Security+ Lab Series

Lab 05: Performing Active Reconnaissance with Linux

Document Version: 2018-08-28

Contents

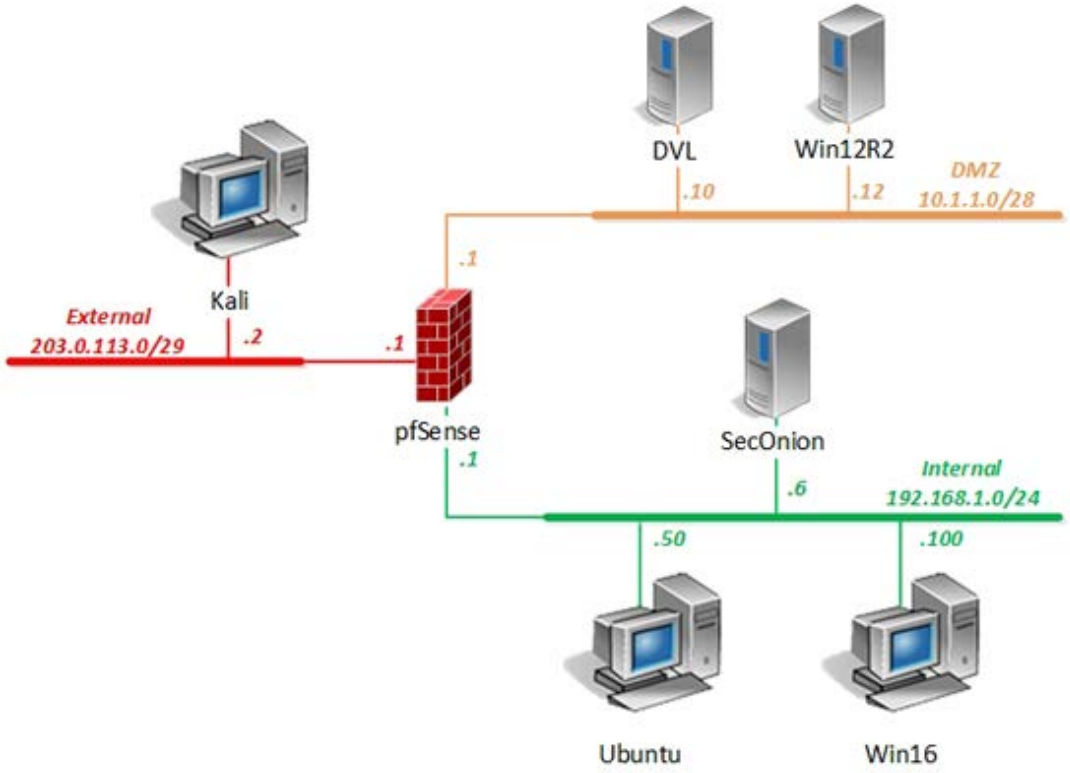
Introduction

In this lab, you will be conducting vulnerability scans using various network-scanning tools.

Objectives

- Explain vulnerability scanning concepts
- Given a scenario, use appropriate software tools to assess the security posture of an organization

Lab Topology



Lab Settings

The information in the table below will be needed to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address	Account	Password
DVL	10.1.1.10 /28	root	toor
Kali	203.0.113.2 /29	root	toor
pfSense	eth0: 192.168.1.1 /24 eth1: 10.1.1.1 /28 eth2: 203.0.113.1 /29	admin	pfsense
SecOnion	192.168.1.6 /24	soadmin	mypassword
		root	mypassword
Ubuntu	192.168.1.50 /24	student	securepassword
		root	securepassword
Win12R2	10.1.1.12 /28	administrator	Train1ng$
Win16	192.168.1.100 /24	lab-user	Train1ng$
		Administrator	Train1ng$

1 Scanning the Network for Vulnerable Systems

1.1 Scanning the Network Using Nmap

1. Launch the **Kali** virtual machine to access the graphical login screen.
2. Log in as **root** with **toor** as the password.
3. Open a new terminal window by clicking on the **terminal** icon located in the top toolbar.

4. View the available options that can be used with *Nmap* by typing ***nmap*** into the *terminal* followed by pressing the **Enter key**.

```
root@Kali-Attacker:~# nmap
Nmap 6.47 ( http://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
```

5. Initiate a **quick ping scan** to identify live hosts with a network ID of *10.1.1.**.

```
root@Kali-Attacker:~# nmap –sP 10.1.1.*
```

```
root@Kali-Attacker:~# nmap -sP 10.1.1.*

Starting Nmap 6.47 ( http://nmap.org ) at 2018-07-26 15:02 EDT
Nmap scan report for 10.1.1.1
Host is up (0.00027s latency).
Nmap scan report for 10.1.1.10
Host is up (0.00047s latency).
Nmap scan report for 10.1.1.12
Host is up (0.00037s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 14.24 seconds
```

You should see three host results; *10.1.1.1* as the *DMZ gateway*, *10.1.1.10* as the *DVL Server*, and *10.1.1.12* as the *Win12R2* server.

6. Initiate a **ping scan** while spoofing the source *MAC* address at the same time.

```
root@Kali-Attacker:~# nmap –v –sP –spoof-mac 0 10.1.1.*
```

```
root@Kali-Attacker:~# nmap -v -sP -spoof-mac 0 10.1.1.*

Starting Nmap 6.47 ( http://nmap.org ) at 2018-07-26 15:04 EDT
Spoofing MAC address B0:70:49:78:4C:32 (No registered vendor)
Initiating Ping Scan at 15:04
Scanning 256 hosts [4 ports/host]
Completed Ping Scan at 15:04, 1.23s elapsed (256 total hosts)
Initiating Parallel DNS resolution of 256 hosts. at 15:04
Completed Parallel DNS resolution of 256 hosts. at 15:04, 13.00s elapsed
Nmap scan report for 10.1.1.0 [host down]
Nmap scan report for 10.1.1.1
Host is up (0.00025s latency).
Nmap scan report for 10.1.1.2 [host down]
Nmap scan report for 10.1.1.3 [host down]
Nmap scan report for 10.1.1.4 [host down]
Nmap scan report for 10.1.1.5 [host down]
Nmap scan report for 10.1.1.6 [host down]
Nmap scan report for 10.1.1.7 [host down]
Nmap scan report for 10.1.1.8 [host down]
Nmap scan report for 10.1.1.9 [host down]
Nmap scan report for 10.1.1.10
Host is up (0.00045s latency).
Nmap scan report for 10.1.1.11 [host down]
Nmap scan report for 10.1.1.12
Host is up (0.00035s latency).
Nmap scan report for 10.1.1.13 [host down]
```

7. When scanning for active systems on a network, *Nmap* also gives the ability to scan for which *IP protocols* are supported by the host involved in the scanning process. Enter the command below.

