Vulnerability Assessment, Remediation, and Automated Reporting: Case Studies of Higher Education Institutions

Christopher R. Harrell, Mark Patton, Hsinchun Chen
Department of Management Information Systems
University of Arizona
Tucson, AZ
{harrellcr, mpatton, hsinchun} @ email.arizona.edu

Sagar Samtani
Department of Management Information Systems
University of South Florida
Tampa, FL
ssamtani@usf.edu

***Abstract*— Scientific advances of higher education institutions make them attractive targets for malicious cyber-attacks. Modern scanners such as Nessus and Burp can pinpoint an organization's vulnerabilities for subsequent mitigation. However, the remediation reports generated from the tools often cause significant information overload while failing to provide actionable solutions. Consequently, higher education institutions lack the appropriate knowledge to improve their cybersecurity posture. In this study, we conduct a large-scale vulnerability assessment of 272 higher education institutions. From the results, we identified vulnerabilities that fail to provide comprehensive remediation strategies. Selected flaws are recreated and remediated in a virtual environment to develop enhanced, automated reporting mechanisms that provide succinct reports to enable the efficient vulnerability remediation. Our enhanced reports address 27.80% of vulnerabilities found in scanned higher education institutions.**

Keywords—higher education, vulnerability assessment, Shodan, Nessus, National Vulnerability Database

I. Introduction

Higher education institutions have made remarkable advances in physics, medicine, and applied sciences. However, these advances create a target for malicious cyber-attacks. Higher education institutions depend on the confidentiality and integrity of sensitive data stored at their facilities (e.g., intellectual property, financial data). Institutions find security implementation difficult due to culture, staffing, and resources conflicting with the demand for robust security [1]. These circumstances create an environment where managing the balance between transparency and security against malware, sensitive data exfiltration, and vulnerability exploitation an arduous process [1]. Data breaches in 2016 cost organizations on average $3.62 million [2]. The education industry has been labeled the slowest to patch, averaging 18% of vulnerabilities addressed during the 12 week patch cycle [3].

Technical vulnerabilities such as weakness in software or misconfigurations are key contributors toward risk in enterprise networks [4]. Systems protecting data regularly contain exploitable flaws, many of which can be detected with popular vulnerability scanners (e.g., Nessus) [5-6]. Unfortunately, reports generated from these tools are notorious for causing information overload by producing extensive data while failing to provide network context, risk prioritization, or actionable solutions [7]. Given the susceptibilities of higher education institutions, the societal impact of breaches, and the issues in current vulnerability remediation, our research aims to conduct a large-scale vulnerability assessment of 272 higher education institutions. From the results, we identify vulnerabilities that fail to provide comprehensive remediation strategies. Select flaws are recreated and remediated in a virtual environment. From this, we develop enhanced, automated reporting mechanisms that provide succinct, actionable fixes to enable the efficient vulnerability remediation.

The remainder of this paper is organized as follows. First, we review two key, relevant aspects of vulnerability assessments: scanning and remediation/reporting. We then identify gaps within extant practices and pose research questions for study. Subsequently, we present our research design. After presenting selected results, we highlight promising future directions and conclude this research.

II. Literature Review

Fundamentally, a vulnerability is "a flaw within a system, application, or service which allows an attacker to circumvent security controls and manipulate systems in ways the developer never intended" [8]. Vulnerability assessment is the systematic process of identifying security issues (i.e., vulnerabilities) in a computer, network, and communications infrastructure. Today, vendors such as Tenable, PortSwigger, and others provide automated vulnerability assessment tools to help organizations understand their cybersecurity posture. While capabilities may vary between vendors, all tools have two major functionalities: scanning and reporting. Both aspects are reviewed in detail to understand current methodologies, configurations, and reporting standards.

A. Vulnerability Scanning

Vulnerability scanners are software used to scan the architecture of a network, report detected vulnerabilities, and provide instructions on how to remediate them [4]. The core architecture of a vulnerability scanner is depicted in Figure 1.

