a related vulnerable application, limiting the database's reliability. Additionally, exploit-db and NVD are not entirely reliable sources of exploit information, as per ACM publication findings (Suprio, 2011).

Another valuable information source is the OWASP Top 10, provided by the Open Web Application Security Project, a global non-profit online community that publishes free articles about web application security. The OWASP Top 10 is an awareness document listing the top 10 critical web application vulnerabilities most commonly exploited by hackers, along with recommendations for addressing these risks (Arora et al., 2004). These vulnerabilities are categorized broadly rather than listed as specific CVEs, and their order may change periodically. Common vulnerabilities include SQL injection, sensitive data exposure, and cross-site scripting. For example, the 2017 report maintained injection as the top vulnerability, consistent with the 2013 report. Remediation measures for these vulnerabilities include stored procedures, input escaping, and whitelists for server-side input validation (Lippmann et al., 2002)

5. Results

5.1 Proposed Prototype Sandbox Framework

The threat and vulnerability information sources mentioned earlier are integral tools used in this study. The Cyber Kill Chain (CKC) is reiterated here to utilize these tools to provide necessary input and lay the groundwork for the proposed frameworks. Kali Linux and the VirusTotal website were used to demonstrate how a malicious PDF can infect a computer once opened. This preparation enables us to delve into our frameworks, beginning with proactive protection against email links and attachments. The subsequent topics will discuss the proposed solutions for patching issues limiting the propagation of internet-connected device information and understanding the impact of receiving an email with a suspicious link or attachment in a corporate network. Even experienced IT staff can fall victim to phishing attempts, as evidenced by a tech support manager who fell for one of these phishing emails and opened the attachment.

Even a high-ranking executive in our department, such as the Vice President, fell victim to clicking on a phishing link or attachment. Although it was a legitimate phishing test conducted by our cybersecurity team, the incident underscores the risk. This is despite the company maintaining a high standard of safety and security practices, including comprehensive cybersecurity awareness and education courses and regular assessments. While these measures are effective, and cybersecurity awareness is a crucial protection practice, incidents still occur. This indicates a need for a more proactive approach. It is essential to reconsider external email policies and evaluate their necessity based on the business's specific needs. This proactive stance is what we will explore next.

A proposed Sandbox framework diagram in an article outlined a method where, if a file were deemed suspicious, it would be sent to a sandboxing environment; otherwise, it would proceed to the user’s computer. However, this approach has two significant flaws based on practical experience. First, a file might appear safe and trick the system into accepting it, thereby infiltrating the environment. For instance, when a malicious PDF file is uploaded

to VirusTotal (at VirusTotal.com), it generates alerts. However, if the same file is compressed and then uploaded, VirusTotal may fail to recognize it as suspicious and mark it as safe across various virus definition engines.

Second, even if file scanners are present, they might only operate occasionally. There have been instances where opening an attachment resulted in a notification from Google email that file scanning was unavailable or inactive, prompting the user to proceed at their own risk. Despite Gmail's generally robust handling of email attachments, technology is not always 100% reliable. A more proactive solution involves leveraging the Cyber Kill Chain (CKC). The proposed framework focuses on phase 3, delivery. By disengaging the attack early at this stage, where the delivery gets lost in transit, the framework prevents the attack from progressing to phase 4, exploitation.

Before considering the complete blocking of external emails, businesses should assess their specific needs and reconsider the necessity of allowing all emails. Implementing an implicit deny principle, where only pre-defined, necessary business emails from third parties are allowed, could be more effective. All other emails should be directed to the user's personal inbox. External emails should go to personal emails unless users coordinate with IT to pre-register or discuss the need for a particular email domain, such as emails from Vanguard, a 401k and pensions company, to be directed to their business inbox. Vendors with whom the business has ongoing relations should be identified and marked in advance rather than allowing random emails to go directly to the business inbox. Although some of these emails might be legitimate, a more structured approach is necessary, especially for departments like Accounts Payable, which deal with vendors daily and frequently receive email attachments as part of normal business operations.

A user from the Accounts Payable section reported that their group often struggles with an influx of external emails with attachments. They forward suspicious emails to the email check group for verification but sometimes do not receive timely responses, leaving them to decide on the safety of the emails independently. This situation underscores the need for a more efficient system to manage and verify external emails. They often have to determine if a vendor is familiar with and whether to accept the attachment or even the email. A control should be in place to identify the vendor since a business contract was established at some point. However, this responsibility often falls on the Purchasing side. The Accounts Payable group has experienced phishing incidents and struggles with emails from multiple contacts within the same vendor. They should focus on the domain after the "@" sign, ensuring it matches the expected vendor domain, regardless of the specific sender's name. For example, different names before the "@" sign should still have the same vendor email domain, such as @contoso.com.