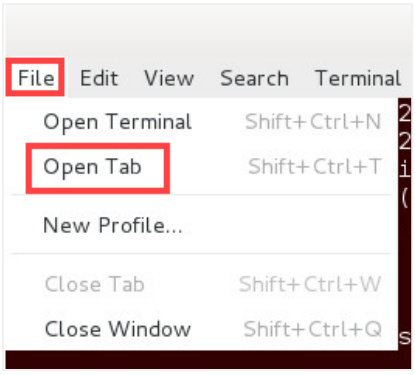
```
root@Kali-Attacker:~# nmap –sO 10.1.1.10
```

```
root@Kali-Attacker:~# nmap -sO 10.1.1.10

Starting Nmap 6.47 ( http://nmap.org ) at 2018-07-26 15:30 EDT
Warning: 10.1.1.10 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.1.1.10
Host is up (0.00058s latency).
Not shown: 250 closed protocols
PROTOCOL STATE         SERVICE
1        open          icmp
2        open|filtered igmp
6        open|filtered tcp
17       open          udp
58       open|filtered ipv6-icmp
136      open|filtered udplite

Nmap done: 1 IP address (1 host up) scanned in 291.84 seconds
root@Kali-Attacker:~#
```

Please Note

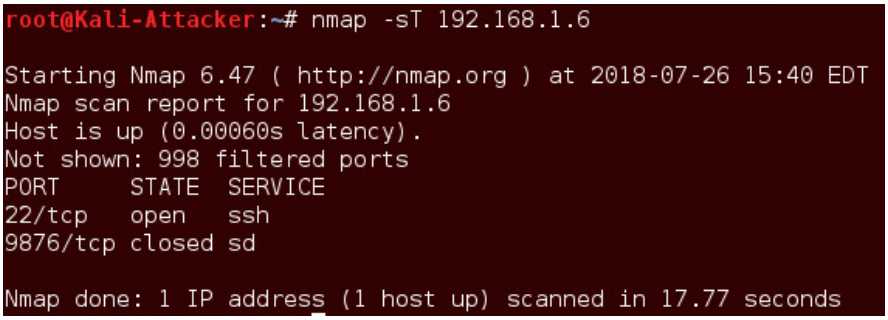
This scan can take up to five minutes to complete. You may proceed to the next step while the scan runs in the background. Once finished, notice the protocols that are running.

8. In the *terminal* window, select **File** from the top menu pane and click on **Open Tab**.



9. While engaged in the new tab, initiate a *Transmission Control Protocol (TCP)* scan against the **SecOnion** system. Type the following command below:

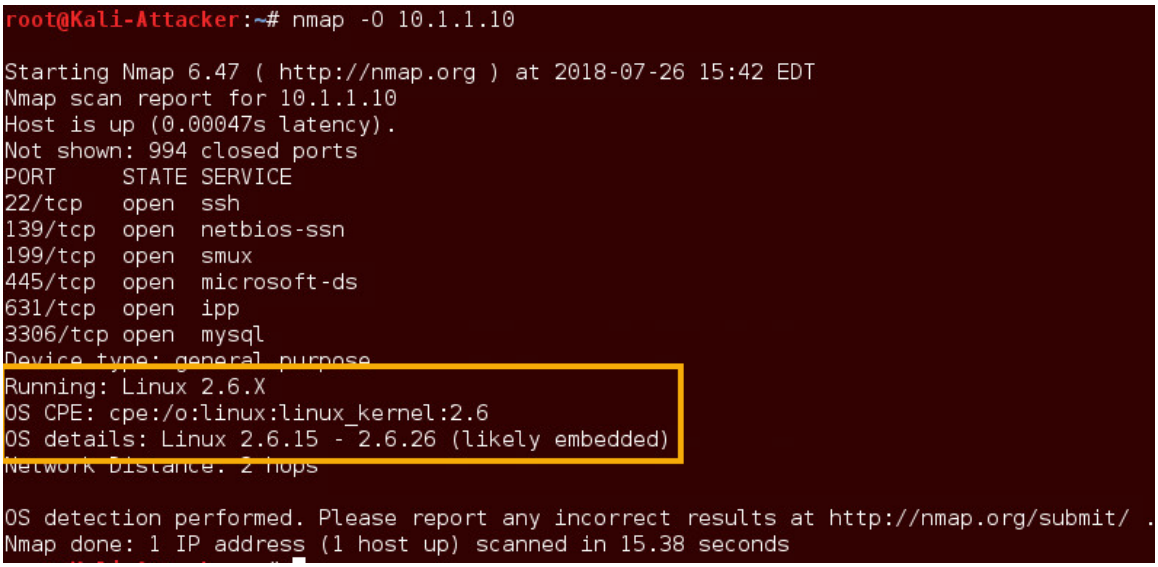
```
root@Kali-Attacker:~# nmap –sT 192.168.1.6
```



Notice the well-known TCP port that is open on the system (*SSH*).