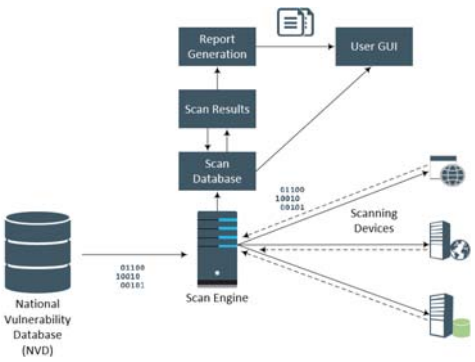

Fig. 1. Vulnerability Scanner System Diagram

Scanners often use definitions provided by the National Vulnerability Database (NVD) [9]. NVD contains

information on Common Vulnerabilities and Exposures (CVE), which is a list of standardized names for known vulnerabilities [6,10]. Definitions are loaded into scanning engines to scan for vulnerabilities on a device. The NVD as of this writing, contains 96,330 signatures. Signatures generated from the NVD have a risk level categorization known as the Common Vulnerability Scoring System (CVSS). Factors such as attack vector, complexity, privileges required, user interaction, and the impact of confidentiality, integrity, and availability are incorporated into CVSS scores [9]. Sample vulnerabilities in each are provided in Table 1.

TABLE I. RISK LEVELS AND EXAMPLES FOR VULNERABILITIES

Risk Level	CVSS Ranges	Example Vulnerabilities
Critical	10.0	Remote Code Execution, Buffer Overflows, Default Credentials, Unsupported Operating System Versions
High	7.0 – 9.9	Malformed Packet Injection, Redirect Denial of Service, Privilege Escalation, Password Hash Disclosure
Medium	4.0 – 6.9	Remote Information Disclosure, Cryptographic Protocol, Command Injection, Web Directory Traversal & File Access
Low	0.1 – 3.9	Unencrypted Communications, Internal Information Disclosure, Browsable Web Directory
Informational	0.0	Software Version Disclosure, Protocol Detection, Operating System Identification, Device Type

All scan results are held in a database. This database helps provide summaries to describe identified susceptibilities, present a risk rating, and give guidance on how to mitigate and remediate the discovered weaknesses [10].

B. *Remediation & Reporting*

Vulnerability remediation aims to remove vulnerabilities through a compensating security control. Security controls involving remediation can provide technical (e.g., applying a patch, security architecture changes, system or application hardening) strategies to address risk within an organization [10]. The length of time a device remains unprotected on a network increases the potential for exploitation. Removal of the susceptible application, protocol, or device is ideal granted removal does not impede functionality. Knowledge of required services, protocols, processes, and applications in an organization enhances a security professionals' ability to identify dangerous assets and apply appropriate defense. Removing a system deficiency often affects several others, creating a domino effect [4]. For example, upgrading SMBv1 to SMBv2/SMBv3 instead of patching, would avoid any exploitative actions against SMBv1.

Reports provide a summary view of systems and corresponding findings. Information such as IP address, risk rating, vulnerability description, and solutions are maintained within reports. Professionals primarily rely on the mitigation and remediation guidelines provided within reports to address vulnerable devices and systematically eliminate network vulnerabilities [4]. However, several studies have identified two key imperfections commonly found in these reports [4,7]. First, reports often contain tremendous amounts of information while failing to produce actionable fixes [7]. Second, vulnerabilities often require multiple resources outside of provided guidelines, many of which provide no resolution to the identified vulnerability. Figure 2 illustrates a report demonstrating both issues.

Fig. 2. Remediation Report: (a) two hyperlinks referencing identical content, (b) selecting hyperlinks leads to retired content, (c) download of retired content contains 28,365 page .pdf with 677 search results as 'smb'

The SMB Signing Disabled remediation solution depicted in Figure 2 is a 28,365 page long PDF. Attempting a find command (CTRL+f) with the SMB keyword returned 677 results, none of which offered concise or actionable content. The report solution also assumes the user knows Group Policy. If not, users are instructed to inspect the reference hyperlinks for additional guidance. However, two hyperlinks were HTTP 404 not found, and two forwarded to the same page referencing, "Windows Server 2003/2003 R2 Retired Content," containing an in-depth description and suggested solution.

