Penetratingz the Defenses: An Investigation into the Achilles' Heel of HTTP and SSH

Atharva Prashant Joshi [1], Shalini Bandrevu[2], Navneet Kaur[1], Ishu Sharma[3], M.W.P. Maduranga[4], WAAM Wanniarachchi[5]

Department of Computer Science and Engineering, Lovely Professional University, Phagwara, India. , Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology, Hyderabad, India., Chandigarh Group of Colleges Jhanjeri, Mohali, Punjab, India – 140307 Chandigarh Engineering College, CSE-APEX, Dept. of Electrical and Electronic Engineering, Faculty of Engineering, University of Sri Jayewardenepura, Nugegoda, Sri Lanka., Department of Information Technology, Faculty of Computing, General Sir John Kotelawala Defence University.

atharvajoship2003@gmail.com, bandrevushalini0317@gmail.com, navneetphul@gmail.com, ishu.sharma001@gmail.com, pasanm@sjp.ac.lk, ashenw@kdu.ac.lk

ABSTRACT

As the yber threat increases, it becomes necessary for organizations to start securing their digital valuables and infrastructures. This research is mainly about analysing the weaknesses lying in the vulnerabilities of the HTTP and SSH protocols. Investigation here is into how the intruder can escalate his privileges and illegally access computers. Under open-source tools like Netcat and Gobuster, the article examines the vulnerability-assessment methodologies culminating in root access to the target machine. This paper emphasizes the need for proactive security measures and gives recommendations on improving defences against future attacks. The study, as bright as the findings may be, awaits empirical dimensions to affirm the proposed measures.

KEYWORDS: *Cybersecurity, HTTP vulnerabilities, SSH vulnerabilities, privilege escalation, reverse shell, penetration testing.*

INTRODUCTION

Cybersecurity is fast becoming a major issue for companies in all industries, given this digital world. The increasing number of Internet-connected devices, along with the services provided by them, has also initiated a parallel increase in the number and complexity of attacks targeting the fundamental network protocols, namely HTTP and SSH [1]. Such fundamental protocols, which are essential to communication online and secure remote access, are the common targets of cyberattacks that take advantage of misconfiguration, unfixed flaws, or even a lack of systems relying on them [2]. This is why protecting these protocols becomes extremely important in the protection of delicate data and the continuation of operations of contemporary organizations [3]. Although they form a key part of any computing environment, HTTP and SSH still feature several potential exploit vectors. As an example of vulnerability, we can cite the easy targeting of HTTP, which is used so widely in web-based communication and is prone to configuration mishaps or even inadvertent directory listing [4]. In contrast, SSH can be a liability when improperly configured, such as either via the use of weak authentication, or open firewall ports, which may be exploited by malicious users [5]. This code is mostly targeted in order to execute unauthorized access, privilege escalation, and take down whole systems [6]. The risks can hence be minimized given that the method of cyberattacks is always developing with time. The project employs practical tools based on penetration testing to investigate the vulnerability of HTTP and SSH and evaluate the security status of targeted systems with the use of free programs, such as Netcat and Gobuster to detect vulnerabilities that can be exploited in order to gain unauthorized access [7]. The techniques used present practical application on how attackers

https://doi.org/10.54389/MGQF6691

can escalate their privileges, and several root access to malicious machines, thus providing useful information on how the exploitation of machines can be done. It is also the goal of the proposed study to improve the general knowledge of efficient protection methods rather than focusing on understanding only the HTTP and SSH vulnerabilities [8]. The other significant contribution made by the research is the illustration of Black-Box vulnerability assessment and exploitation methods in practical life. The study offers good information to a system defender on how to strengthen his system by explaining the methodologies that were followed in the exploitation of HTTP and SSH flaws [9, 10]. Also, this study provides some tactics for strengthening the defence, like the implementation of an effective configuration practice and proactive vulnerability scans [11].