A comprehensive framework can be developed to address various business scenarios and requirements in a single diagram. This Sandbox approach differs from the previous one. In this model, all external emails are directed to the Sandbox, regardless of whether they contain links or attachments or appear suspicious. The only exceptions are emails from the company's defined email domain or pre-approved external emails. This flexible setup can be adjusted as needed and may not create significant overhead for companies with minimal external email traffic. However, for larger organizations, refined controls can still be effectively implemented.

The diagram includes numbered callouts to explain each step. Starting with [400], the framework begins by asking if external emails are allowed. If not, no further action is needed [401]. If yes, diamond [402] checks if the email is from the company's domain. If it is, it is allowed [403]. If not, it asks if it is an approved external email [404]. If the external email is not approved, it automatically goes to the sandbox [405], regardless of its content. Diamond [406] determines if the external email's attachment or link is safe. If unsafe, the email is rejected [407]. If deemed safe, IT security has three options [408] for handling it.

First, the emails can remain in a sandbox or a virtual environment where users log in to access the links and attachments, even if they are tested to be safe [409]. Second, a software or hardware-based email filter can be implemented to release the email but with inactive links and attachments. This device or service can display the actual link behind the embedded link as if the user is hovering over it. This measure ensures that users are aware of the real destination of the link, preventing phishing attempts where a link displays as a legitimate site but redirects to a malicious one [410]. Third, the email can be released with its links and attachments fully active, as it has been verified as safe [411]. This approach provides a robust and flexible framework to enhance email security, ensuring that external emails are managed effectively and potential threats are mitigated.

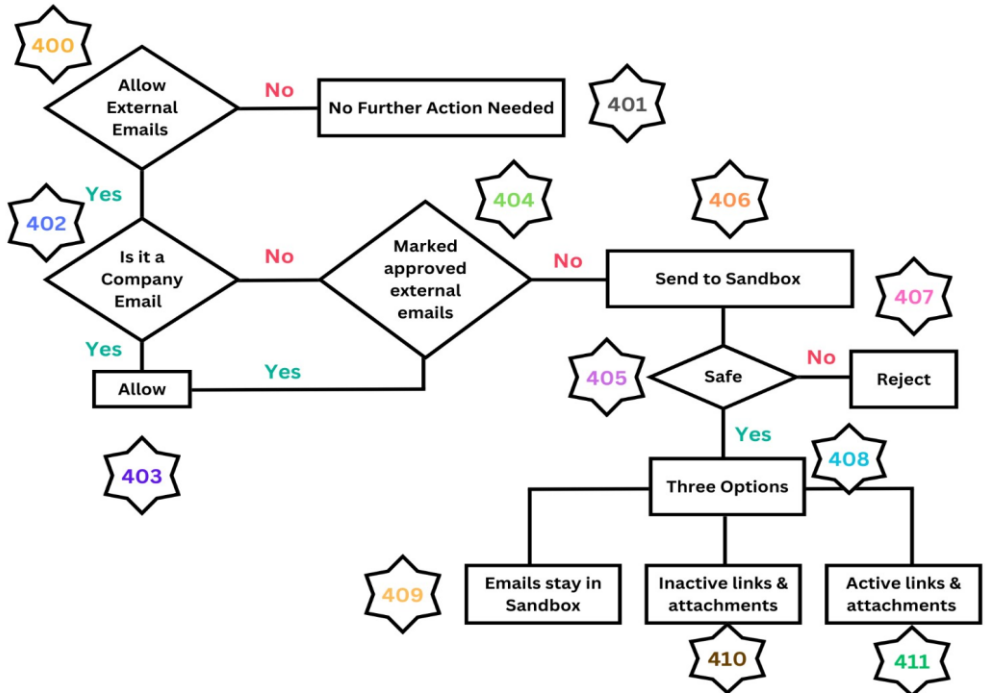

Figure 3. External Emails and Phishing Threat Prevention

5.2 Vulnerability Patching

On the second Tuesday of every month, Microsoft releases patches for Windows operating systems and related applications. While some companies manage their patching cycles effectively, many others lag behind and face various challenges in patch management. According to "The Conversation" (Wang et al., 2004), breaches often occur due to software vulnerabilities that require fixing. Frequently, articles about breaches reveal that despite defenses, organizations fell short due to ignored known weaknesses. For example, in the Target breach case, an outdated web browser ActiveX add-on/plug-in was exploited. Remote code execution is one of the most common vulnerability categories, so patching affected software is crucial. It is important to note that no solution is perfect or suitable for all scenarios.

