HSJ

Vol. 2, Issue. 2

Health Sciences Journal
EISSN: 2959-2259; PISSN: 2959-2240

Received: 04 JUNE 2024 | Accepted: 06 JUNE 2024 | Published: 07 JUNE 2024

MDRI

Research Article

VULNERABILITY ASSESSMENT OF OPERATING SYSTEMS IN HEALTHCARE: EXPLOITATION TECHNIQUES AND SECURITY IMPLICATIONS

Muhammad Ahsan Qureshi[1] | Shakeel Ahmed[2]| Asim Mehmood[3*]| Reema Shaheen[4]| Muhammad Shahid Dildar[5]

[1]Department of Computer Science, Air University, Islamabad, Pakistan
Email: ahsan.qureshi@aack.au.edu.pk
[2]Department of eLearning Center (ELC), Jazan University, Saudi Arabia
Email: shakeel@jazanu.edu.sa
[3]Department of Health Informatics, College of Public Health and Tropical Medicine, Jazan University, Saudi Arabia.
Email: assimrza@gmail.com
[4]Department of e-learning Center (ELC), Jazan University, Saudi Arabia.
Email: rima@jazanu.edu.sa
[5]Department of Computer Science, King Khalid University, Saudi Arabia
Email: mdildar@kku.edu.sa

Correspondence
Asim Mahmood
Email: assimrza@gmail.com

Citation
Qureshi MA, Ahmed S, Mahmood A, Shaheen R, Dildar MS, Vulnerability assessment of operating systems in healthcare: Exploitation techniques and security implications. Health Sciences Journal, 2024; 2(2): 104-111

ABSTRACT:

Aim: This research highlights the importance of vulnerability assessment and techniques for exploiting the Windows operating system (OS) in health care. Utilizing CVE data and other vulnerability reports plays a crucial role in evaluating healthcare's operating system security posture. Security tools such as Metasploit, msfvenom, Nessus, and Nmap were required to conduct vulnerability assessments and intrusion experiments in a simulated environment. **Material & Methods:** The research followed a typical ethical hacking procedure, including reconnaissance, network scanning, vulnerability assessment, exploit creation, and gaining access to the latest version of the Windows OS. Despite installing the latest version of Windows, complete protection against attacks is not guaranteed.
Future Research Directions: Further research is necessary to assess the system's vulnerabilities and recommend improved solutions thoroughly.

KEYWORDS:

Operating System, Healthcare, Security Implications

1 | INTRODUCTION

In the contemporary interconnected digital environment, cybersecurity has surfaced as a significant apprehension, where operating systems are primary targets for malevolent entities aiming to exploit vulnerabilities for malicious intents[1]. The Windows OS holds a prominent status among these platforms, being extensively utilized, thus attracting attention from security experts and cybercriminals alike[2]. A comprehensive strategy is essential for grasping and alleviating the potential hazards linked with Windows 10 vulnerabilities, involving thorough vulnerability assessment and an in-depth investigation of exploitation methodologies. This study initiates a crucial evaluation of Windows 10 vulnerability assessment, elucidating the complex relationship between security vulnerabilities and exploitation techniques within the domain of this pervasive OS[3].

Health Sciences Journal
EISSN: 2959-2259; PISSN: 2959-2240

By referencing data from Common Vulnerabilities and Exposures (CVE) reports and additional vulnerability databases, the aim is to clarify the multifaceted aspect of Windows 10 security and its repercussions for system administrators, security analysts, and end-users[4]. The core of our analysis revolves around the application of state-of-the-art security utilities and approaches, encompassing tools such as Metasploit, msfvenom, Nessus, and Nmap. Through an ethical hacking perspective, an exploration is conducted on reconnaissance, network scanning, vulnerability assessment, and exploit development, culminating in an endeavor to breach the most recent version of Windows 10 OS[5, 6]. This comprehensive methodology uncovers the intrinsic vulnerabilities within Windows 10 and accentuates the shortcomings of existing security protocols in combating sophisticated cyber perils. Despite the relentless enhancements to reinforce Windows 10 against potential breaches, our investigation discloses that achieving robust defense mechanisms remains an enduring obstacle. Even with the deployment of the latest OS iteration, the prevalence of vulnerabilities accentuates the necessity for continuous vigilance and preemptive security strategies[7]. Consequently, this study advocates for further research initiatives dedicated to elucidating and rectifying the fundamental flaws within the Windows 10 framework, with the primary objective of enhancing its security stance and guaranteeing the endurance of digital infrastructure in an ever-changing landscape of threats[8].