III. RESEARCH GAPS AND QUESTIONS

Prevailing vulnerability assessment approaches have several limitations. Vulnerability assessment literature has focused on scanning against different devices, comparing scanners, risk management, and vulnerabilities by industry. Despite the maturity in scanning capabilities, we are unaware of any studies systematically enhancing remediation and reporting features; past work has only underscored current reporting deficiencies. Consequently current vulnerability assessment reporting mechanisms do not provide technical and actionable insight into how vulnerabilities are remediated beyond the scanner reporting

guidelines. These gaps motivate the following research questions:

- What vulnerabilities afflict higher education institutions?
- How can vulnerability scanning generated reports be enhanced to decrease the time to remediation?
- What additional features can be created to enhance current reporting mechanisms?

IV. Research Testbed and Design

To answer the posed research questions, we developed a system design (Figure 3) with four major components: (1) Data Collection, (2) Vulnerability Assessment, (3) Vulnerability Replication, and (4) Report Generation.

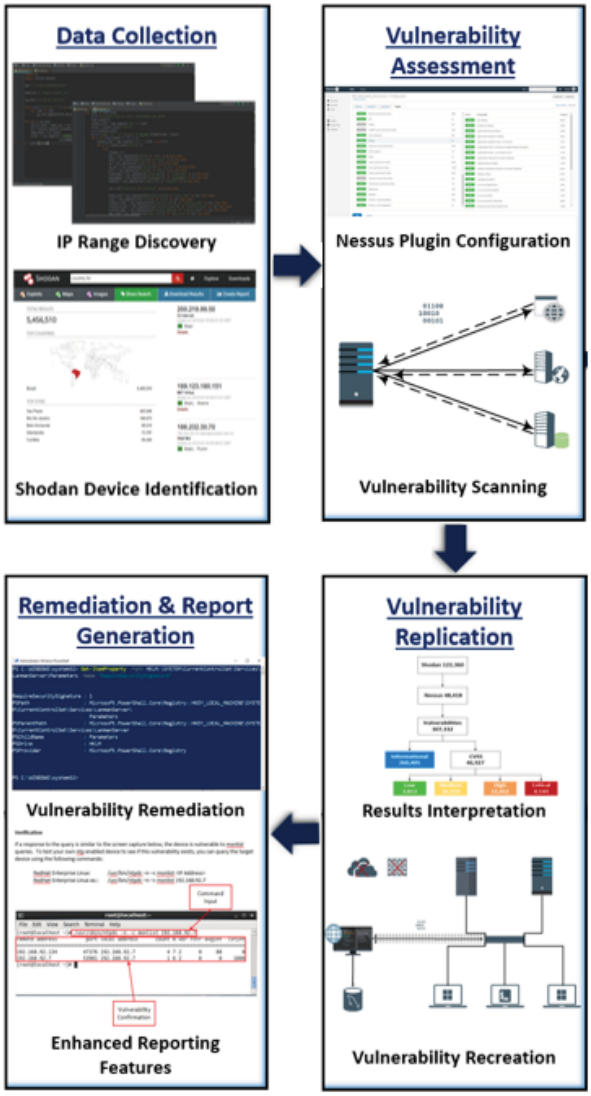

Fig. 3. Vulnerability Assessment, Remediation, & Reporting System Design

The data collection component of the design identifies the openly accessible devices of higher education institutions. Our data collection focuses on 272 higher education institutions, further information was gathered using open source intelligence tools. Using ipinfo API, we pinpointed 168 unique IP ranges from the 272 institutions comprising of 2,900,912 million potential devices. Each IP range was passed through Shodan, a search engine which scans and indexes billions of publicly facing (i.e., not behind a NAT firewall) Internet-connected devices.