5.2.1 Issues

While development and operations may simplify the process of regular and prompt patches (Yadav, Tarun, & Mallari Rao, 2015), other issues related to the computer systems themselves still exist. The article “Shield: Vulnerability-Driven Network Filters for Preventing Known Vulnerability Exploits” proposes a solution called Shield Filter that corrects traffic exploiting vulnerable applications (Lee et al., 2017). It suggests that these filters are easier to install and test than patches. However, patches are eventually necessary. While easier to install, Shield Filters may create traffic or administrative overhead since they are host-based on every machine. Even companies with good patch management, such as the one in this study, encounter shortcomings. For instance, SCCM administrators send reports to LAN admins about outdated or outdated systems. This manual process can only be effective if LAN admins address the issues promptly, as there is no strict schedule for completion and feedback. Additionally, disconnecting devices from the network via the NetMotion console was previously a manual process applied only to mobile devices. Server-side patching may be effective, but issues arise when the SCCM client on a computer is outdated or unreachable, even if the device is online. This is particularly common with portable devices, such as laptops, due to VPN applications not connecting or reporting back to the network correctly, a situation frequently encountered in this corporate environment.

5.2.2 Proposed Solution

This issue leads to a client-side solution, particularly relevant for mobile devices. The proposal involves having an SCCM client or agent on the computer disconnect the device if it is not in compliance or outside the patching cycle. A prompt would be displayed on the screen, instructing the user to contact the IT help desk to have a LAN admin update the system. A simple diagram (Figure 4) illustrates this process. The tool ensures that tech support proceeds only once the SCCM client is updated and the computer reports back to the network as expected. This

guarantees the device is reachable and ready to receive, handle, and install its assigned patches, leaving no room for oversight and forcing computers to stay updated and connected to the network. In the current network environment, a computer is automatically removed from the domain if it has not been logged in for more than 30 days. However, this solution addresses the need for compliance before the 30-day period, ensuring the computer is protected against attacks. Users must be online to take action and call for support to get back to business. The machine remains safe in this disconnected state, mainly if it is not being used.

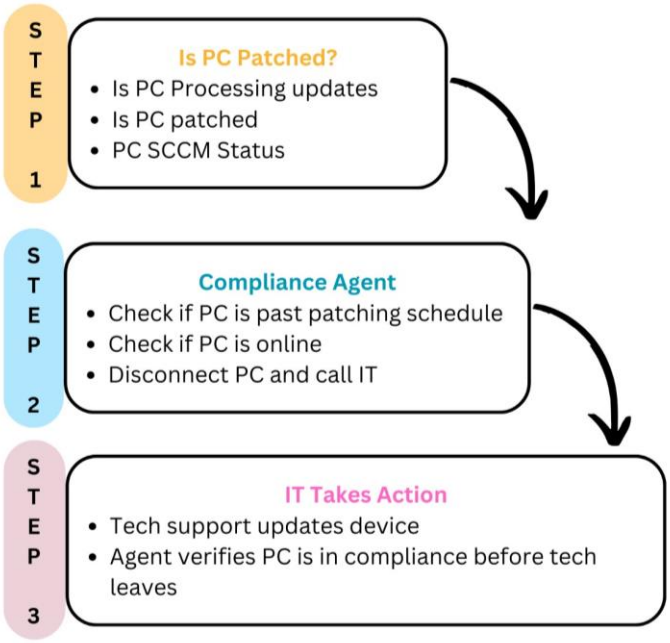

Figure 4. PC-Based SCCM Agent Enforces System Patching

5.3.1 Prototype

Internet-connected device search engines like Shodan can search and display device names and related data, such as IP addresses, when a device name is entered. While this capability aids in finding internet-connected devices, it also provides attackers with a powerful reconnaissance tool (Huddleston & David E., 2010). Furthermore, Shodan can now monitor these devices and send alerts. Our proposed tool (Figure 5) is similar but offers additional features. It alerts users and takes action by either blocking device information from leaving the network or preventing it from being accessed on the web. More importantly, it can be configured to act as a decoy, sending false or masked device information to mislead adversaries into believing they have obtained valuable data. The alert logs generated by the tool will include additional information such as IP addresses, device owners, the date the device was found on the internet, and recommendations for mitigating data or device information exposure. This solution can be implemented as either a hardware or software-based system. Figure 5 presents a hierarchical diagram outlining its various functions.