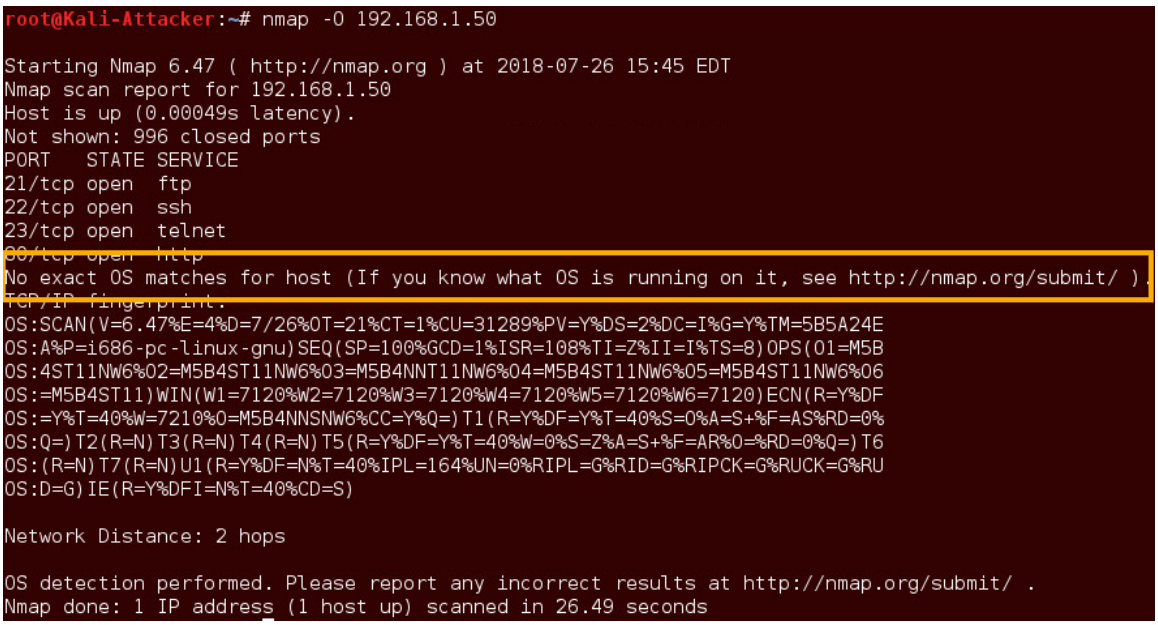
10. Initiate an **operating system scan** against the **DVL** system to help identify what version of *Linux* it is running on.

```
root@Kali-Attacker:~# nmap -O 10.1.1.10
```



11. Initiate the same scan from the previous step but this time against the **Ubuntu** system.

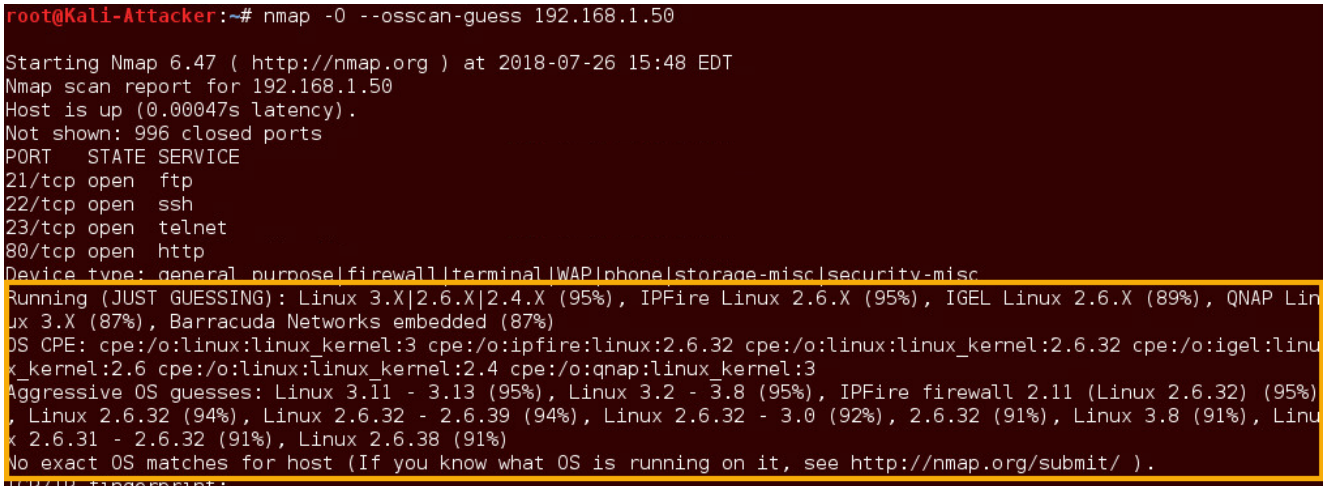
```
root@Kali-Attacker:~# nmap -O 192.168.1.50
```



Notice that *Nmap* has a tough time trying to identify the operating system information.

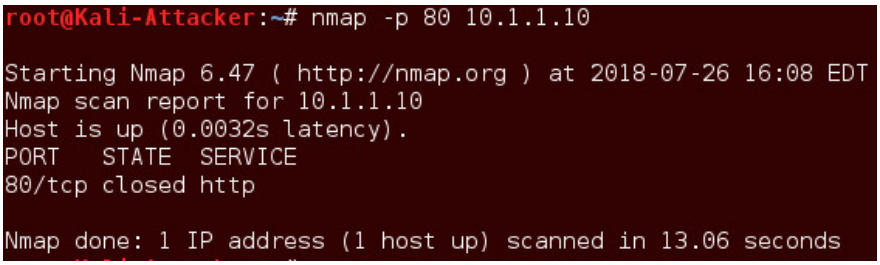
12. To try and gather more information about the same host regarding its *OS*, make *Nmap* take approximate guesses as to what the *OS* may be, by using the command below with an included script.

```
root@Kali-Attacker:~# nmap –O –-osscan-guess 192.168.1.50
```



13. Initiate a scan specifically for **port 80** against the **DVL** system.

```
root@Kali-Attacker:~# nmap –p 80 10.1.1.10
```



14. Initiate a scan specifically for **port 80** but this time for all hosts on both networks (*Internal & DMZ*).

```
root@Kali-Attacker:~# nmap –p 80 192.168.1.0/24 10.1.1.0/28
```

```
root@Kali-Attacker:~# nmap -p 80 192.168.1.0/24 10.1.1.0/28

Starting Nmap 6.47 ( http://nmap.org ) at 2018-07-26 16:10 EDT
Nmap scan report for 192.168.1.1
Host is up (0.00036s latency).
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for 192.168.1.6
Host is up (0.00071s latency).
PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 192.168.1.50
Host is up (0.00055s latency).
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for 192.168.1.100
Host is up (0.00053s latency).
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for 10.1.1.1
Host is up (0.00027s latency).
PORT   STATE SERVICE
80/tcp open  http

Nmap scan report for 10.1.1.10
Host is up (0.00082s latency).
PORT   STATE  SERVICE
80/tcp closed http

Nmap scan report for 10.1.1.12
Host is up (0.00083s latency).
PORT   STATE SERVICE
80/tcp open  http
```