2 | RELATED RESEARCH

A study emphasizes the importance of assessing operating system vulnerabilities for cybersecurity by comparing risk levels among various OSs using the CVSS for the healthcare environment. The authors introduce a quantitative method to evaluate OS performance based on vulnerabilities, emphasizing reliability and security in OS design. The methodology calculates risk indexes to identify significant differences in risk levels among OSs. Test results demonstrate varying risk levels among OSs based on assumptions and limitations. Future research may focus on forecasting vulnerabilities and enhancing risk assessment models for better cybersecurity measures[9]. Research explains the Vulnerability Intensity Function (VIF) and Vulnerability Index Indicator (VII) for computer OS in healthcare systems, using real data for Microsoft Windows and Apple MacOS. Likelihood function is used to estimate VII and the vulnerability rate for MacOS. Non-linear statistical Models assess the risk factor of vulnerabilities and predict attacks. The stochastic process characterizes vulnerabilities' probabilistic behavior in OS, introducing VIF and VII as key concepts in cybersecurity. Non-Linear Statistical Models evaluate risk factors and predict vulnerability using VIF and VII. VIF and VII are crucial in understanding vulnerabilities in computer OS, providing insights into vulnerability rates and security risks[10,11].

The author compares vulnerabilities in modern Windows operating systems with different vulnerability scanners, emphasizing the importance of choosing Windows OS based on security needs and recommending timely updates and security investments. Various tools such as Nessus were used to scan vulnerabilities across different Windows OS versions. The assessment was done using three widely known vulnerability scanners to provide practical results of vulnerabilities in popular Windows operating systems. Tools like Nessus were utilized for vulnerability scanning during the assessment to help individuals make informed decisions when selecting Windows OS. The results of the assessment aim to guide users in choosing Windows OS based on security needs, stressing the importance of security updates and investments[12]. Study examines security issues in modern computer systems due to operating system vulnerabilities, stressing the necessity of complex defense mechanisms against security threats. It points out a concerning trend where security update rates are not based on vulnerability severity, revealing flaws in update policies. Statistics mentioned were gathered by querying a database that combines data from CVE and NVD repositories. The study's research methodology involves querying a MySQL database that collects data from CVE and NVD repositories. Vulnerability databases such as NVD identify various software products and analyze common vulnerabilities in different operating systems[13].

The study assessed Windows 10 vulnerabilities and its ability to withstand cyber-attacks. The methodology included testing procedures with information gathering, scanning, vulnerability selection, launch attacks, and gaining system access. Eight penetration tests were conducted, two successful against different Windows 10 versions. CVE data and other reports measured system performance. Identifying vulnerabilities in Windows 10 through CVE data and reports can aid in understanding system performance and weaknesses. Testing system resilience with tools like Metasploit and Nmap can show security measure effectiveness. Different Windows 10 versions' susceptibility to specific attacks underscores need for continuous monitoring and updates to improve system protection. Research highlights importance of further studies to evaluate vulnerabilities and recommend enhanced solutions for Windows 10 security[14,15]. The study uses modern internet tools to analyze weaknesses in Windows OS. Two scanning methods

with three scanners evaluate vulnerabilities in versions like Windows XP and 10. Results compare scanners' identification abilities and service packs' effectiveness in fixing flaws. Various scanners are compared to assess their performance in revealing Windows vulnerabilities. Data includes vulnerability scanning results and scanner comparisons for modern Windows systems. The study analyzes Windows XP and 10 vulnerabilities using internet tools and multiple scanners. Two scanning methods with three scanners evaluate system vulnerabilities comprehensively[16].

The research shows how neural networks and vulnerability data predict and understand Windows 10 weaknesses, highlighting factors affecting severity [17,18]. The study examines the causes and solutions for the lack of free space in Windows 10 system partitions and discusses registry structure[19]. It compares software for system partition optimization, evaluating their efficiency. Methods analyze system partition organization during Microsoft OS installation. It also explores main system file objects and their roles in system partition space usage. Folders taking up most space on system partition are identified for potential cleaning. The methodology compares software efficiency for system partition optimization based on cleaning functions. Limitations of focusing on Windows 10 system partition cleaning are discussed. The importance of using proper tools to optimize system partition space in Windows 10 is highlighted for enhanced performance [20,21, 22.]