Following device identification, we used Nessus, a state-of-the-art vulnerability assessment tool, to discover the vulnerabilities of the devices returned by Shodan. While other tools exist, experts have cited Nessus as "one of the most comprehensive and widely deployed" [11]. Nessus has over 93,500 plugins to assess a variety of technologies on large enterprise networks. The scanning engine for Nessus uses definitions from the NVD. As a result, Nessus can identify web application flaws, discover outdated software, test default credentials, and find database issues. Each is categorized into a CVSS risk threshold. For this study, we configured the Nessus scan to be unauthenticated, disabled irrelevant plugins, avoided brute-forcing credentials, and turned off port scanning to avoid potentially crashing systems. Executing scans in this fashion is consistent with recent literature assessing vulnerabilities in consumer Internet of Things (IoT), medical, and Supervisory Control and Data Acquisition (SCADA) devices [12 - 16]

After completing the assessment, we carefully examine the detected vulnerabilities to identify which could be recreated in a virtual environment. Three criteria were used to identify vulnerabilities to recreate: 1) frequency of occurrence, 2) impact if exploited, and 3) report solutions that were deemed not comprehensive enough to remediate in a timely manner. Following vulnerability recreation, we remediated, documented, and implemented results into a novel, enhanced reporting template. Additional details of selected vulnerabilities are provided in the following section.

V. Results and Discussion

A. Device Discovery and Vulnerability Assessment Results

Shodan's API resolved the 168 IP ranges from the 272 institutions to 122,360/2,900,912 (4.21%) unique devices having 256,558 flaws. We parsed the data into IP subnets and noticed that several of the subnets have increased exposure. Table 2 shows the /19, /22, /23, and /24 ranges having significantly higher vulnerable devices then their counterparts, 13.24%, 14.23%, 14.06%, and 12.68% respectively.

TABLE II. Sample Higher Education Institutions Exposure on Shodan by IP Range

IP Range	# of Possible Devices	# of Institutions in IP Range	# of Potential Devices	# of Devices on Shodan	Exposure on Shodan
/15	131,070	1	131,070	3,804	2.9022%
/16	65,534	26	1,703,884	60,926	3.5757%
/17	32,766	12	393,192	13,613	3.4621%
/18	16,382	26	425,932	22,478	5.2773%
/19	8,190	12	98,280	13,022	**13.2498%**
/20	4,094	26	106,444	4,593	4.3149%
/21	2,046	10	20,460	998	4.8778%
/22	1,022	6	6,132	873	**14.2367%**
/23	510	12	6,120	861	**14.0686%**
/24	254	37	9,398	1,192	**12.6835%**

Nessus scans revealed 48,418/122,360 (39.57%) unique vulnerable devices. Devices had various open ports, the most prominent being web specific ports (e.g., 8008, 80, 8080, 443, 8443, etc.), and ports for general purpose services such as SSH, DNS, SNMP, Telnet, FTP, etc. Devices with informational vulnerabilities amounted to 31,567/48,418 (65.19%) and devices containing a CVSS were 16,851/48,418 (34.80%). On the 48,418 devices, we discovered 307,332 vulnerabilities. 46,927 contained a CVSS and risk level. Specifically, 4,143 were 'Critical,' 12,452 were 'High,' 26,519 were 'Medium,' 3,813 were 'Low,' and 260,405 were 'Informational.' 334 vulnerabilities did not contain a CVSS, but had a risk level. A graphical summary of the Shodan and Nessus data showing the devices and corresponding vulnerabilities can be seen in Figure 4.