15. Scan the **DVL** system while at the same time displaying all packets being sent and received while initiating the scan.

```
root@Kali-Attacker:~# nmap –-packet-trace 10.1.1.10
```

```
root@Kali-Attacker:~# nmap --packet-trace 10.1.1.10

Starting Nmap 6.47 ( http://nmap.org ) at 2018-07-26 16:12 EDT
SENT (0.0461s) ICMP [203.0.113.2 > 10.1.1.10 Echo request (type=8/code=0) id=24485 seq=0] IP [ttl=46 id=44959 ip
len=28 ]
SENT (0.0463s) TCP 203.0.113.2:63277 > 10.1.1.10:443 S ttl=47 id=34521 iplen=44  seq=2315796458 win=1024 <mss 14
60>
SENT (0.0464s) TCP 203.0.113.2:63277 > 10.1.1.10:80 A ttl=54 id=36565 iplen=40  seq=0 win=1024
SENT (0.0465s) ICMP [203.0.113.2 > 10.1.1.10 Timestamp request (type=13/code=0) id=1225 seq=0 orig=0 recv=0 tran
s=0] IP [ttl=39 id=65411 iplen=40 ]
RCVD (0.0467s) ICMP [10.1.1.10 > 203.0.113.2 Echo reply (type=0/code=0) id=24485 seq=0] IP [ttl=63 id=26236 iple
n=28 ]
NSOCK INFO [0.0470s] nsi_new2(): nsi_new (IOD #1)
NSOCK INFO [0.0470s] nsock_connect_udp(): UDP connection requested to 8.8.8.8:53 (IOD #1) EID 8
NSOCK INFO [0.0470s] nsock_read(): Read request from IOD #1 [8.8.8.8:53] (timeout: -1ms) EID 18
NSOCK INFO [0.0470s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [8.8.8.8:53]
NSOCK INFO [0.0470s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [8.8.8.8:53]
NSOCK INFO [4.0480s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 35 [8.8.8.8:53]
NSOCK INFO [8.0490s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 43 [8.8.8.8:53]
NSOCK INFO [13.0500s] nsi_delete(): nsi_delete (IOD #1)
NSOCK INFO [13.0500s] msevent_cancel(): msevent_cancel on event #18 (type READ)
SENT (13.0510s) TCP 203.0.113.2:63533 > 10.1.1.10:256 S ttl=45 id=37023 iplen=44  seq=1299286868 win=1024 <mss 1
460>
SENT (13.0510s) TCP 203.0.113.2:63533 > 10.1.1.10:110 S ttl=57 id=1991 iplen=44  seq=1299286868 win=1024 <mss 14
60>
SENT (13.0510s) TCP 203.0.113.2:63533 > 10.1.1.10:1025 S ttl=58 id=2105 iplen=44  seq=1299286868 win=1024 <mss 1
```

16. *Nmap* can also be used to show local host data about which interfaces are up and what the route table looks like. Enter the command below.

```
root@Kali-Attacker:~# nmap --iflist
```

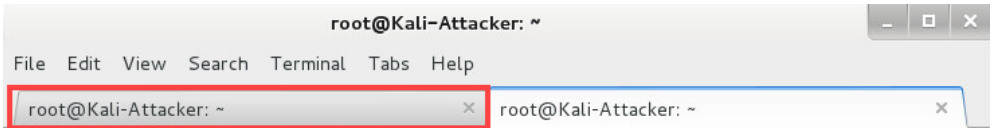
```
root@Kali-Attacker:~# nmap --iflist

Starting Nmap 6.47 ( http://nmap.org ) at 2018-07-26 16:18 EDT
************************INTERFACES************************
DEV  (SHORT) IP/MASK                     TYPE     UP   MTU   MAC
lo   (lo)    (none)/0                    loopback down 65536
eth0 (eth0)  203.0.113.2/29              ethernet up   1500  00:50:56:9C:FE:5B
eth0 (eth0)  fe80::250:56ff:fe9c:fe5b/64 ethernet up   1500  00:50:56:9C:FE:5B

**************************ROUTES**************************
DST/MASK                      DEV  METRIC GATEWAY
203.0.113.0/29                eth0 0
0.0.0.0/0                     eth0 0      203.0.113.1
fe80::250:56ff:fe9c:fe5b/128  eth0 0
fe80::/64                     eth0 256
ff00::/8                      eth0 256
```


17. To detect remote services, both *services* and *daemons*, along with their respective version numbers, initiate the *Nmap* command below.

```
root@Kali-Attacker:~# nmap –sV 10.1.1.10
```

```
root@Kali-Attacker:~# nmap -sV 10.1.1.10

Starting Nmap 6.47 ( http://nmap.org ) at 2018-07-26 16:20 EDT
Nmap scan report for 10.1.1.10
Host is up (0.00039s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 4.4 (protocol 1.99)
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
199/tcp  open  smux        Linux SNMP multiplexer
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
631/tcp  open  ipp         CUPS 1.1
3306/tcp open  mysql?
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 174.58 seconds
```

