|  |  |
| --- | --- |
| Penetration Test Report Prepared for Hotel Dorsey | Corporate Logo Haverbrook security lab corporate logo  Name: Mohammad Raza Hakeem  Team Number: 7  Student Number: 3 |

**Introduction**

In this section, provide an overview and discuss the scope of the penetration test. Note the name of the operating system of the attack machine and discuss the tools that you will use (e.g., Zenmap, Metasploit). Keep in mind that this report will be reviewed by nontechnical people who may not know about the tools.

This penetration test aims to find security weaknesses in the **vsftpd (Very Secure FTP Daemon) service** on the target system. The goal is to check if an attacker can exploit any vulnerabilities and gain unauthorized access.

The **attack machine** used for this test runs **Kali Linux**, a popular operating system for ethical hacking and cybersecurity testing. The focus of this test is on scanning and exploiting the **vsftpd service**, which has had known security flaws in older versions.

### **Tools Used**

* **Zenmap** – A user-friendly version of **Nmap**, a tool that scans networks to find open ports and running services. Zenmap helps identify if **vsftpd** is active and accessible.
* **Metasploit Framework** – A powerful tool used to test security vulnerabilities. If a known exploit exists for **vsftpd**, Metasploit can be used to simulate an attack.
* **Netcat** – A simple tool for checking network connections and sending data. It helps verify if the **vsftpd** service is responding.

By using these tools, this test will help determine if **vsftpd** has any security risks.

**Target**

In this section, list the IP address and hostname of the target system as well as the IP address assigned to the attack machine. Focus on the open ports and describe the corresponding services associated with the open ports, placing this information into a table. Keep in mind that this report will be reviewed by nontechnical people who will not know about FTP or SSH or may not know what functions the services provide on a network. Provide a screenshot of the Zenmap scan. All screenshots and tables should be labeled for your report.

The IP address for the attacking machine (Kali) is 10.7.3.50. and for the victim machine (Metasploitable) it is 10.7.3.100.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

| **Port** | **Service Name** | **Description** |
| --- | --- | --- |
| **21** | **FTP (File Transfer Protocol)** | Used to transfer files between computers. It lacks encryption, making data transmission insecure unless protected by FTPS or SFTP. |
| **22** | **SSH (Secure Shell)** | Provides secure remote login and command execution. SSH encrypts data, making it safer than Telnet. |
| **23** | **Telnet** | Allows remote login but transmits data in plain text, making it highly insecure and replaced by SSH in most cases. |
| **25** | **SMTP (Simple Mail Transfer Protocol)** | Used for sending emails. Modern email servers use encryption (SMTPS) to enhance security. |
| **53** | **DNS (Domain Name System)** | Translates domain names (e.g., google.com) into IP addresses. DNS queries typically use UDP, while zone transfers use TCP. |
| **80** | **HTTP (Hypertext Transfer Protocol)** | Handles web traffic for websites. It is unencrypted, meaning sensitive data should be transmitted over HTTPS (port 443). |
| **111** | **RPCbind (Remote Procedure Call)** | Helps map RPC requests to the correct services on a system. It is commonly used in Unix/Linux environments. |
| **139** | **NetBIOS Session Service** | Supports file sharing and communication on older Windows networks (SMB over NetBIOS). |
| **445** | **SMB (Server Message Block)** | Used for file sharing and network communication on Windows systems. Exploitable in older versions (e.g., EternalBlue vulnerability). |
| **512** | **Remote Shell (rsh)** | Allows users to run commands on remote systems. It lacks encryption, making it insecure and largely obsolete. |
| **513** | **rlogin (Remote Login)** | Provides remote login access but is insecure since it transmits credentials in plaintext. |
| **514** | **rsh (Remote Shell – Again)** | Similar to port 512, allowing remote command execution but lacks security features. |
| **1099** | **Java RMI (Remote Method Invocation)** | Used for executing Java objects remotely. It can be a security risk if exposed to the internet. |
| **1524** | **Ingreslock** | Historically used for database access but is now often targeted for backdoor exploits. |
| **2049** | **NFS (Network File System)** | Allows remote file access in Unix/Linux environments. Misconfigured NFS shares can lead to unauthorized access. |
| **3306** | **MySQL Database** | The default port for MySQL database connections. Open exposure can lead to database breaches if not secured. |
| **5432** | **PostgreSQL Database** | The default port for PostgreSQL database connections, which should be secured against unauthorized access. |
| **6667** | **IRC (Internet Relay Chat)** | Used for online chat services. Unsecured IRC servers can be exploited for botnet communication. |
| **8009** | **Apache JServ Protocol (AJP13)** | Connects web servers with application servers (e.g., Apache Tomcat). Vulnerabilities can allow attackers to bypass security controls. |
| **8180** | **Tomcat Web Server** | Alternative HTTP port for Apache Tomcat. If misconfigured, it can expose sensitive administrative functions. |