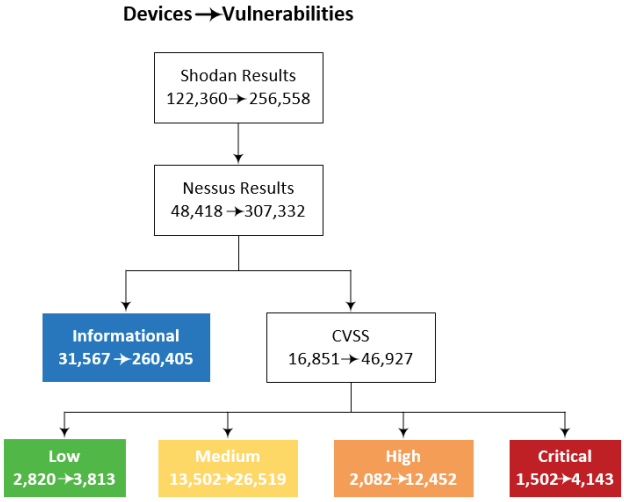

Fig. 4. Vulnerability Scanning Results

B. *Vulnerability Selection for Recreation*

Using Nessus' built-in device identification, we identified the most common operating systems: RedHat Enterprise Linux, Ubuntu, CentOS, Cisco IOS, Windows Server 2000, 2003, 2008 R2, 2012 R2, and Windows XP, Vista, 7, and 10. We selected RedHat Enterprise Linux, Windows 7, 10, Windows Server 2008 R2, and 2012 R2 to replicate vulnerabilities due to their occurrence and popularity in the dataset. Table 3 lists the selected vulnerabilities ordered by risk level. Remediating all vulnerabilities in Table 3 would equate to 4,685/16,851 (27.80%) removal of devices in our dataset. 'Informational' weaknesses do not contain a CVSS and are commonly ignored due the absence of risk level.

The 'Critical' vulnerability associated with the protocol Windows' Server Message Block (SMB) was found to be a catalyst for a myriad of 'Informational' flaws. We found that SMB was linked to information disclosures, these types of disclosures provide attackers valuable material for further reconnaissance. In fact, SMB effects 15,114/48,418 (31.21%) of hosts resulting in operating system identification, uncredentialed checks, registry enumeration, unauthenticated logins, host information disclosure, and host enumeration. SMBv1 has been linked to several critical exploits, one of which is EternalBlue which affected all Windows machines which running the outdated protocol version [17]. A result of this exploit was the WannaCry ransomware attack in May 2017, which affected over 300,000 computers worldwide [17]. SNMP Agent Default Community Name (public) is the largest flaw in the 'High' category. SNMPv1 and SNMPv2 community string is commonly defaulted to 'public'.

TABLE III. NESSUS VULNERABILITY SELECTION FROM SHODAN RESULTS

Risk Level	# of Devices	Vulnerability Names(s)	Vulnerability
Critical	504	Microsoft Windows SMBv1 Multiple Vulnerabilities	Multiple Remote Code Execution, Multiple DoS, Multiple Information Disclosure
High	1,540	SNMP Agent Default Community Name (public)	Information Disclosure
Medium	4,457	Network Time protocol (NTP) Mode 6 Scanner	DoS Amplification, Information Disclosure
	423	SMB Signing Disabled	Information Disclosure
	8	Network Time Protocol Daemon (ntpd) monlist Command Enabled DoS	DDoS, Information Disclosure
Low	675	SSH Server CBC Mode Ciphers Enabled	Plaintext Communications
	652	SSH Weak MAC Algorithms Enabled	Weak Algorithms

While Telnet was the most frequent 'Medium' flaw, the remediation solution is well documented (disable Telnet and use SSH). Therefore, we choose the NTP weakness due to a lack of resolution in reporting and unauthenticated remote attack exploitability. The monlist and mode 6 query flaws can be leveraged in amplification and distributed denial of service (DDoS). Although the monlist vulnerability only affected eight devices, if the NTP version allowed for mode 6 scanning, monlist command execution was possible. SMB Signing Disabled was chosen due to the number of hosts affected and the information overload provided from current reports. SSH CBC Mode Ciphers and SSH Weak MAC Algorithms were chosen due to the highest number of hosts affected in the 'Low' category, and the absence of a solution within the reporting feature.