RELATED WORK

RootAsRole David W. Chadwick et al. The paper [1] introduces a Linux Security module that aims to provide administrative privileges on a finer scale than, for example, the su or sudo command. In this module, role-based access is permitted and this means that administrators give certain capabilities to both a user, a group of users as well as applications thus limiting the chances of privilege escalation. It contains management commands to address roles and appraisal of application capabilities, improving control and limiting the possible exploitation of Linux systems. In the meantime, Keshav Kaushik et al. [2] pay attention to the reverse shell attack, under which attacked machines open the connections to the machines of an attacker. Their article describes how to install reverse shells, how to exploit application, network, or configuration vulnerability, and points out the dangers of data leakage and the loss of the entire system. Michael et al. [3] deal with the issue of privilege escalation in Linux-based computing clusters where the issue of security in cluster high-value computing is discussed. In their research, they suggest methods of detecting unauthorized privilege elevation and tools such as NVisionCC that can be used for this process, this action is a strategy commonly utilized by the attackers. Irma Resendez et al. [4] switch to the topic of digital forensics and describe the legal, technical, and procedural issues digital evidence processing rests on. They provide a broad range, including not only traditional computer forensics but also new areas such as network and cloud forensics, and place a particular focus on adhering to legal requirements when preparing and processing evidence and interpreting it during analysis. Kr. Boyanov et al. [5] present experimental contributions that involve practical applications of Netcat in Linux in terms of network security tests as it shows that controlling its operational exploits can be accomplished effectively in port scanning, banner capturing, and reverse shells. Likewise, Rizdqi et al. [6] introduce Sudomy, an efficient tool to enumerate and analyze the subdomains, which are important during reconnaissance stages of penetration testing. Lastly, Kaivan et al. [7] provide a comprehensive overview of the SSH protocol, including its combination of various cryptographic tools including the use of the public key infrastructure, symmetric encryption, and message authentication. They stress the contribution of SSH in facilitating safe communication of current remote and web services.

EXPERIMENTAL PHASE

The exploitation phase comprises four significant stages: reconnaissance, scanning, gaining access, and clearing trails. Information about the target system or network is obtained during the reconnaissance phase by the attackers. Next comes scanning to discover the weaknesses in readiness to be exploited. With such knowledge, the attacker uses the disclosed vulnerabilities and gets access to the district and establishes a foothold in it [14][15]. Finally, the attackers have their trail entered meticulously erased, thus minimizing the risk for detection, to conceal his existence as well as activity. Those four phases make up the foundation of the exploitation process, which enables an attacker to breach and navigate within target systems efficiently.

1.1. Reconnaissance or Information Gathering

Initial Gathering of Critical Information about the Target System Passively and Actively. We used netdiscover to run a search on the network for the Earth computer's IP address. The target's address ip found was 192.168.72.131. We also took nmap to perform a full scan of the open ports and services on this

machine. Two major services were discovered as follows: SSH on port 22 and HTTP on port 8080, which were later attacked. The above Fig 1 shows the steps undertaken to determine the IP address of the Kali Linux machine using the 'ifconfig' command. With this command, there is a detailed view of the IP addressing as well as other network interface related information. In Fig. 2, we used the command "netdiscover -r 192.168.72.0/24" to find the IP addresses within the given networks ranging from 192.168.72.0 to 192.168.72.255. This command actually scans and discovers devices in the network range specified. This is shown in Fig 3, which indicates the results obtained by executing the command in Fig 2, which describes which of the IP addresses were active on the network. After conducting thorough research, it was discovered that 192.168.72.131 is the designated target IP address for our future activities.

