Project Report :

Identifying and Mitigating Network Vulnerabilities using OpenVAS

Submitted by

by TADI PURNA KARTHIK

Project Objective :

This report provides a comprehensive overview of OpenVAS, a widely-used vulnerability assessment tool, and its application in identifying and exploiting 5 vulnerabilities within the Metasploitable 2 virtual machine. The report outlines the key features of OpenVAS, its scanning capabilities, and the methods employed to analyze and mitigate the identified vulnerabilities. Additionally, it explores lessons learned and best practices for effective vulnerability assessment, concluding with recommendations for future work.

Introduction to the Metasploitable 2 VM and its purpose :

The Metasploitable 2 VM is a virtual machine designed specifically for penetration testing and security research. It comes pre-loaded with a wide range of vulnerabilities, making it an ideal environment for practicing ethical hacking techniques, learning about various exploits, and testing security tools. This virtual machine serves as a safe and controlled environment to experiment with and understand real-world security threats without risking harm to live systems.

Overview of the OpenVAS vulnerability assessment tool :

OpenVAS (Open Vulnerability Assessment System) is a free and open-source vulnerability scanner that utilizes the Nessus engine. It is a powerful tool designed for comprehensive security assessments of networks and individual systems. OpenVAS allows users to scan for vulnerabilities, identify potential security risks, and generate detailed reports on the findings. The tool supports a vast database of known vulnerabilities, regularly updated to include the latest threats. OpenVAS offers a user-friendly interface, making it accessible to security professionals and beginners alike.

Key Features :

- Automatic scanning of networks and systems for known vulnerabilities
- Extensive database of vulnerabilities that is regularly updated
- Ability to generate detailed reports on the findings of the security assessments

Advantages :

- OpenVAS is free and open-source, making it accessible to users with different budgetary constraints.
- The tool leverages the powerful Nessus engine, ensuring reliable and comprehensive vulnerability sc anning.
- OpenVAS provides a user-friendly interface, making it easy for both experienced security professionals and beginners to navigate and utilize its features effectively.

Scanning the Metasploitable 2 VM using OpenVAS

The first step in the assessment involved scanning the Metasploitable 2 VM using OpenVAS. This process involved configuring OpenVAS to target the specific IP address of the virtual machine and selecting the appropriate scan profile. The selected profile included a comprehensive set of vulnerability checks designed to uncover known vulnerabilities in common services and applications. The scan identified various vulnerabilities, including open ports, weak passwords, and outdated software versions.

Tips:

- Ensure network connectivity between OpenVAS and the Metasploitable VM (check firewalls and routing if necessary).
- Metasploitable 2 is designed for testing; never connect it to a public network.

Vulnerability Assessment Report for Metasploitable 2 VM :

This report focuses on vulnerabilities identified in specific services (FTP, Apache Tomcat, Samba, MySQL, Telnet) within a Metasploitable 2 VM using OpenVAS.

Top 5 Vulnerabilities :

1. FTP (vsftpd)

- **Vulnerability:** Potential for anonymous access, weak default credentials, or known exploits in the vsftpd version.
- **Impact:** Unauthorized access to files on the system, potential for data exfiltration or modification.
- **Recommendation:**
  - Disable anonymous FTP access.
  - Utilize strong, unique passwords for the FTP user account.
  - Update vsftpd to the latest version with security patches.
  - Implement proper access controls (e.g., chroot jails) to restrict user access to specific directories.

2. Apache Tomcat

- **Vulnerability:** Potential for:
  - **Remote Code Execution:** Exploiting vulnerabilities in Java libraries or misconfigurations in Tomcat's configuration.
  - **Cross-Site Scripting (XSS):** If web applications deployed on Tomcat are vulnerable to XSS.
  - **Denial of Service (DoS):** Through resource exhaustion attacks or improper input handling.
- **Imp act:**
  - Server compromise, data theft, system disruption, and potential for further attacks within the network.
- **Recommendation:**
  - Keep Tomcat and all Java libraries updated with the latest security patches.
  - Implement robust input validation and sanitization in web applications.
  - Configure Tomcat with appropriate security settings (e.g., disable unnecessary features, restrict access).
  - Regularly scan web applications for vulnerabilities.

3. Samba

- **Vulnerability:** Potential for:
  - **Unauthorized access:** Weak or default passwords for Samba accounts, insecure shares.

- **Privilege escalation:** Exploiting vulnerabilities in Samba's implementation to gain higher privileges.
- **Denial of Service:** Through overloading Samba services.
- **Imp act:**
    - Unauthorized access to shared files and resources.
    - Potential for data theft, modification, or deletion.
    - System instability and disruption.
- **Recommendation:**
    - Utilize strong, unique passwords for all Samba accounts.
    - Restrict access to shares based on the principle of least privilege.
    - Keep Samba updated with the latest security patches.
    - Regularly review and audit Samba configurations.

4. MySQL

- **Vulnerability:**
    - **Weak passwords:** Default or easily guessable passwords for the root user or other accounts.
    - **SQL injection:** If applications using MySQL are vulnerable to SQL injection attacks.
    - **Unauthorized access:** If MySQL is exposed to the network without proper firewall rules.
- **Imp act:**
    - Unauthorized access to sensitive data within the database.
    - Data manipulation, deletion, or theft.
    - Potential for further attacks within the network.
- **Recommendation:**
    - Enforce strong password policies for all MySQL accounts.
    - Regularly review and update MySQL to the latest version with security patches.
    - Implement proper access control mechanisms and least privilege principles.
    - Restrict network access to MySQL to authorized sources.