C. *Virtual Environment*

Our virtual environment enables us to recreate vulnerabilities discovered in our scans, test remediation techniques, and verify the removal by rescanning the systems. Operating systems for vulnerabilities can change the way remediation is achieved. We took this into consideration while creating vulnerabilities in the virtual environment and documented solutions on operating systems we thoroughly tested. For instance, NTP can be installed on both RedHat Enterprise Linux and Windows, we chose to replicate it on Windows 2008 R2 and RedHat version 6.6. However, the same NTP flaws can also affect other distributions of Windows and Linux / UNIX. To attain the most precise results, we disabled firewalls on the virtual machines (VMs) and disconnected the environment from the

internet to protect against asset exposure and tampering of results that could occur if exploited. Figure 5 depicts our virtual environment configuration.

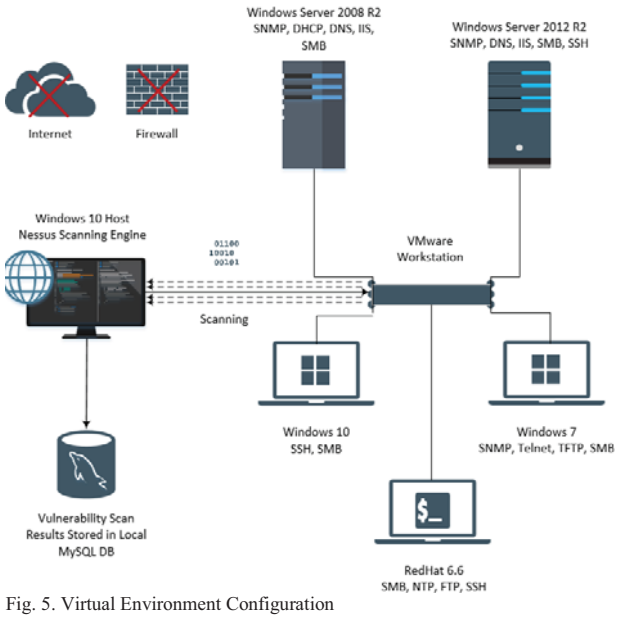

Fig. 5. Virtual Environment Configuration

Our virtual environment consists of five VMs and one host. VMs were hosted on VMware Workstation on the Windows 10 Professional host. The VMs are Windows Server 2008 R2, Windows Server 2012 R2, Windows 10 Professional, Windows 7 Professional, and RedHat Enterprise Linux version 6.6. Several services (e.g., SMB, SSH, SNMP, DHCP, DNS, NTP, FTP, Telnet, TFTP, IIS) were activated on these machines to support the creation of vulnerabilities. The host served outside of the domain connected machines to simulate the original scans.

D. Remediation Process

Thorough testing was required to achieve remediation of the vulnerabilities. In order to ascertain if the solution existed within current reporting mechanisms, we manually reviewed each report and associated hyperlinks. We then searched open source information (e.g., NVD, CVE, vendor websites, technical manuals, etc.) relating to the vulnerability attributes such as the service, process, application, system calls, and configurations. Following this, we tested potential solutions on the vulnerable machines. The most direct solution was applied to the machine and scanned to verify the vulnerability was removed. This solution was then documented for our enhanced report.

E. Enchanced Report Generation

During manual review of vulnerability reports, we detected several anomalies and shortcomings. A regular occurrence was minimal information within the solution section such as, “Contact vendor for more information”, or an overwhelming amount of information such as the SMB Signing Disabled flaw. We identified other common flaws in reporting mechanisms as failing to provide alternative solutions, verification of remediation, and non-specific information in references. We address these reporting mechanisms in four ways: 1) improving recommended and alternative solutions, 2) developing verification section, 3) improving references, and 4) implementing an automated reporting feature. Our enhanced reports range from 2-8 pages in length, including step-by-step guides with screenshots on how to remediate and verify. We developed scripts to automatically alert users of the enhanced reports. Figure 6 illustrates an example of an enhanced ntp monlist vulnerability report that can be automatically provided to end users. The example report provides a solution for the vulnerability, an alternative resolution if the primary solution is not feasible, and verification that the vulnerability exists by executing a bash script.

