|  |  |
| --- | --- |
| Penetration Test Report Prepared for Hotel Dorsey | Corporate Logo Haverbrook security lab corporate logo  Name: Mark Go  Team Number: RedTeam 2  Student Number: Student 2 |

**INTRODUCTION**

This penetration test aimed to evaluate the security of the target system, with a particular focus on port 5432. The attack machine, with the IP address 10.2.2.50, was utilized to identify vulnerabilities and potential exploits. The test employed various tools, including Metasploit, to assess the system's security posture. This report outlines the findings and recommendations derived from the penetration test.

**TARGET SYSTEM INFORMATION**

|  |  |
| --- | --- |
| **Target System** | **Penetration Testing System (Attacker)** |
| IP Address: 10.2.2.100 Operating System: Linux Metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:59:00 UTC 2008 i686 GNU/Linux | IP Address: 10.2.2.50  Operating System: Linux kali 4.19.0-kali5-amd64 #1 SMP Debian 4.19.37-2kali1 (2019-05-15) x86\_64 GNU/Linux |

Initial scan using Zenmap/nmap revealed several open ports and associated services on the target system. Among these, the most important ones are:

* **21/tcp (FTP)**: Port 21 is open for File Transfer Protocol (FTP), allowing for the transfer of files between the client and the server.
* **22/tcp (SSH)**: Port 22 is open for Secure Shell (SSH), a secure remote access protocol used for secure logins and encrypted file transfers.
* **25/tcp (SMTP)**: Port 25 is open for Simple Mail Transfer Protocol (SMTP), which is essential for email communication.
* **53/tcp (Domain)**: Port 53 is open for DNS (Domain Name System), responsible for translating domain names into IP addresses.
* **80/tcp (HTTP)**: Port 80 is open for Hypertext Transfer Protocol (HTTP), facilitating web server communication and website access.
* **111/tcp (RPCbind)**: Port 111 is open for RPCbind, which manages remote procedure calls (RPC) on the network.
* **139/tcp (NetBIOS-SSN)**: Port 139 is open for NetBIOS Session Service, often used for Windows file and print sharing.
* **445/tcp (Microsoft-DS)**: Port 445 is open for Microsoft-DS (Microsoft Directory Service), providing access to shared resources on Windows networks.
* **3306/tcp (MySQL)**: Port 3306 is open for MySQL, a popular relational database management system.
* **5432/tcp (PostgreSQL)**: Port 5432 is open for PostgreSQL, a powerful open-source relational database system.

These open ports represent a mix of essential network services, including web, email, file sharing, and database services. Analyzing these ports and their corresponding services is crucial for assessing the security and functionality of the target system.

**A screenshot of a computer

Description automatically generated**  
Figure 1. Result of Zenmap Scan.

In this penetration test exercise, the focus was on exploiting a known vulnerability in PostgreSQL version 8.3 [1], particularly targeting port 5432. The vulnerability leveraged the ability of the PostgreSQL service account on certain default Linux installations to write to the /tmp directory and load payloads from there, potentially enabling the execution of arbitrary code.

In addition to the PostgreSQL payload exploit, the penetration test exercise also includes an exploit targeting a vulnerability in udev versions prior to 1.4.1 [2] [3] [4]. This vulnerability arises from the lack of verification in these versions that netlink messages originate from the kernel. As a result, local users can potentially escalate their privileges.

**TOOLS USED**

This comprehensive penetration test harnessed a suite of powerful tools to assess and exploit vulnerabilities in the target system. Metasploit [5], the central framework, facilitated vulnerability identification, payload delivery, and post-exploitation activities, while PostgreSQL [6] served as the backend database, enhancing data organization and query capabilities. Nmap and its Metasploit-integrated counterpart, db\_nmap [7], enabled network reconnaissance and automated importation of scan results into the database. Zenmap provided user-friendly visualization of scan results, aiding in the identification of open ports and services. Base64 played a pivotal role in decoding encoded passwords, while John the Ripper [8] excelled at password cracking, revealing password weaknesses. The strategic use of these tools, including Metasploit, PostgreSQL, Nmap, db\_nmap, Zenmap, Base64, and John the Ripper, collectively ensured the test's effectiveness, enabling the identification of vulnerabilities, successful exploits, and actionable security recommendations for the target system.

**VULNERABILITY EXPLOITATION**

**Vulnerability Scan:**  
nmap was again used to perform a focused scan on the target machine port 5432. We determined that the installed PostgresSQL is version 8.3.x

A computer screen shot of a program

Description automatically generated  
Figure 2. Result of db\_nmap scan.

**Credential Theft:**

By using the ‘scanner/postgress/postgres\_login’ module in MetaSploit Framework, we were able to obtain valid credentials to the PostgreSQL database.

A screenshot of a computer

Description automatically generated  
Figure 3. Result of MetaSploit login scanner.

host origin service public private realm private\_type JtR Format

---- ------ ------- ------ ------- ----- ------------ ----------

10.2.2.100 10.2.2.100 5432/tcp (postgres) postgres postgres template1 Password

Figure 4. PostgreSQL credentials

**Payload delivery:**

Following discovery of credentials, a payload was delivered to the target machine by using the ‘linux/postgres/postgres\_payload’ exploit and a remote session was established.

A screenshot of a computer

Description automatically generated  
Figure 5. Metasploit session, postgres\_payload

**Exploitation and Root Shell Access:**

The udev exploit was a pivotal component of our penetration test, demonstrating how vulnerabilities in system components can be leveraged to execute payloads and gain root shell access.   
  