1.2. Scanning

After reconnaissance, a more intensive probing phase was launched to assess the vulnerabilities of exposed services. The nmap command was used with -sV, -sC, and -v options to determine service versions and run default scripts [16][17]. This resulted in effective and potentially exploitable information in the HTTP and SSH services identified. Gobuster, a very powerful directory brute-forcing tool, has been used for hidden directory and file enumeration on the HTTP service. Using SecLists' preconfigured wordlist, Gobuster discovered folders such as /admin and /cgi-bin, as well as sensitive files like .htaccess and robots.txt. These findings pointed to web server misconfigurations, which might provide entry points for exploitation. Here is command nmap -sV -sC -v -T4 192.168.72.131 in Fig. 4. This command has multiple options; "-sV" for port version detection, "-sC" for a default script scan, "-v" for activating verbose mode, and "-T4" for setting the timing template to level 4. The result of these scans is shown in Fig. 4 and Fig. 5, which will further give details regarding open ports, vulnerabilities present on the target machine, and the domain names "earth.local terratest.earth.local". These findings serve as a platform for further investigation and exploitation of the system's possible flaws.

1.3. Gaining access

With the target system's vulnerabilities identified, the next step was to acquire illegal access. The gobuster scan showed the existence of an admin panel for the HTTP service. We attempted to access the /admin directory and successfully signed in using the username "terra" and an encrypted password by brute-forcing login credentials obtained from the server. To further abuse the system, we used Netcat to create a reverse shell that allowed us to run commands on the Earth computer remotely. This was done by setting up a listener on the Kali Linux box, then configuring the target machine to connect back to this listener. With the following command nc -e /bin/bash, the reverse shell was launched, giving basic command-line access to the Earth computer. In Fig 6, we utilized the "nano /etc/hosts" command to access and modify the host file. This command enables the addition of IP addresses and corresponding domain names necessary for accessing the Earth Machines website. Fig 7 shows the command output from Fig 6 and the changes applied to the host file. These changes include the addition of the IP address 192.168.72.131 and the domain names "earth.local" and "terratest.earth.local." In Fig 8, we have searched for 192.168.72.131/earth.local.we are actually examining the domain earth.local which we discovered in the fig4 and fig5. In Fig 9, we see a message box, a message key and encrypted messages. In the message box and message key we have typed a random message to ensure that we are getting the output in the form of an encrypted message. These encrypted messages can be used in the further process of exploitation. In Fig 10, the command "gobuster dir -u http://earth.local/ -w /usr/share/wordlist/dirb/common.txt" was used. The '-w' denotes the wordlists used, whereas the '-u' denotes the URL of the target machine. The command result indicated the discovery of the 'admin' and 'cgi-bin' directories linked with the specified URL will be used in the further process of exploitation. In Fig. 11, we are accessing terratest.earth.local, which we have discovered from Fig. 4 and Fig. 5. After accessing that site, we can see that we are getting a warning message that this site is not secure, which means this site is using http protocol. In Fig. 12, the command "gobuster dir -u http://terratest.earth.local/ -k -w /usr/share/wordlist/dirb/common.txt" is used. '-w' indicates

the wordlists used, '-u' denotes the destination computer's URL, and '-k' disables SSL certificate checking. The command result indicated the discovery of the '/.hta', '/.htaccess', '/.htpasswd', '/cgi-bin', '/index.html', '/. robots.txt' directories linked with the specified URL. In Fig. 13, we access the '/.robots.txt' directory from the above Fig. 12. We got a list of subdirectories after examining all the subdirectories. We found that /testingnotes.* is leading us to further process of exploitation. In Fig. 14, we get to know that the encrypted messages in Fig. 9 are using the XOR algorithm, and testdata.txt was used to test encryption, and terra is used as a username in the admin portal, which we have discovered in Fig. 10**.** In Fig 15, we have gained access to the testdata.txt file identified in Fig 14. When we accessed testdata.txt, we discovered the XOR input key, which is critical for the next phases of the exploitation process.