5. Telnet

- **Vulnerability:** Telnet transmits data in plain text, making it extremely insecure.
- **Imp act:**
    - Passwords and other sensitive information transmitted over Telnet are easily intercepted.
- **Recommendation:**
    - **Disable Telnet entirely.** Use more secure protocols like SSH for remote access.

Mitigation Plan for the found vulnerabilities :

Once the vulnerabilities were identified and their potential impact understood, steps were taken to mitigate them. This involved a combination of patching, configuring security settings, and implementing access controls. The vulnerabilities related to outdated software versions were addressed by installing the latest security patches for the affected services. Weak password policies were enforced, requiring users to use strong passwords that meet complexity requirements. Network access control measures were implemented to limit access to critical services and prevent unauthorized connections. These mitigation strategies significantly reduced the overall risk profile of the Metasploitable 2 VM.

1. FTP (File Transfer Protocol)

- **Risk**: FTP is an insecure protocol that transmits data (including passwords) in plaintext, which can be intercepted easily.

Mitigation:

- **Disable FTP**: If FTP is not needed, disable the service completely by running:

bash

```
sudo service vsftpd stop
```

```
sudo systemctl disable vsftpd
```

- **Use Secure Alternatives**: Replace FTP with **SFTP** (Secure FTP) or **FTPS** (FTP Secure) for encrypted communic ation.
- **Use Strong Authentication**: If FTP is essential, enforce strong passwords and limit user access.

2. Apache Tomcat

- **Risk**: Apache Tomcat may have outdated configurations, exposed directories, or weak default credentials. The default web management interfaces (like `/manager` or `/host-manager`) are often targeted.

Mitigation:

- **Update Tomcat**: Ensure Apache Tomcat is updated to the latest stable version.
- **Restrict Access to Tomcat Manager**: Limit access to the Tomcat management interface by IP address or disable it entirely:
  - Edit `conf/context.xml` to restrict access:

xml

```
<Context ...> <Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+\.\d+|::1" /> </Context>
```

- **Use Strong Credentials**: Set strong passwords for the Tomcat manager, and avoid default credentials.

3. Samba (SMB File Sharing)

- **Risk**: Samba shares files and directories over the network, and if misconfigured, it could expose sensitive data or allow remote code execution via vulnerabilities like **EternalBlue**.

Mitigation:

- **Disable Samba if not needed**:

bash

```
sudo service smbd stop sudo systemctl disable smbd
```

- **Limit Samba Shares**: If Samba is needed, restrict access to authorized IP addresses and shares.
  - Edit `/etc/samba/smb.conf`:

ini

```
[global] hosts allow = 192.168.1.0/24
```

- **Use Strong Authentication**: Enforce strong user credentials and disable guest access.
- **Update Samba**: Ensure you are running the latest version of Samba to patch vulnerabilities.

4. MySQL

- **Risk**: MySQL may have weak default credentials (like `root` with no password) or be exposed to unauthorized access over the network.

Mitigation:

- **Set a Strong MySQL Root Password**:

bash

```
sudo mysql_secure_installation
```

- **Restrict MySQL Bind Address**: Ensure MySQL binds only to **localhost** (`127.0.0.1`) unless remote access is required.
  - Edit `/etc/mysql/my.cnf`:

ini
bind-address = 127.0.0.1

- **Limit Access**: Use proper user permissions and avoid granting ALL PRIVILEGES to users.
- **Keep MySQL Updated**: Regularly update MySQL to patch known vulnerabilities.

5. Telnet

- **Risk**: Telnet sends data, including usernames and passwords, in plaintext over the network, making it highly insecure.

Mitigation:

- **Disable Telnet**: If Telnet is not required, disable it:

bash

sudo service telnet stop sudo systemctl disable telnet

- **Use SSH Instead**: Replace Telnet with **SSH** (Secure Shell), which encrypts the communication.
  - Install OpenSSH:

bash

sudo apt install openssh-server sudo systemctl enable ssh

- **Use Strong SSH Keys**: Disable password authentication and enforce the use of SSH keys for authentic ation.

Lessons learned and best practices for vulnerability assessment

This project highlighted the importance of regular vulnerability assessments as an essential part of a comprehensive security program. The process of identifying, analyzing, and mitigating vulnerabilities provided valuable insights into the security posture of the Metasploitable 2 VM. The experience demonstrated the effectiveness of OpenVAS as a powerful vulnerability assessment tool, capable of uncovering even the most subtle vulnerabilities. It emphasized the need for a thorough analysis of the identified vulnerabilities, prioritizing those with the highest potential impact. The successful exploitation of these vulnerabilities underscored the importance of implementing effective mitigation strategies to address them promptly.

Conclusion and recommendations for future work

The vulnerability assessment of the Metasploitable 2 VM using OpenVAS proved to be a valuable exercise in understanding the process of identifying, analyzing, and mitigating security vulnerabilities. The project successfully demonstrated the effectiveness of OpenVAS as a powerful tool for vulnerability scanning and provided insights into the potential impact of vulnerabilities. Future work should focus on expanding the scope of vulnerability assessments to include a broader range of systems and networks. Continuously updating OpenVAS and Metasploit with the latest vulnerability databases and exploit modules is crucial for staying ahead of evolving threats. Implementing a comprehensive security program that includes regular vulnerability assessments, proactive patching, and strong access controls is essential for maintaining a secure operating environment.

1.Project Outcomes

This project successfully identified and mitigated numerous network vulnerabilities using OpenVAS. The comprehensive scanning and analysis provided valuable insights into the security posture of the network, enabling targeted and effective remediation efforts.

2.Future Work

Future work includes establishing a regular vulnerability assessment schedule, continuous monitoring for new threats, and ongoing improvements to the network security infrastructure. Adopting a proactive approach to network security will help maintain a robust and resilient network environment.