3 | EXPERIMENTAL SETUP AND ANALYSIS

3.1 | PROPOSED METHOD

Hackers systematically follow a process to identify vulnerabilities in operating systems, software, or web applications that they could exploit to harm individuals or organizations. The exploitation process utilized in this research is illustrated in Figure 1 and explained in the experimental phases as follows.

3.2 | PLANNING AND RECONNAISSANCE

In this phase, the attacker carefully gathers all the necessary information about the target. They do this by using various OSINT techniques to find publicly available information, such as from social media or other news sources. This process helps attackers gather actionable intelligence and identify potential attack vectors to launch an attack.

3.3 | SCANNING AND ENUMERATION

Attackers identify critical exploitable vulnerabilities and misconfigurations within operating systems, software, or web applications as a preliminary step to launching an effective cyber-attack. During this phase, they gain more technical insights by scanning the target's digital infrastructure by using various tools. Nmap and Nessus Expert were used in this research to conduct networking scanning and vulnerability assessment. Nessus Expert classified vulnerabilities as High, Medium, Low, or Info using CVSS v3.0.

3.4 | GAINING ACCESS (EXPLOITATION)

An attacker then utilizes all means to get unauthorized access to the target's networks, applications, or systems. An attacker attempts to get into the system and exploit the system by downloading malicious software or applications, stealing sensitive information, getting unauthorized access, etc. In this research, msfvenom is used to create the payload and Metasploit is used to gain access. Computer-based social engineering techniques are utilized to deliver the maliciously crafted payload to the target system.

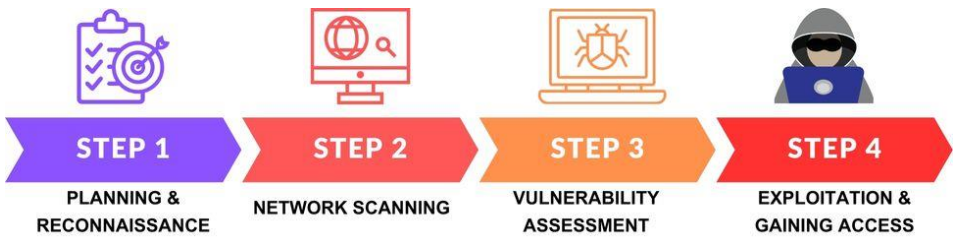

Figure 1 Experimental Phases

Health Sciences Journal
EISSN: 2959-2259; PISSN: 2959-2240

3.5 | LAB SETUP AND ANALYSIS

A real-life network simulated lab environment has been created to conduct this research, incorporating diverse deployments of operating systems such as Kali Linux, Windows 10, Windows 7, Ubuntu, and Metasploitable2 (see Figure 2). Having these different systems helped in this research to test for different vulnerabilities. This setup helped to grasp a clear understanding of how network security functions and provided insight into the methods attackers might employ to infiltrate systems. The specifications of the research workstation used for the experimental part are provided in Table 1.

Table 1 Workstation Details

Device	Specifications
Hardware	HP Omen 15
Operating System	Windows 10 Home, version 22H2
CPU	Intel(R) Core (TM) i7-8750H CPU @ 2.20GHz 2.21 GHz
RAM	32 GB
System Type	64-bit operating system, x64-based processor
Graphics Card	NVIDIA GeForce GTX 1060

VMware Workstation Pro 16 was used to establish a virtual environment for analyzing Windows vulnerabilities, resulting in five virtual machines (see Table 2). All the virtual machines' operating systems were officially updated before experimentation, and no third-party software or patches were applied during the OS installation. Kali Linux 2024.1 was the attacker on VM 1, while the VM 4 configuration involved Nessus Expert.