18. While on the *terminal* screen, switch back to the **first tab** by clicking on the tab.

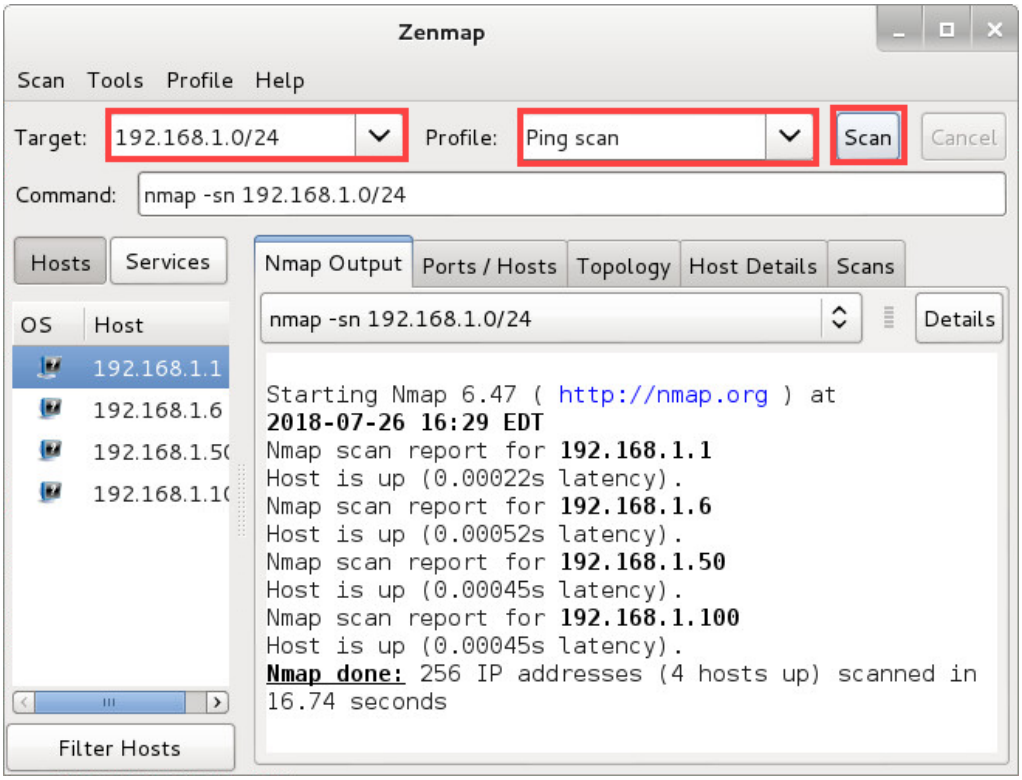


19. If you didn't wait for the scan to finish from *Step 7*, then the scan should be finished by now. Review the output of the IP protocols.
20. Leave the *Kali* window open to continue with the next task.

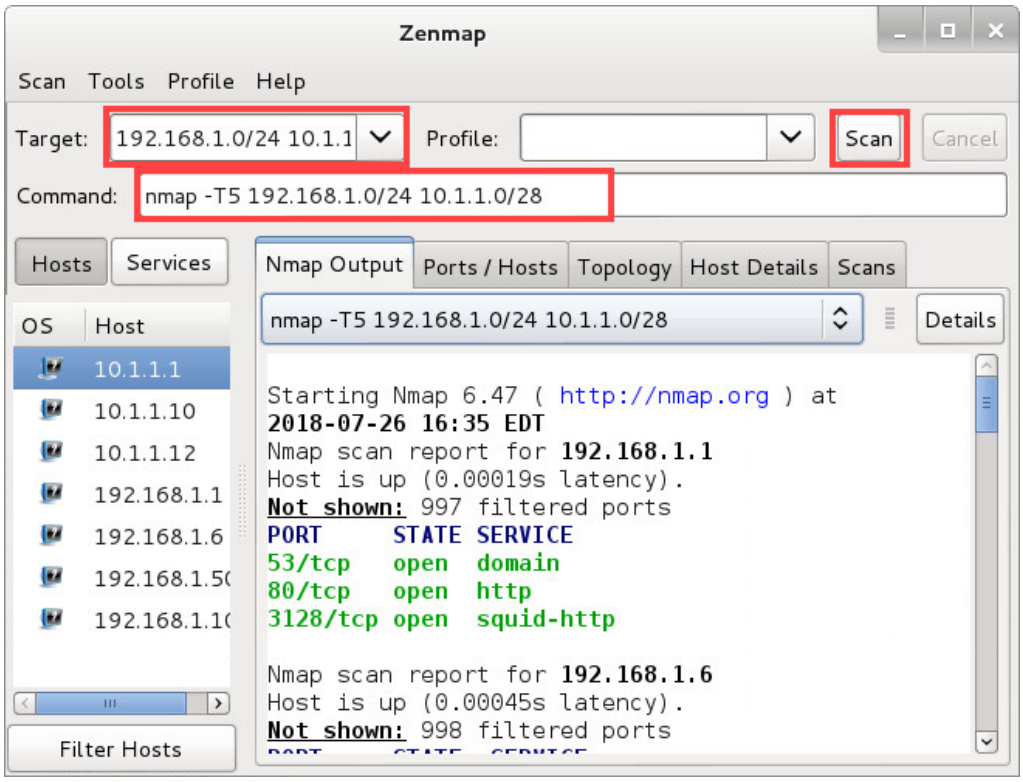
1.2 Scanning the Network Using Zenmap

1. While on the *Kali* system, focus on the terminal and type **zenmap** followed by pressing the **Enter** key. This will launch the *Zenmap* application from the terminal.

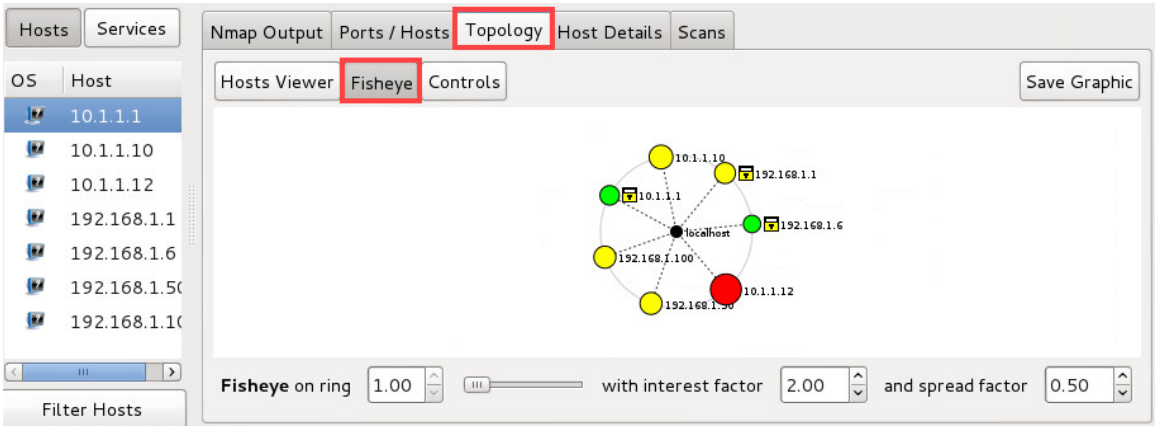
2. A new *Zenmap* window appears. Initiate a quick *ping scan* on the **192.168.1.0/24** network.
    a. Type **192.168.1.0/24** in the *Target* field.
    b. Choose **Ping scan** from the *Profile* drop-down menu.
    c. Click **Scan**.
    d. Notice the output.