1.4. Privilege Escalation

Once basic access had been achieved, the next step was to escalate privileges to get root access [18][19]. By examining files discovered during the scanning process, including testdata.txt, we discovered an XOR-based encryption mechanism used to safeguard passwords. We used the CyberChef program to decrypt these passwords, revealing the credentials required to elevate access. Further investigation discovered the availability of a vulnerable script (reset_root), which could be used to reset the root password. After getting access to the script, we updated the required files and ran it to successfully reset the root password. This enabled us to switch to the root user using the su root command, giving us complete administrative authority over the Earth computer. In Fig. 16, the CyberChef program is used to decode hexadecimal data using the XOR operation, with UTF-8 as the decryption key. The decryption key is determined from Fig. 15. After decrypting the complete hexadecimal sequence in Figure 8, the password for the username 'terra' is revealed. Fig. 17 shows that access to the earth.local/admin was achieved successfully. The required username, as shown in Fig. 14, has been matched with the correct password, as seen in Fig. 16. In Fig. 18, we successfully secured administrative control of earth.local. Upon logging into the admin portal, we gained command line interface (CLI) access, allowing us to communicate with the earth.local servers.In Fig 19, we ran the simple command 'whoami,' and the username was detected as 'apache.'. In Fig. 20, the command "nc -e /bin/bash 192.168.72.133 4444" is used. This command makes use of a tool known as netcat, which is widely used to establish remote access via a reverse shell or bind shell. Certain command-line restrictions are in place during the exploitation process to improve the security of reverse shell connections. To properly gain reverse shell access, the complete command must be encoded before being executed in the command shell.

In Fig. 21, an echo and base24 scripted message are used. This entails encrypting the command shown in Fig. 20 and then decrypting it using base24. The decryption command is then performed to allow remote access to the machine. In Fig. 22, we configured LVNP with the following options: 'l' for listen mode, 'v' for verbosity, 'n' to disable DNS or service resolution, and 'p' to define the source port. This configuration allowed us to obtain rudimentary terminal access. We also imported the Python modules "bin" and "bash" with the command python -c 'import pty; pty.spawn("/bin/bash")'. While browsing through numerous folders and executable files, we came upon 'usr/bin/reset_root'. However, on attempting to run the 'reset_root' file, it became clear that some triggers were missing. In Fig 23, we went to a separate terminal and used the netcat command on port 3333, which resulted in the successful download of the'reset root' file. In Fig 24, we examined the download status of the'reset_root' file. After confirming the successful download, we used the command "chmod +x reset_root" to grant the file executable permissions. Then, using the command "ltrace./reset_root," we traced the files executed by'reset_root' and discovered the lack of three critical triggers. In Fig. 25, we return to the previous terminal session from Fig. 10. We used the 'touch' command to create the missing files, ensuring that'reset_root' executed properly. In Fig. 26, we run the ' the'reset_root' file, which successfully resets the root password to "Earth". Apparently, in Fig. 27, the 'su root' command is to log in as a root user. With the access obtained, it enabled traversal of all the files on the machine and root capability through the entry of the password shown in Fig. 26.

RESULTS AND DISCUSSIONS

The penetration testing research experiments have portrayed how the commonplace utilization of the HTTP and SSH protocols can be used to earn unauthorized access and further privileges in a system [20][21]. Instead, we started the diversification by collecting information where Net Discover was used to detect the target IP (192.168.72.131), and a Nmap scan was conducted to check the result of finding two open ports, 22 (SSH) and 8080 (HTTP) [24]. This was used as the platform on which later scanning and exploitation were based. Vulnerability discovery employed Nmap to identify the service versions [25] and Gobuster to find hidden folders, including /admin, and/cgi-bin, as well as sensitive files, like. htaccess, all as a result of bad web server set-up [26][27][28]. These results were important to understand the weaknesses of the target system and possible routes to exploit it. Brute-forcing the /admin login with already known credentials was used as the means to gain access, along with a subsequent setup of the reverse shell through Netcat to perform commands on a remote machine [30][31]. Privilege escalation was reached by decrypting the password of the user terra in testdata.txt into CyberChef [32][33], and by abusing a vulnerable script, reset_root, that allowed to reset of the root password and subsequently achieved full control of the system [34][35][36]. Lastly, the logs were wiped after the exploitation to remove all evidence of the intrusion [37][38], which highlights the relevance of a proactive security approach, which includes protecting administrative scripts, properly securing sensitive information, having a solid monitoring and logging system, and so on; otherwise, the attack will go without a trace and it will not be mitigated in time.