Table 2 Lab setup Details

Sr. No.	VM Name	OS	Role	Version	IP Address
	Kali Linux	Linux	Attacker Machine	2024.1	192.168.230.139
	Windows 10	Windows	Target Machine	Home (22H2)	192.168.230.141
	Windows 07	Windows	Host Machine	Pro	192.168.230.136
	Ubuntu	Linux	Host Machine	22.4	192.168.230.143
	Metasploitable2	Linux	Web Server	Ubuntu 8	192.168.230.137

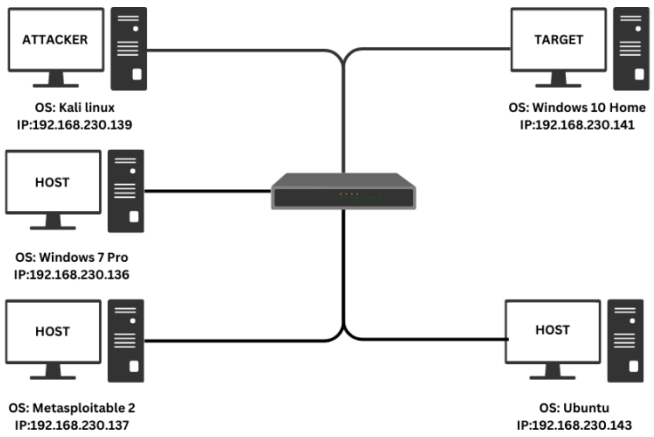

Figure 2 Lab Setup for Testing Environment

3.6 | NETWORK SCANNING

All virtual machines were connected to the same local area network (LAN). Address Resolution Protocol (ARP) is a protocol used to dynamically map IP addresses to permanent physical machine addresses within a LAN. The "arp -v -n" command is utilized to reveal all computers within the same LAN (Figure 3). To identify the target machine, a network aggressive scan was conducted using Nmap with the parameters “nmap -A 192.168.230.0/24” (Figure 4).

Health Sciences Journal
EISSN: 2959-2259; PISSN: 2959-2240

```
┌──(kali㉿kali)-[~]
└─$ arp -v -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.230.143          ether   00:0c:29:cd:2e:5c   C                     eth0
192.168.230.254          ether   00:50:56:f4:98:bd   C                     eth0
192.168.230.1            ether   00:50:56:c0:00:08   C                     eth0
192.168.230.136          ether   00:0c:29:71:bc:e1   C                     eth0
192.168.230.2            ether   00:50:56:fa:4f:1e   C                     eth0
192.168.230.141          ether   00:0c:29:57:90:45   C                     eth0
192.168.230.137          ether   00:0c:29:69:8c:24   C                     eth0
Entries: 7    Skipped: 0    Found: 7
```

Figure 3 List of Host Machines on LAN

```
Nmap scan report for 192.168.230.141 (192.168.230.141)
Host is up (0.00051s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
MAC Address: 00:0C:29:57:90:45 (VMware)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1709 - 1909
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2024-05-01T19:47:14
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_nbstat: NetBIOS name: TARGETPC-WIN10, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:57:90:45

TRACEROUTE
HOP RTT     ADDRESS
1   0.51 ms 192.168.230.141 (192.168.230.141)
```

Figure 4 Scanning Target Machine

3.7 | VULNERABILITY ASSESSMENT

The target Windows system underwent scanning with both Nmap and Nessus Expert scanners. The Nmap scan, utilizing the vuln scan script, revealed no associations with CVEs from well-known databases. The parameters used for the Nmap scan were as follows: nmap --script vuln 192.168.230.141 (see Figure 5).

```
┌──(root㉿kali)-[~]
└─# sudo nmap --script vuln  192.168.230.141
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-02 00:58 PKT
Nmap scan report for 192.168.230.141 (192.168.230.141)
Host is up (0.00026s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:57:90:45 (VMware)

Host script results:
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_smb-vuln-ms10-054: false
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR

Nmap done: 1 IP address (1 host up) scanned in 25.96 seconds
```

Figure 5 Nmap Vulnerability Scanning

3.8 | NESSUS EXPERT SCAN

The target system was scanned with Nessus Expert by utilizing the latest updated from CVSS v3.0 (see Figure 6).

0	0	2	0	38
CRITICAL	HIGH	MEDIUM	LOW	INFO

Figure 6 Nessus Scans Results

3.9 | PAYLOAD GENERATION

During the vulnerability assessment phase, we found no critical exploitable vulnerabilities in the target machine. To proceed with exploitation, we created a custom payload, a Windows executable file (virus.exe), using msfvenom (see Figure 7). This payload includes a reverse TCP Meterpreter connection to the attacker's IP on port 4444, enabling remote access to the target system upon execution.