### **Security Considerations**

* **Insecure services**: Ports **21 (FTP), 23 (Telnet), 25 (SMTP without encryption), 512-514 (rsh, rlogin), and 6667 (IRC)** are often insecure if exposed to the internet.
* **Database ports**: Ports **3306 (MySQL) and 5432 (PostgreSQL)** should be restricted to trusted users only.
* **Remote access risks**: Ports **22 (SSH) and 3389 (RDP, not listed here but relevant)** should be protected with strong authentication and firewalls.

**Vulnerability**

In this section, discuss the specific vulnerability you are using to exploit the remote system. Provide a description of the vulnerability and how you used the vulnerability to exploit the victim machine. Include the time of intrusion as well as a screenshot of your connection to the victim system. Finally, discuss the level of access you achieved on the box (Apache, root, etc.). Show screenshots using the *getuid* or *whoami* command to verify your level of access for your client (two screenshots). All screenshots and tables should be labeled for your report.

Port 21 is used for **File Transfer Protocol (FTP)**, which allows computers to transfer files over a network. This service is commonly used for uploading and downloading files from remote servers, such as managing website files or sharing documents. FTP works in a client-server setup, where a user connects to a server to transfer files. However, standard FTP does not encrypt data, making it vulnerable to security risks unless additional protections like FTPS or SFTP are used.

After exploiting the vsftpd 2.3.4 backdoor, I gained root access to the victim machine. This allowed me to execute commands with administrative privileges, explore the file system, and gather sensitive information, such as passwords stored in /etc/passwd and /etc/shadow.

This screenshot below shows the whoami command then getting access to root.

A screenshot of a computer

Description automatically generated

**Data Exfiltration**

In this section, explain how you were able to take the company’s sensitive data out of the network. Provide a screenshot of the method by which the data was extracted. Talk about the implications to the company and possible costs of losing proprietary data.

I was able to get the password from a specific user and and crack the passcode, also to exfiltrate the password, I used the base64decode website, typing the code the needs to be cracked, after typing I clicked decode, and and return I got the password redteam3student3. A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Implications to the Company:**

Stealing sensitive data can have serious consequences for the company. If proprietary data, intellectual property, or customer information is lost, the company might face:

* **Financial Losses**: The company could lose money from legal fees, fines, or a drop in business due to damaged reputation.
* **Reputation Damage**: If customers find out that their data was stolen, they might lose trust in the company, which could lead to a loss of clients and partners.
* **Legal Issues**: The company might face lawsuits or penalties if laws protecting customer data, such as GDPR or HIPAA, are violated.

**Costs of Losing Proprietary Data:**

Losing proprietary data can be very costly for a company. The financial impact might include:

1. **Loss of Competitive Edge**: If intellectual property or trade secrets are stolen, competitors could use them to gain an advantage, harming the company’s market position.
2. **Regulatory Fines**: If the stolen data includes customer information, the company could face heavy fines for not securing it properly.
3. **Response Costs**: The company would need to hire experts to investigate the breach, which could be expensive.
4. **Customer Compensation**: The company may need to offer compensation to affected customers, which could further strain resources.

**Recommendations**

In this section, you will explain what security controls should be implemented to remediate the vulnerability that you have exploited on the remote system. Talk about the steps the client should take to ensure that the vulnerability is no longer present as well as any other suggestions you may have to help improve the overall security posture.

To fix the vulnerability exploited on the remote system, the client should follow several important steps to ensure the issue is resolved and improve the overall security of their system.