Exploiting the previously delivered payload, we gained root shell access and the ability to further investigate the system, exfiltrate data, and assess its overall security posture.

A screenshot of a computer

Description automatically generated  
Figure 6. Metasploit session, udev\_netlink exploit

**Data Exfiltration**

After gaining root shell access, it was trivial to view or download the /etc/shadow file and the /redteam2/student2/mypass.txt file.

A screenshot of a computer

Description automatically generated  
Figure 7. Metasploit session, data exfiltration.

**Decoding mypass.txt file**

In Fig. 8 below, the base64-decoding command "base64 -d" is applied to the content of "mypass.txt," which reveals the plaintext password "redteam2student2."

A screenshot of a computer

Description automatically generatedFigure 8. Decoding mypass.txt file.

**Decrypting shadow file**

Figure 9 below demonstrates the usage of the John the Ripper password cracking tool to analyze and crack password hashes stored in the "shadow.txt" file. For demonstration purposes, only four hashed passwords were processed instead of the entire shadow file. All four password hashes were successfully cracked. Any of these username and password pairs could be used to conduct further operations within the target machine.

A computer screen shot of a computer screen

Description automatically generatedFigure 9. Decrypting shadow file.

**CONCLUSION AND RECOMMENDATIONS**

This penetration test of Hotel Dorsey's systems, with a specific focus on port 5432, provided a thorough evaluation of the target system's security. We used various tools, including Metasploit, to identify vulnerabilities and potential exploits, and this report summarizes our findings and recommendations.

The examination revealed several open ports and services, with port 5432 (PostgreSQL) being a key point of interest. We successfully exploited a known vulnerability in PostgreSQL version 8.3, highlighting the importance of patch management and securing this service. Additionally, we demonstrated the risk associated with vulnerabilities in udev versions prior to 1.4.1, underlining the significance of regular system updates.

Our recommendations for Hotel Dorsey include:

**1. Patch and Update Regularly:** Implement a robust patch management process to keep all systems and software up to date and resilient against known vulnerabilities. [9]

**2. Enforce Strong Password Policies:** Enact stringent password policies, including complexity and rotation requirements, and consider implementing multi-factor authentication.[10]

**3. Enhance Database Security:** Strengthen PostgreSQL and other database security with access controls, strong authentication, and regular audits.[11]

**4. Periodic Security Audits:** Conduct routine security audits and vulnerability assessments to proactively address weaknesses. [12]

**5. User Training and Awareness:** Invest in comprehensive security awareness training to educate employees about potential threats and best practices.[13]

**6. Establish an Incident Response Plan:** Develop an incident response plan to effectively respond to security incidents.[14]

**7. Data Encryption:** Implement data encryption mechanisms to protect sensitive data both at rest and in transit.[15]

These measures will significantly enhance Hotel Dorsey's security posture, reduce the risk of breaches, and maintain the confidence of customers and stakeholders. Proactive security practices are essential in today's threat landscape, and these recommendations will help ensure the safety of critical assets.

**References**:

[1] CVE-2007-3280. (n.d.). CVE Details. <https://cvedetails.com/cve/CVE-2007-3280/>

[2] CVE-2009-1185. (n.d.). CVE Details. <https://cvedetails.com/cve/CVE-2009-1185/>

[3] OSVDB (53810). (n.d.). Open Sourced Vulnerability Database.

[4] BID 34536. (n.d.). SecurityFocus. <http://www.securityfocus.com/bid/34536>

[5] Metasploit. (n.d.). Metasploit Framework. [Online]. Available: <https://www.metasploit.com/>.

[6] "Metasploit Basics - Part 4: Connecting and Using the PostgreSQL Database with Metasploit," Hackers-Arise, March 29, 2017. [Online]. Available: <https://www.hackers-arise.com/post/2017/03/29/metasploit-basics-part-4-connecting-and-using-the-postgresql-database-with-metasploit>

[7] "Port Scanning - Metasploit Unleashed," Offensive Security. [Online]. Available: <https://www.offsec.com/metasploit-unleashed/port-scanning/>

[8] "John the Ripper - Official Openwall Release," Openwall, [Online]. Available: <https://www.openwall.com/john/>.

[9] T. Keary, "Patch Management: The Best IT Security Defense," CIO Dive, June 03, 2019. Available: <https://www.ciodive.com/news/patch-management-the-best-it-security-defense/555040/>.

[10] S. Brink, "The Password Policy Guide," NortonLifeLock. Available: <https://www.nortonlifelock.com/blogs/security-response/password-policy-guide>.

[11] PostgreSQL Documentation, "Chapter 19. Database Security," PostgreSQL. Available: <https://www.postgresql.org/docs/current/security.html>.

[12] N. K. Malani, "What Is a Security Audit?," NortonLifeLock. <https://www.nortonlifelock.com/blogs/security-response/security-audit>

[13] P. Martini, "Security Awareness Training: 7 Best Practices," Cybersecurity Insiders, October 22, 2020. Available: <https://cybersecurityinsiders.com/security-awareness-training-7-best-practices-26764/>

[14] P. S. Turner, "Incident Response Planning Guide: 6 Key Steps," Digital Guardian, April 14, 2020. Available: <https://digitalguardian.com/blog/incident-response-planning-guide-6-key-steps>.

[15] U. S. Department of Health & Human Services, "Security 101: Data Encryption," HealthIT.gov. Available: <https://www.healthit.gov/faq/what-data-encryption-and-how-does-it-work>.