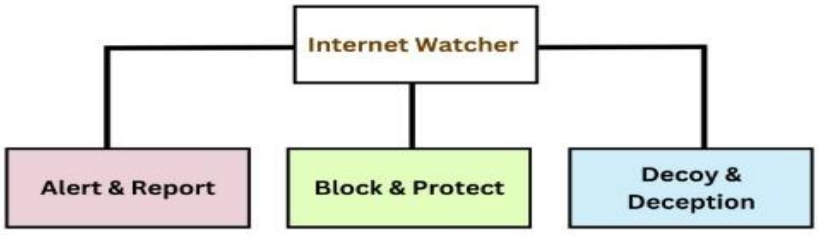

Figure 5. Internet Watcher Tools and Functions

6. Discussion

People are inherently among the sources of threats and vulnerabilities, making it impossible to eliminate them; instead, their effectiveness must be reduced. In reality, removing threat actors is impractical. It is not feasible to

expect governments to negotiate with unethical hackers or offer them jobs to deter their activities, as was the case with the individual who hacked NASA and was later hired by the agency. A more proactive approach would involve blocking illegitimate sites that are unauthorized to host exploits or hacking tools, restricting access to a few authorized sources, and preventing their widespread availability to internet users.

When the red team (attackers) and blue team (defenders) are equal in power, the red team often wins by using deception (Teri, 2014). This highlights the importance of developing strategies that go beyond mere compliance. The concept of compliance versus commitment is crucial. Instead of treating cyber hygiene and security practices as mere compliance tasks, which often lead to minimal effort to meet requirements, these practices should become a commitment embedded in the business culture. This means integrating cyber hygiene into daily operations so that actions like changing default passwords to stronger ones become standard practice without needing enforcement. Device manufacturers should also play a role by making the recommendation to change default passwords mandatory through technical controls that enforce password changes upon setup. Ultimately, security is everyone's responsibility. Cybersecurity awareness and education are invaluable for both individuals and organizations, fostering a culture of proactive and committed cybersecurity practices.

7. Conclusion

This research has comprehensively analyzed cybersecurity threats and vulnerabilities, presenting three targeted solutions to address the most pressing issues organizations encounter. Phishing and patching are two of the most significant challenges, requiring strategic and proactive measures. Additionally, a proposed mitigation framework has addressed the exposure of devices to the internet and the consequent risks posed by internet scanners and web applications. Figures 1 and 2 serve as hypothetical scenarios to illustrate the context of these issues, while Figures 3, 4, and 5 offer detailed frameworks and models specifically designed to tackle these challenges. These practical and feasible solutions promise substantial improvements in current cybersecurity practices.

To conclude this research emphasizes the critical need for proactive cybersecurity measures and continuously enhancing security protocols. Adopting the proposed frameworks can significantly strengthen organizational defenses against the ever-evolving threat landscape. By implementing these strategies, organizations can address current vulnerabilities and pave the way for future refinements in cybersecurity. This study highlights the imperative of ongoing vigilance and innovation, underscoring the commitment to maintain robust cybersecurity defenses.

Acknowledgments

Not applicable.

Authors' contributions

Mr. Faris Sharaf, was responsible for study design and data collection, and drafted the manuscript. Dr. Abdullah Alhayajneh, & Dr. Thaier Hayajneh supervised the project, edited the paper, and revised the manuscript to address all the comments.

Funding

Not applicable.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Informed consent

Obtained.

Ethics approval

The Publication Ethics Committee of the Canadian Center of Science and Education.

The journal's policies adhere to the Core Practices established by the Committee on Publication Ethics (COPE).

Provenance and peer review

Not commissioned; externally double-blind peer reviewed.

Data availability statement

The data that support the findings of this study are available on request from the corresponding author. The data

are not publicly available due to privacy or ethical restrictions.

Data sharing statement

No additional data are available.