### **Steps to Fix the Vulnerability:**

1. **Update vsftpd Software**:  
   The client should immediately update **vsftpd** to the latest version. The vulnerability exploited in this case is present in version 2.3.4, so updating to a newer version that has patched the vulnerability will close the backdoor and prevent further attacks.
2. **Remove vsftpd if Not Needed**:  
   If the FTP service is not necessary, the client should disable or completely remove vsftpd. This can be done by stopping the service, disabling it from starting automatically, and uninstalling the software. This reduces the system’s attack surface by eliminating an unneeded service.
3. **Check for Other Signs of Compromise**:  
   The client should look for other backdoors or suspicious activity left by the attacker. This can include unauthorized user accounts, hidden files, or unusual changes to system files. Tools like chkrootkit or rkhunter can be used to scan for rootkits or other malicious software.
4. **Revoke Unauthorized Access**:  
   Any user accounts or credentials created by the attacker should be removed. The client should also change passwords for all accounts, especially for admin or root access, to ensure any stolen credentials are no longer valid.
5. **Review System Logs**:  
   The client should go through system logs to check for any signs of ongoing malicious activity. Logs such as /var/log/auth.log can provide valuable information on how the attacker gained access and what actions were taken after the breach.

### **Additional Security Measures:**

1. **Configure Firewalls and Network Segmentation**:  
   The client should configure firewalls to limit access to critical services, such as FTP, and only allow trusted IP addresses to connect. Network segmentation should also be used to separate sensitive systems from the rest of the network to reduce exposure to attacks.
2. **Enable Multi-Factor Authentication (MFA)**:  
   The client should implement **multi-factor authentication (MFA)** for accessing sensitive systems. MFA provides an extra layer of security by requiring more than just a password to gain access, which makes it harder for attackers to compromise accounts.
3. **Use Intrusion Detection and Prevention Systems (IDPS)**:  
   Installing an **intrusion detection and prevention system (IDPS)** will help the client monitor network traffic for signs of malicious activity. These systems can automatically detect and block attacks, offering an additional layer of defense.
4. **Apply Security Patches Regularly**:  
   The client should set up a schedule for applying security patches and updates. This ensures that any new vulnerabilities are fixed promptly and reduces the chances of future exploitation.
5. **System Hardening**:  
   To secure the system further, the client should follow best practices for hardening. This includes disabling unnecessary services, limiting user permissions to only what's necessary, and securing system configurations to minimize potential weaknesses.

By taking these steps, the client can effectively remove the vulnerability and strengthen the security of their systems, helping prevent future attacks and reducing the risk of data breaches.

## **References**

1. **VSFTPD Vulnerability and Fixes:**
   * "vsftpd 2.3.4 Backdoor Vulnerability." (2011). Exploit Database.  
     URL: <https://www.exploit-db.com/exploits/17496>
   * "vsftpd – Vulnerabilities and Patching." (2011). SecurityFocus.  
     URL: <https://www.securityfocus.com/bid/46555>
2. **System Hardening and Security Best Practices:**
   * "Security Hardening Guide for Linux Systems." (2020). SANS Institute.  
     URL: <https://www.sans.org/white-papers/136/>
   * "Linux Hardening: Best Practices for 2020." (2020). Linux Foundation.  
     URL: <https://www.linuxfoundation.org/blog/linux-hardening/>
3. **Intrusion Detection and Prevention:**
   * "Intrusion Detection Systems: An Overview." (2019). SANS Institute.  
     URL: <https://www.sans.org/white-papers/145/>
   * "Network Intrusion Detection and Prevention Systems." (2020). TechTarget.  
     URL: <https://www.techtarget.com/searchsecurity/definition/intrusion-detection-system-ID>
4. **Multi-factor Authentication (MFA):**
   * "Best Practices for Multi-Factor Authentication." (2021). Cybersecurity & Infrastructure Security Agency (CISA).  
     URL: <https://www.cisa.gov/mfa>
   * "Multi-Factor Authentication: The Basics." (2020). Microsoft.  
     URL: <https://www.microsoft.com/security/blog/2020/07/15/what-is-multi-factor-authentication-mfa/>
5. **Patching and Security Updates:**
   * "How to Manage Security Patches in Linux." (2020). Linux.com.  
     URL: <https://www.linux.com/training-tutorials/how-manage-security-patches-linux/>
   * "The Importance of Regular Security Patching." (2019). Cisco.  
     URL: <https://www.cisco.com/c/en/us/products/security/patch-management.html>