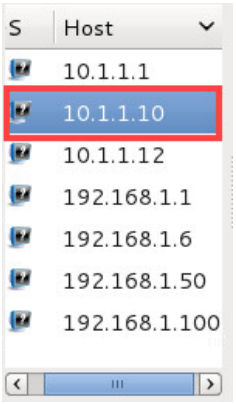
3. Initiate a new *Zenmap* scan for both networks: **192.168.1.0/24** and **10.1.1.0/28**.
   a. Type **192.168.1.0/24 10.1.1.0/28** in the *Target* field.
   b. In the *Command* field, remove the **-sn** option and add the **-T5** option so that the entire command reads **nmap -T5 192.168.1.0/24 10.1.1.0/28**.
   c. Click **Scan**.



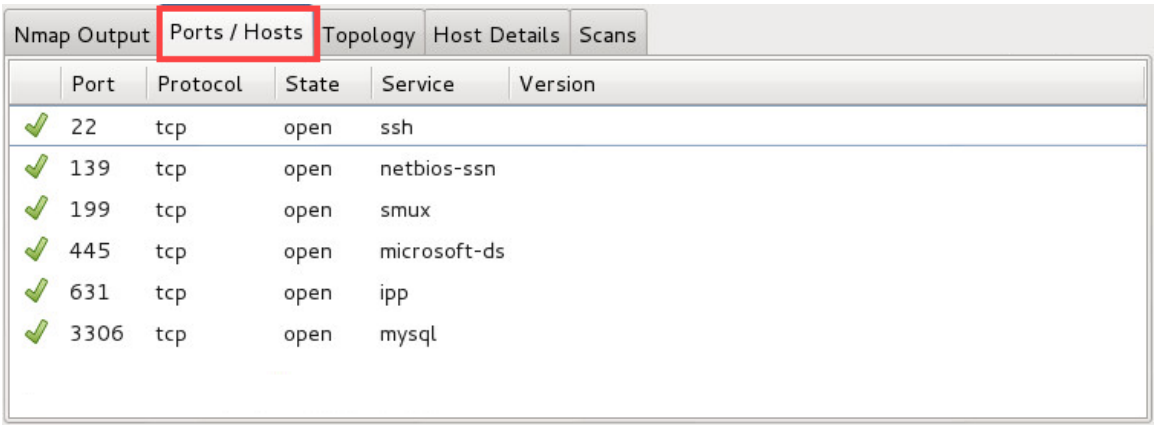
4. Once the scan is completed, click on **Topology** > **Fisheye** and view the content.



5. Select the **10.1.1.10** host from the *Host* menu located in the left pane.

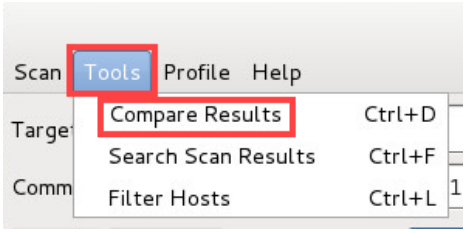


6. Once the *second host* is selected, click on the **Ports/Hosts** tab and view the specific opened ports for that host.

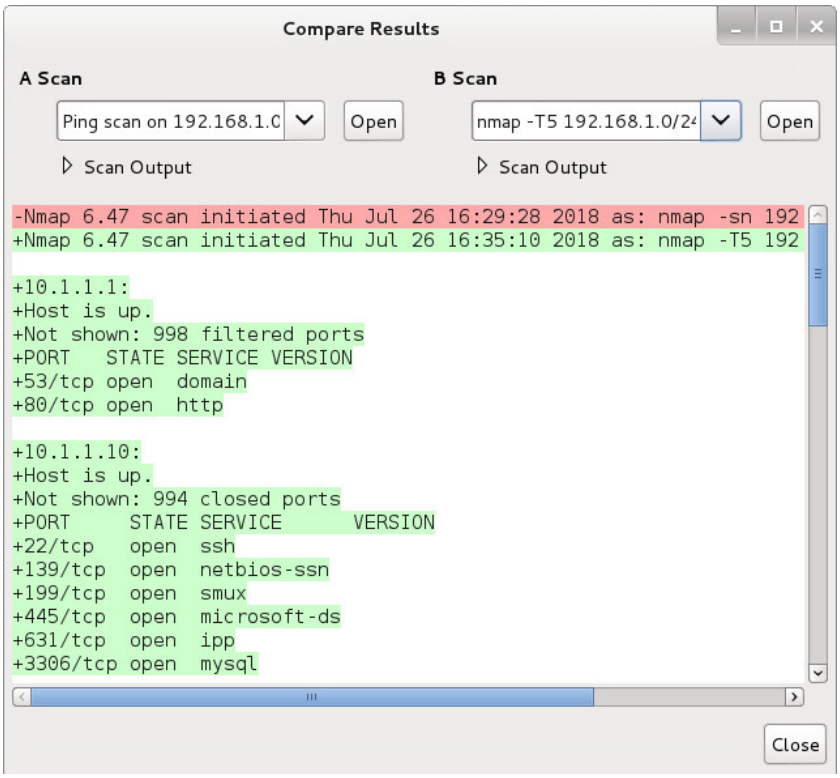


7. On the menu to the left, click the **Services** button. Notice that you may also filter by services when analyzing scan results.

8. When doing more in-depth network scans, you may compare results from different scans. To do so, select the **Tools** menu option and click on **Compare Results**.



9. A *Compare Results* window appears. Choose the **Ping scan** from the drop-down menu as the *A Scan*. Choose the **nmap –T5** scan from the drop-down menu as the *B Scan*.



When comparing scans, differences between the scans will be highlighted in red. The only difference between these two scans is the date/time, otherwise everything else is similar. This tool can be more useful when using different scan options against the same host/network.