References

Allodi, L., & Massacci, F. (2012). *A preliminary analysis of vulnerability scores for attacks in wild: the ekits and sym datasets.* In Proceedings of the 2012 ACM Workshop on Building analysis datasets and gathering experience returns for security (pp. 3-12). ACM. https://doi.org/10.1145/2382416.2382427

Arbaugh, W. A., Fithen, W. L., & McHugh, J. (2000). Windows of vulnerability: a case study analysis. *Computer, 33*(12), 52-59. https://doi.org/10.1109/2.889093

Arora, A., et al. (2004). *Impact of vulnerability disclosure and patch availability: An empirical analysis*. In Third Workshop on the Economics of Information Security (Vol. 24).

Brickell, E. F., Hall, C. D., Cihula, J. F., & Uhlig, R. (2006). *Method of improving computer security through sandboxing.* Google Patents. Retrieved from https://patentimages.storage.googleapis.com/fc/70/93/9626e7cea48016/US7908653.pdf

Brinkley, M. D., & Permeh, R. R. (2013). *Application sandboxing using a dynamic optimization framework.* U.S. Patent No. 8,590,041. November 19.

Eschelbeck, G. (2005). The laws of vulnerabilities: Which security vulnerabilities really matter? *Information Security Technical Report, 10*(4), 213-219. https://doi.org/10.1016/j.istr.2005.09.005

Harris, B., & Hunt, R. (1999). TCP/IP security threats and attack methods. *Computer Communications, 22*(10), 885-897. https://doi.org/10.1016/S0140-3664(99)00064-X

Huddleston, D. E. (2010). *Method and system for isolating suspicious email.* U.S. Patent No. 7,832,012. November 9.

Lee, S., Shin, S. H., & Roh, B. H. (2017). *Abnormal behavior-based detection of Shodan and Censys-like scanning.* In 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN) (pp. 326-331). IEEE. https://doi.org/10.1109/ICUFN.2017.7993960

Lippmann, R., Webster, S., & Stetson, D. (2002). The effect of identifying vulnerabilities and patching software on the utility of network intrusion detection. In International Workshop on Recent Advances in Intrusion Detection (pp. 371-386). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-36084-0_17

Onwubiko, C., & Lenaghan, A. P. (2007). *Managing security threats and vulnerabilities for small to medium enterprises.* In 2007 IEEE Intelligence and Security Informatics (pp. 13-18). IEEE. https://doi.org/10.1109/ISI.2007.379479

Pal, S. (2011). *Downtime reduction for enterprise manager patching.* U.S. Patent Application 12/634,518, filed June 9.

Paller, A. (2006). Utilizing SANS free resources to improve your internet security posture. *IEEE Communications Magazine, 44*(7), 17-17. https://doi.org/10.1109/MCOM.2006.1668374

Prevelakis, V., & Spinellis, D. (2001). *Sandboxing applications.* USENIX Annual Technical Conference, FREENIX Track. Retrieved from http://www.usenix.org/events/usenix01/freenix/prevelakis/

Radichel, T. (2014). *Case study: Critical controls that could have prevented Target breach.* SANS Institute Information Security Reading Room. Retrieved from https://www.sans.org/reading-room/whitepapers/casestudies/case-study-critical-controls-prevented-target-breach-35412

Schmidt, D., & White, J. (2017). *Why don't big companies keep their computer systems up-to-date? The Conversation.* Retrieved from

https://theconversation.com/why-dont-big-companies-keep-their-computer-systems-up-to-date-84250

"Vulnerability Details: CVE-2019-1222" *CVE Details; The Ultimate Security Vulnerability Datasource*, MITRECorporation, 14 August, 2019, Retrieved from https://www.cvedetails.com/cve/CVE-2019-1222/

Wang, H. J., et al. (2004). *Shield: Vulnerability-driven network filters for preventing known vulnerability exploits.* In Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (pp. 261-272). https://doi.org/10.1145/1015467.1015489

Wichers, D. (2013). *OWASP Top 10 2013*. OWASP Foundation. Retrieved from https://wiki.owasp.org/images/1/17/OWASP_Top-10_2013--AppSec_EU_2013_-_Dave_Wichers.pdf

Yadav, T., & Rao, A. M. (2015). *Technical aspects of cyber kill chain. In International Symposium on Security in Computing and Communication* (pp. 17-28). Springer, Cham. https://doi.org/10.1007/978-3-319-22915-7_40