Health Sciences Journal
EISSN: 2959-2259; PISSN: 2959-2240

```
┌──(root㉿kali)-[~]
└─# sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.230.139 LPORT=4444 -f exe -o virus.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: virus.exe
```

Figure 7 Custom Payload Generation Using msfvenom

3.10 | EXPLOITATION

Attackers gain unauthorized access to the systems by exploiting vulnerabilities. After crafting a custom payload, we successfully deployed it to the target system through social engineering tactics (see Figure 8). Upon execution of the planted executable on a target system, a reverse TCP connection is established from the target to the attacker's Kali Linux system, granting access to a Meterpreter shell (see Figure 9).

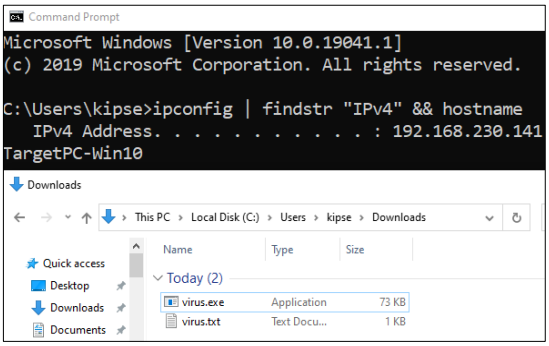

Figure 8 Payload deployment on Target machine

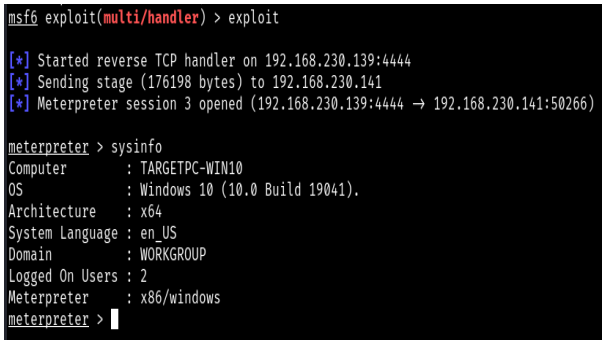

Figure 9 Gaining Target access with Meterpreter shell

The "ps -S virus.exe" command is executed in Meterpreter lists processes with the name "virus- .exe", to identify our payload running on the compromised system (see Figure 10).

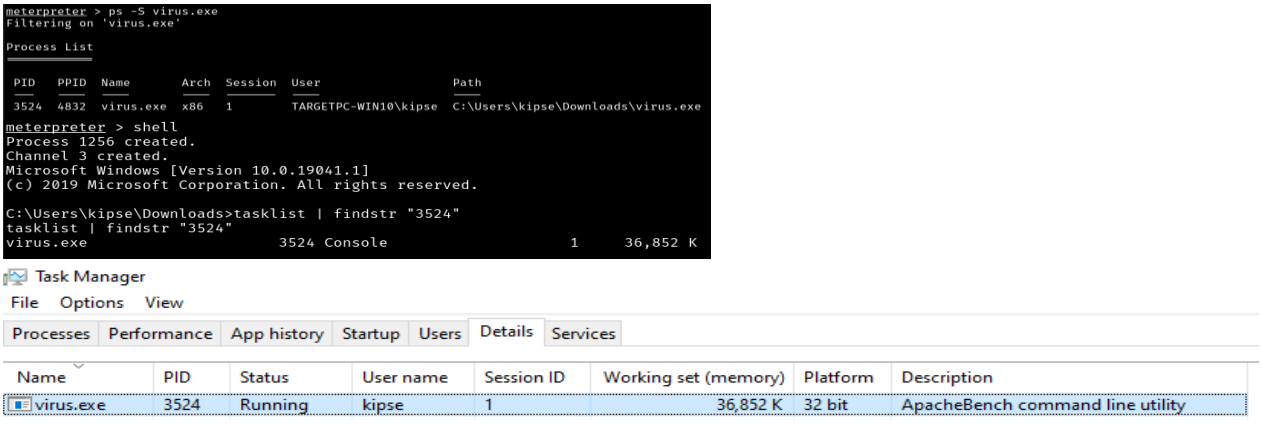

Figure 10 Displaying list of running process from Attacker and Target machines

Health Sciences Journal
EISSN: 2959-2259; PISSN: 2959-2240

The "shell" command executed in Meterpreter opens a new interactive command shell on the compromised system, granting direct access to the system's command line interface. This allows navigation of the file system, enabling actions such as uploading backdoors and copying or accessing sensitive documents. (see Figure 11). The "screenshare" command executed in Meterpreter initiates a remote desktop session, enabling the attacker to view and interact with the graphical user interface (GUI) of the compromised system in real time. This enables the execution of further commands for additional exploitation to accomplish the attacker's objectives (see Figure 12).