10. Click the **Close** button.
11. Close the **Zenmap** application, if prompted, click **Close anyway**.
12. Leave the *Kali* window open to continue with the next task.

2 Scanning the Network Using OpenVAS

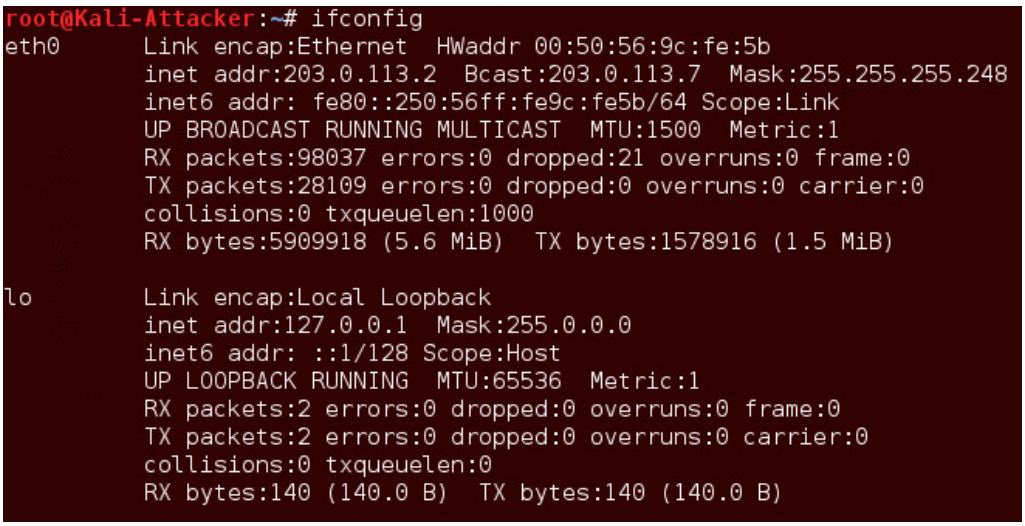
2.1 Scanning with OpenVAS

1. While on the *Kali* system, focus on the **terminal** window. Use the **ifconfig** command to bring the *loopback interface* to an **up** state. Enter the command below.

```
root@Kali-Attacker:~# ifconfig lo up
```

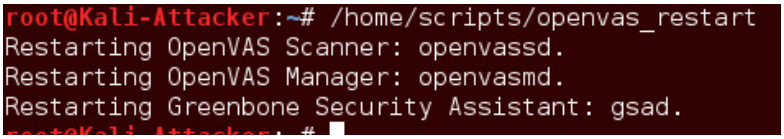
2. Verify that the loopback is now up and running.

```
root@Kali-Attacker:~# ifconfig
```



3. Initiate the **openvas_start** script to initialize the *OpenVAS Network Scanning* application.

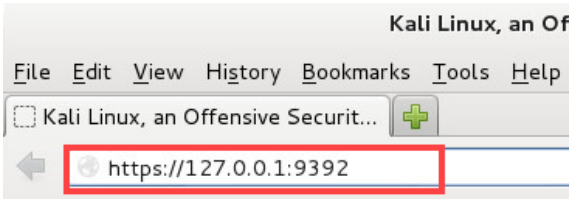
```
root@Kali-Attacker:~# /home/scripts/openvas_restart
```



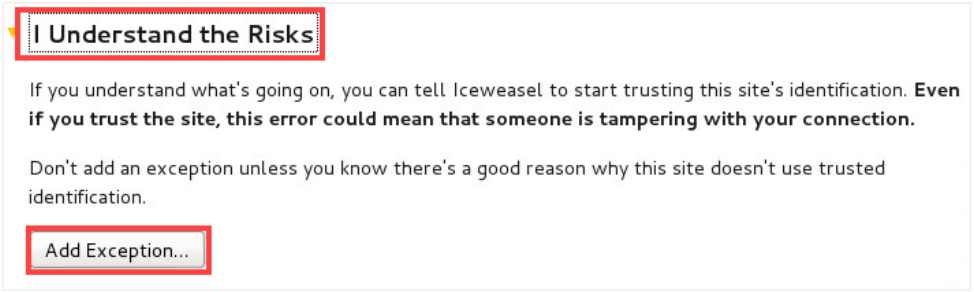
If an error appears from the script, you may ignore it and proceed as normal.

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

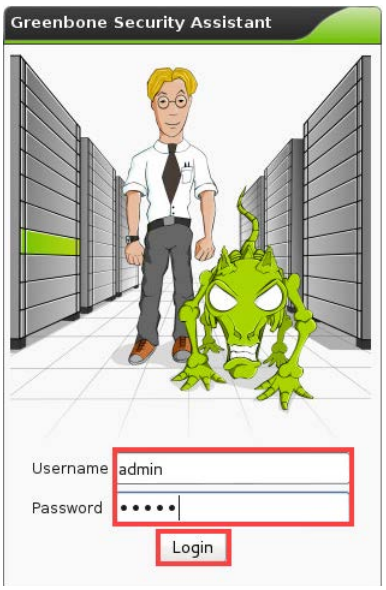
5. In the web browser window, type **`https://127.0.0.1:9392`**. in the address bar, followed by pressing the **Enter** key.



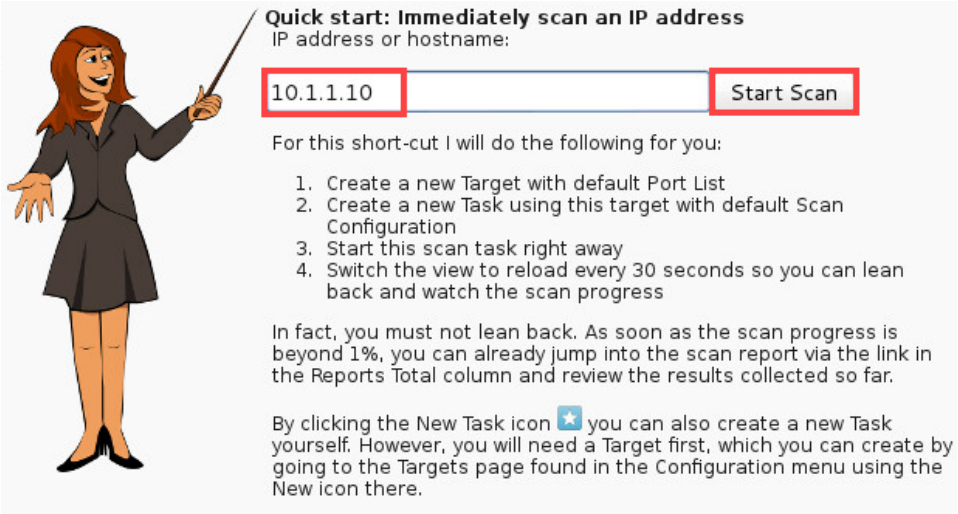
6. If presented with a connection is untrusted message, click **I Understand the Risks** followed by clicking on the **Add Exception** button. A pop-up window then appears, click **Confirm Security Exception** to continue.