CONCLUSION

It is clear from this study that there are important threats that exist within the HTTP and SSH protocols and that basic misconfigurations and risk practices are quite sufficient to cause a complete system breach. We could find some exposed services, exploit them with tools such as Netcat and Gobuster, and finally obtain root access, which was a privilege escalation. These observations indicate why there is a pressing need to constantly monitor and establish a proactive security position. The key learning points are that reconnaissance and vulnerability scanning form the base of any penetration testing process since they give the required information about the weaknesses of systems. This article revealed the mechanism through which publicly available information could be used by people to gain access (by finding exposed directories and improperly configured files). In addition to that, the relative conditions that allow reverse shells and brute-force attacks to be effective highlight the necessity of stricter authentication measures and configuration restrictions. The contribution of the study to the cybersecurity domain is that it provides a real-life case study that reflects the practice of exploitation of general security vulnerabilities in HTTP and SSH. Not only does it provide an example of how such attacks can be carried out, but it also provides workable information in reinforcement of defensive measures. It is advised to introduce strong authentication methods, periodically test the vulnerability of systems, and include stronger encryption along with limiting access to administrative scripts. In the future, it would be interesting to look further into the network defense mechanisms and the scenarios of more advanced attacks to enhance the understanding of threat mitigation. Also, real-time machine learning and automated monitoring systems might be highly beneficial with regard to early detection. With the ever-changing cyber threats, there is a need to balance the level of threat awareness and flexible security mechanisms in the bid to secure systems against complex attacks.

REFERENCES

1. Ahmad Samer Wazan, David W Chadwick, Remi Venant,Eddie Billoir, Romain Laborde, Liza Ahmad, Mustafa Kaiiali. RootAsRole: a security module to manage the administrative privileges for Linux. 2022.
2. Keshav Kaushik, Sakshi Aggarwalb , Shashank Mudgalc , Shubh Saravgid, Vibhor Mathure. A novel approach to generate a reverse shell: Exploitation and Prevention. 2021.
3. Michael Treaster, Gregory A. Koenig, Xin Meng, William Yurcik.Detection of Privilege Escalation for Linux Cluster Security. 2014.
4. Irma Resendez, Pablo Martinez, and John Abraham.An Introduction to Digital Forensics. 2014.
5. Petar Kr. Boyanov.Basic network penetration testing with the network tool Netcat in Linux-based Operating Systems. 2023.
6. Rizdqi Akbar Ramadhan, Redho Maland Aresta, Dedy Hariyadi.Sudomy: Information Gathering Tools for Subdomain Enumeration and Analysis.2020.
7. Kaivan Kaighobadi and Eduardo B. Fernandez.A Pattern for the Secure Shell Protocol.2012
8. Bruce A. Mah.An Empirical Model of HTTP Network Traffic.1997.
9. Toshihiro Yamauchi, Yohei Akao, Ryota Yoshitani , Yuichi Nakamura and Masaki Hashimoto.Additional Kernel Observer to Prevent Privilege Escalation Attacks by Focusing on System Call Privilege Changes.2018.
10. J. N. Goel and B. M. Mehtre, "Vulnerability assessment & penetration testing as a cyber defence technology," *Procedia Computer Science*, vol. 57, pp. 710-715, 2015.
11. Y. Liu, et al., "A novel exploit traffic traceback method based on session relationship," *CS & IT Conference Proceedings*, vol. 13, no. 7, 2023.
12. K. Kujanpää, W. Victor, and A. Ilin, "Automating privilege escalation with deep reinforcement learning," in *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, 2021.
13. B. Kallus, et al., "The HTTP Garden: Discovering parsing vulnerabilities in HTTP/1.1 implementations by differential fuzzing of request streams," *arXiv preprint arXiv:2405.17737*, 2024.
14. R. Andrews, D. A. Hahn, and A. G. Bardas, "Measuring the prevalence of the password authentication vulnerability in SSH," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, IEEE, 2020.
15. Y. Liu, et al., "An exploit traffic detection method based on reverse shell," *Applied Sciences*, vol. 13, no. 12, p. 7161, 2023.
16. Z. Hu, R. Beuran, and Y. Tan, "Automated penetration testing using deep reinforcement learning," in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, IEEE, 2020.
17. A. Johnson and R. J. Haddad, "Evading signature-based antivirus software using custom reverse shell exploit," in *SoutheastCon 2021*, IEEE, 2021.
18. I. Koniaris, G. Papadimitriou, and P. Nicopolitidis, "Analysis and visualization of SSH attacks using honeypots," in *Eurocon 2013*, IEEE, 2013.
19. S. W. Woo, et al., "Modeling vulnerability discovery process in Apache and IIS HTTP servers," *Computers & Security*, vol. 30, no. 1, pp. 50-62, 2011.
20. M. Monshizadeh, P. Naldurg, and V. N. Venkatakrishnan, "Mace: Detecting privilege escalation vulnerabilities in web applications," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014.
21. S. Bugiel, et al., "Towards taming privilege-escalation attacks on Android," in *NDSS*, vol. 17, 2012.
22. S. Bugiel, et al., "Xmandroid: A new android evolution to mitigate privilege escalation attacks," Technische Universität Darmstadt, Technical Report TR-2011-04, 2011.
23. V. Visoottiviseth, et al., "PENTOS: Penetration testing tool for Internet of Thing devices," in *TENCON 2017-2017 IEEE Region 10 Conference*, 2017.
24. N. Gavric, G. Bhandari, and A. Shalaginov, "Introducing the Slowloris E-DoS Attack: a Threat Arising From Vulnerabilities in the FTP and SSH Protocols," 2024.