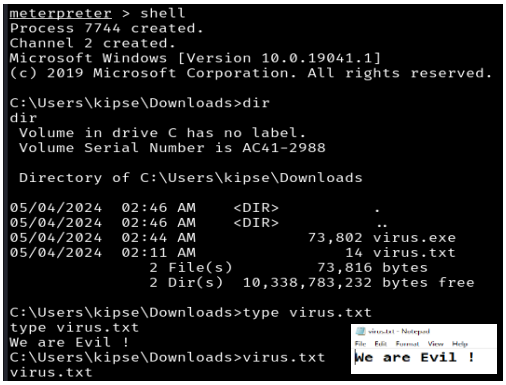

Figure 11. Remote access of Target system

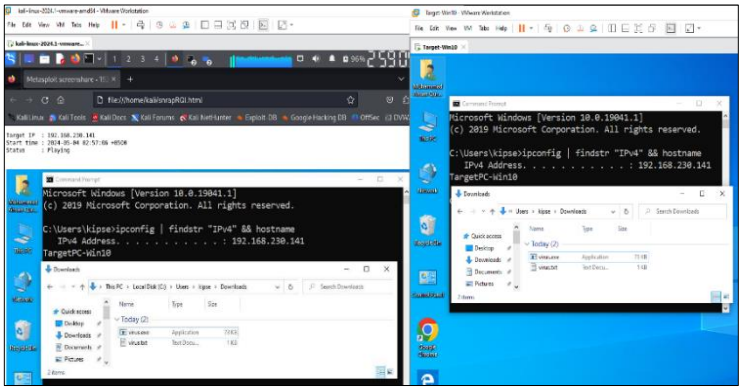

Figure 12. Real-time screenshare of Target system

4 | RESULTS AND DISCUSSION

Vulnerability assessment (VA) in healthcare systems is crucial for enhancing security posture by identifying weaknesses and potential entry points within systems or networks. It allows organizations to proactively address vulnerabilities in the healthcare systems before they can be exploited by malicious actors, thus reducing the risk of cyber-attacks and data breaches. Turning off the firewall and antivirus revealed that the targeted version of Windows lacks OS-level vulnerabilities. While disabling security measures may offer insights into system vulnerabilities, it doesn't account for all potential attack vectors. However, it is crucial to acknowledge the limitations of automated vulnerability assessments. Not a single VA tool can give 100 percent accurate results; there are chances of false positives. Security analysts should also perform security testing manually. Most attacks are successful due to untrained users and their inadequate security knowledge. Running specially crafted malicious files on Windows machines is met with a relatively reliable defense mechanism.

5 | CONCLUSION AND IMPLICATIONS

This study assumes the worst-case scenario in the healthcare systems where the user disables the firewall and antivirus. Limitations in user security awareness and vulnerabilities in the OS and other software are the primary causes of system compromise. A future study should explore different vulnerability assessment tools and attack approaches in healthcare systems, including creating specialized backdoors and maintaining persistence through them. Finally, the study will cover digital forensics techniques to investigate such compromised systems and recommend the best mitigation and prevention practices.