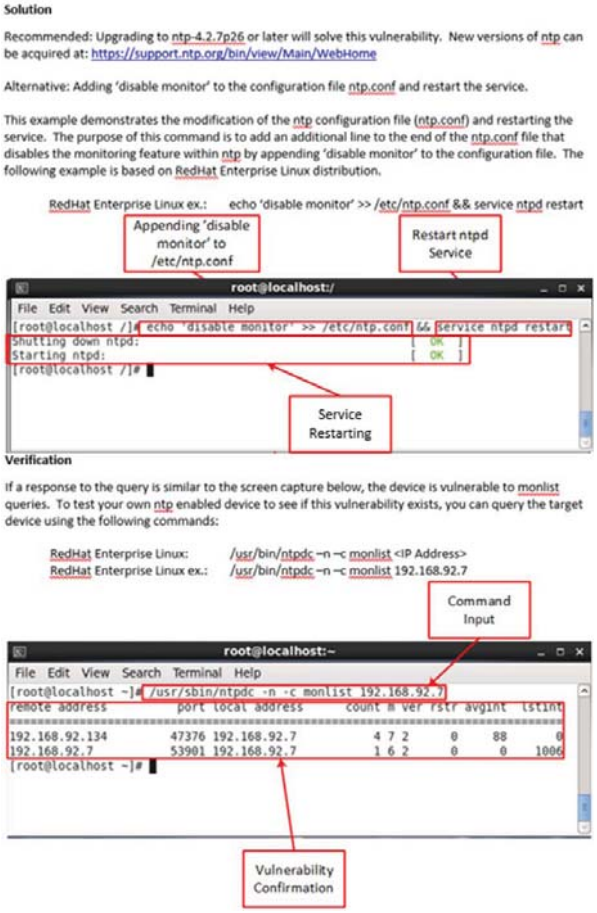

Fig. 6. Remediation Report Solution Recommended, Alternative, and Verification monlist Example

Each report is enhanced by adding and expanding on current subfields. Reports often recommend solutions such as “apply a patch” but rarely provide alternatives; we provide an alternative solutions if the recommended is not feasible to implement. To avoid any false positive or false negative issues, the verification section offers the user an option to manually test and automate the solution. This has the added benefit for full scale automated verification of vulnerable devices through technical means (e.g., PowerShell, bash, Group Policy). Should the end users need additional information, we provide links in the reference section with more direct solutions related to the vulnerability as opposed to general information. Taken together, these advanced reporting capabilities enable

professionals' timely execution of remediation efforts, a critically needed capability in prevailing vulnerability assessment practices.

VI. Conclusion and Future Directions

Researchers and educators are making remarkable progress in pushing the boundaries of knowledge within higher education. However, these advances often make higher education institutions the target of malicious cyber-attacks (often cited as the most vulnerable industry) [3]. While many vulnerability assessment tools can aid institutions identify the significant information overload and lack of actionable fixes provided by reports generated by these tools prevents the efficient and effective remediation of detected vulnerabilities.

In this study, we leveraged Nessus, a state-of-the-art vulnerability assessment tool, to identify numerous vulnerabilities in 272 higher education institutions. In addition to identifying vulnerabilities, we enhanced the current, verbose reporting features provided by existing tools to deliver clear, concise, and comprehensive vulnerability assessment reporting features. The newly developed scripts can help systems administrators in our scanned higher education institutions address 27.80% of their vulnerabilities, many of which can help prevent dangerous cyber-attacks (e.g., DDoS).

There are several promising directions for future research. First, work can be done in collaboration with selected and interested higher education institutions to assist in remediation efforts. Second, the enhanced reports can be created for additional vulnerabilities and posted on a vulnerability-wiki to assist in unifying remediation efforts. Finally, a usability study could be conducted to further identify faults in current reporting mechanisms to quantitatively measure the time to remediate a vulnerability compared to the current reporting mechanisms. Each direction can enhance an organization's ability to remediate detected vulnerabilities and help ultimately ensure a safer, more secure society.