25. A. Z. Agghey, et al., "Detection of username enumeration attack on ssh protocol: Machine learning approach," *Symmetry*, vol. 13, no. 11, pp. 2192, 2021.
26. W. Alsabbagh, et al., "Hacking the Backbone: Shell Reverse Attacks on IIoT Systems."
27. P. S. Shinde and S. B. Ardhapurkar, "Cyber security analysis using vulnerability assessment and penetration testing," in *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*, 2016.
28. D. Suciu, et al., "Horizontal privilege escalation in trusted applications," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020.
29. G. K. Sadasivam, C. Hota, and B. Anand, "Classification of SSH attacks using machine learning algorithms," in *2016 6th International Conference on IT Convergence and Security (ICITCS)*, 2016.
30. W. Qiang, et al., "PrivGuard: Protecting sensitive kernel data from privilege escalation attacks," *IEEE Access*, vol. 6, pp. 46584–46594, 2018.
31. I. Yaqoob, et al., "Penetration testing and vulnerability assessment," *Journal of Network Communications and Emerging Technologies (JNCET)*, vol. 7, no. 8, pp. 10–18, 2017.
32. S. Calzavara, et al., "Postcards from the post-http world: Amplification of https vulnerabilities in the web ecosystem," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.
33. P. M. Cao, et al., "{CAUDIT}: Continuous Auditing of {SSH} Servers To Mitigate {Brute-Force} Attacks," in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, 2019.
34. C. Ntantogian, et al., "NodeXP: NOde. js server-side JavaScript injection vulnerability detection and exploitation," *Journal of Information Security and Applications*, vol. 58, p. 102752, 2021.
35. S. N. H. B. Johari, et al., "Design of IMC-PID controller with fractional-order filter for steam distillation essential oil extraction process," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 2, pp. 801–810, 2021.
36. D. Trizna, et al., "Living-off-The-Land Reverse-Shell Detection by Informed Data Augmentation," *arXiv preprint arXiv:2402.18329*, 2024.
37. A. P. Joshi, N. Kaur, and S. Chauhan, "Encrypting the Unseen: Exploring Steganography Techniques in HTTP Environments," in *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2024.
38. A. P. Joshi, N. Kaur, and A. Kaur, "Unveiling the Achilles' Heel: A Comprehensive Exploration of Vulnerabilities in RMI and Bindshell Communication," in *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2024.