REFERENCES

1. Boyanov P. Vulnerability penetration testing the computer and network resources of windows based operating systems: vulnerability penetration testing the computer and network resources of windows based operating systems. Journal scientific and applied research, 2014; 5(1): 85-92.
2. G Quilantang KA, C Rivera JA, M Pinili MV, R Magpantay AJN, Busia Blancaflor E, & M Pastrana JR A. Exploiting Windows 7 vulnerabilities using penetration testing tools: A case study about Windows 7 vulnerabilities. In Proceedings of the 9th International Conference on Computer and Communications Management, 2021. (pp. 124-129).
3. Zegzhda PD, Zegzhda DP, & Kalinin MO. Vulnerabilities detection in the configurations of MS Windows operating system. In International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security 2005; (pp. 339-351). Berlin, Heidelberg: Springer Berlin Heidelberg.
4. Sharma H. Exploiting vulnerabilities of Metasploitable 3 (Windows) using Metasploit framework.2020

5. Faturrohman M, Salsabila A, Mardiah Z, & Kardian AR. Attack in to The Server Message Block (CVE-2020-0796) Vulnerabilities in Windows 10 using Metasploit Framework. JEEMECS (Journal of Electrical Engineering, Mechatronic and Computer Science), 2023; 6(1): 37-44.
6. Sreerag M, Sethumadhavan M, & Amritha PP. Identifying and Mitigating Vulnerabilities of Hardened Windows Operating System. In Information and Communication Technology for Competitive Strategies (ICTCS 2020) ICT: Applications and Social Interfaces 2022; (pp. 623-632). Springer Singapore.
7. Novokhrestov A, Kalyakin A, Kovalenko A, & Repkin V. Creating a vulnerable node based on the vulnerability MS17-010. arXiv preprint arXiv:2401.14979.2024
8. Kaluarachchilage PKH, Attanayake C, Rajasooriya S, & Tsoko CP. An analytical approach to assess and compare the vulnerability risk of operating systems. International Journal of Computer Network and Information Security, 2020; 12(2), 1.
9. Softić J, & Vejzović Z. Operating Systems Vulnerability-An Examination of Windows 10, macOS, and Ubuntu from 2015.
10. Karki R, & Tsokos CP. Cybersecurity: Identifying the Vulnerability Intensity Function (VIF) and Vulnerability Index Indicator (VII) of a Computer Operating System. Journal of Information Security, 2022; 13(4):337-362.
11. Đuranec A, Topolčić D, Hausknecht K, & Delija D. Investigating file use and knowledge with Windows 10 artifacts. In 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) 2019; (pp. 1213-1218). IEEE.
12. Arambatzis T, Lazaridis I, & Pouros S. Modern Windows Operating Systems Vulnerabilities. In The Second International Conference on Information Security and Digital Forensics 2015; (ISDF2015) (p. 53).
13. Gorbenko A, Romanovsky A, Tarasyuk O, & Biloborodov O. Experience report: Study of vulnerabilities of enterprise operating systems. In 2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE) 2017; (pp. 205-215). IEEE.
14. Ahmed, S., Ghashem, I. A., Aalsalem, M. Y., & Khan, W. Z. (2017, September). Thin Client Technology for Higher Education at Universities of Saudi Arabia: Implementation, Challenges and Lesson Learned. In 2017 International Conference on Computer and Applications (ICCA) (pp. 195-199). IEEE.
15. Softić J, & Vejzović Z. Windows 10 Operating System: Vulnerability Assessment and Exploitation. In 2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH) (pp. 1-5). IEEE.2022
16. Alenezi FN, & Mehmood T. Data-driven Predictive Model of Windows 10's Vulnerabilities. In 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) 2022; (pp. 1-5). IEEE.
17. Ahmed S, & Moukali KH. An efficient implementation of thin client technology for e-learning in the Jazan University. In 2014 International Conference on Web and Open Access to Learning (ICWOAL) 2014; (pp. 1-6). IEEE.
18. Arambatzis T, Lazaridis I, & Pouros S. Modern Windows Operating Systems Vulnerabilities. In The Second International Conference on Information Security and Digital Forensics (ISDF2015) 2015; (p. 53).
19. Guo H, Wang YY, Pan ZL, & Liu SW. Research on detecting windows vulnerabilities based on security patch comparison. In 2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC) 2016; (pp. 366-369). IEEE.
20. Vogels W. File system usage in Windows NT 4.0. ACM SIGOPS Operating Systems Review, 1999; 33(5): 93-109.
21. Ahmed S, Noor ASM, Khan WZ, Mehmood A, Shaheen R, & Fatima T. Students' Perception and Acceptance of e-Learning and e-Evaluation in Higher Education. Pakistan Journal of Life & Social Sciences, 2023; 21(1).
22. Joh H. Software risk assessment for windows operating systems with respect to CVSS. European Journal of Engineering and Technology Research, 2019; 4(11): 41-45.