Acknowledgment

This material is based upon work supported in part by the National Science Foundation (NSF) DUE-1303362 (Scholarship-for-Service) and SES-1314631 (Secure and Trustworthy Cyberspace).

References

[1] R. Marchany, "Higher Education: Open and Secure?," SANS Institute InfoSec Reading Room, 2014, p. 28.

[2] Ponemon Institute LLC, "2017 Cost of Data Breach Study," 2017. [Online]. Available: https://www.ponemon.org/library/. [Accessed: 17-Sept-2017].

[3] Verizon, "2017 Data Breach Investigations Report," 2016. [Online]. Available:http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/. [Accessed 22-Aug-2017].

[4] H. Holm, "Performance of Automated Network Vulnerability Scanning at Remediating Security Issues," Computer and Security, vol. 31, no. 2, 2012, pp. 164–175.

[5] B. Filkins, "Security by Design: The Role of Vulnerability Scanning in Web App Security," SANS Institute InfoSec Reading Room. p. 18, 2017.

[6] H. Homaei and H. R. Shahriari, "Seven Years of Software Vulnerabilities: The Ebb and Flow," IEEE Security and Privacy, vol. 15, no. 1, 2017, pp. 58–65.

[7] C. Vazquez, "Auditing Using Vulnerability Tools to Identify Today's Threats to Business Performance," SANS Institute InfoSec Reading Room, 2014, p. 24.

[8] G. Weidman, "Penetration Testing: A Hands-On Introduction to Hacking," 1st ed No Starch Press, 2014.

[9] Gorbenko, A. Romanovsky, O. Tarasyuk, and O. Biloborodov, "Experience Report: Study of Vulnerabilities of Enterprise Operating Systems," IEEE 28th Int'l Symp. Software Reliability Engineering, 2017, pp. 205–215.

[10] NIST, "Security and Privacy Controls for Information Systems and Organizations," NIST Special Publication 800-53, 2017, p. 494.

[11] P. Stephenson, "Tenable Network Security Nessus," 2015. [Online]. Available: https://www.scmagazine.com/tenable-network-security-nessus/review/6977/. [Accessed: 11-Apr-2017].

[12] S. Samtani, S. Yu, H. Zhu, M. Patton, & H. Chen, "Identifying SCADA Vulnerabilities Using Passive and Active Vulnerability Assessment Techniques," in 2016 IEEE International Conference on Intelligence and Security Informatics (ISI), 2016, pp. 25–30.

[13] R. Williams, E. McMahon, S. Samtani, M. Patton, & H. Chen, "Identifying vulnerabilities in consumer IoT devices: A scalable approach," IEEE International Conference on Intelligence and Security Informatics (ISI), 2017, pp. 179-181.

[14] E. McMahon, R. Williams, M. El, S. Samtani, M. Patton, & H. Chen, "Assessing medical device vulnerabilities on the Internet of Things," IEEE International Conference on Intelligence and Security Informatics (ISI), 2017, pp. 176-178.

[15] E. McMahon, R. Williams, M. El, S. Samtani, M. Patton, & H. Chen, "Benchmarking vulnerability scanners: An experiment on SCADA devices and scientific instruments," IEEE International Conference on Intelligence and Security Informatics (ISI), 2017, pp. 83-88.

[16] S. Samtani, S. Yu, H. Zhu, M. Patton, J. Matherly, & H. Chen. Identifying SCADA Systems and Their Vulnerabilities on the Internet of Things: A Text-Mining Approach. IEEE Intelligent Systems, vol. 33 no. 2, pp. 63–73, Mar. 2018.

[17] Q. Chen and R. A. Bridges, "Automated Behavioral Analysis of Malware A Case Study of WannaCry Ransomware," IEEE 16th Int'l Conf. Machine Learning and Applications, 2017, pp. 454–460.