7. At the login prompt, enter **`admin`** as the *username* and **`admin`** as the *password*. Click **Login**.



8. Upon logging in, you will be presented with the *homepage* for the *OpenVAS* management dashboard. Underneath *Quick start*, type the IP address **`10.1.1.10`** and click the **Start Scan** button to initialize a default quick scan.



9. Notice the page refreshes automatically and a new name "*Immediate scan of IP 10.1.1.10*" listed along with the *Status as Requested*.

10. Wait for the page to automatically refresh, and once it does, it will show a percentage bar underneath the *Status* column. If the page does not automatically refresh, click on the **Greenbone Security Assistant** logo in the upper-left corner.

Please Note

This scan will approximately take 4-5 minutes to complete. While it scans, you may proceed to the next step.

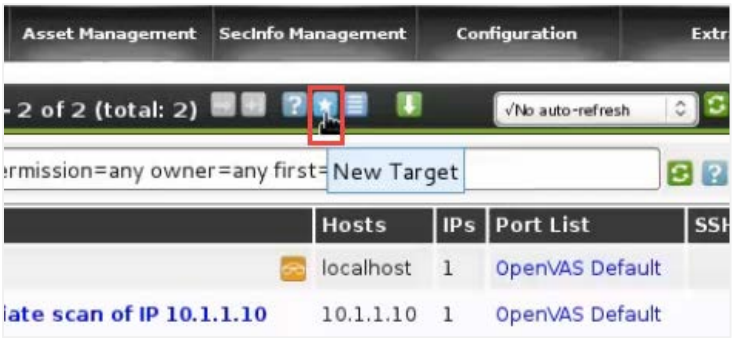
11. While the scan is running, navigate to the **SecInfo Management** menu option and click on **CVEs**.

12. Here you will find different *Common Vulnerabilities and Exposures (CVE)* loaded into the database. When initializing a *network scan*, the scan compares results to the stored *CVEs* and determines which of the known vulnerabilities are found. It is important to update *CVEs* consistently.
13. Leave the *Kali* window open to continue with the next task.

2.2 Create New Target

1. Click on the **Configuration** menu option and select **Targets**.

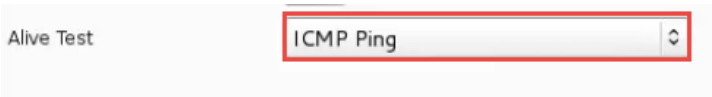
2. To add new known network targets such as systems that are being administered, click the **New Target** icon.



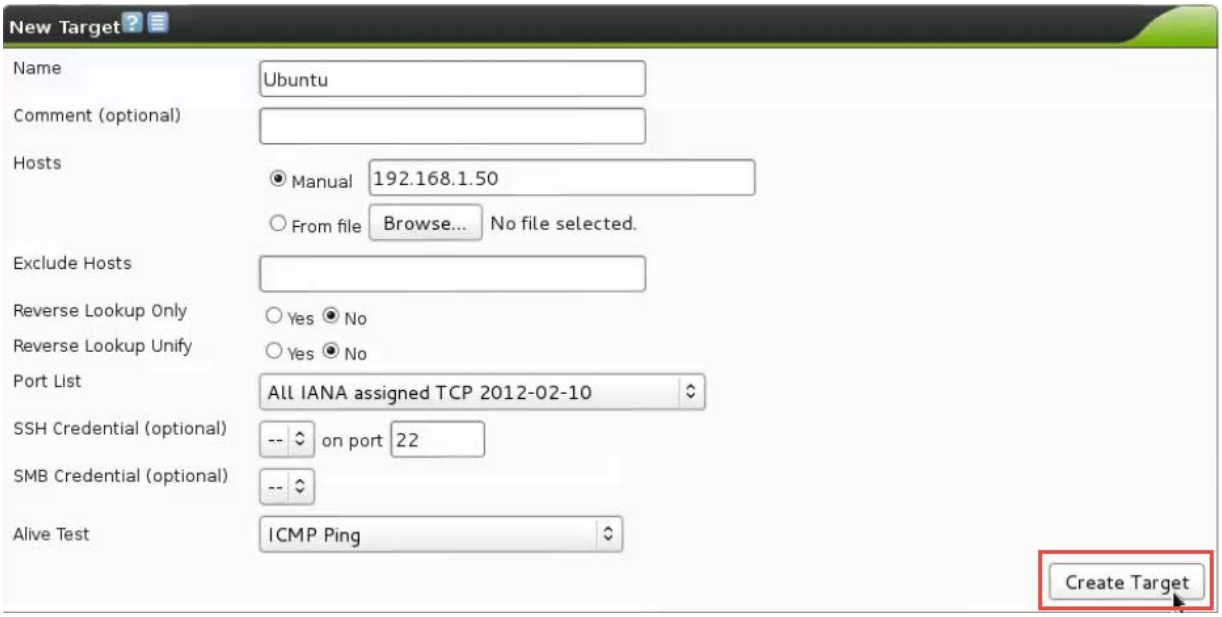
3. Type **Ubuntu** in the *Name* text field.

4. For *Hosts*, make sure it is set to **Manual** and type **192.168.1.50** as the hosts' identifier.

5. Select **ICMP Ping** for the *Alive Test*.



6. Click on the **Create Target** button.



7. Navigate to **Configuration > Targets** and verify that the new *Ubuntu* target has been successfully added.

Name	Hosts	IPs	Port List
Localhost	localhost	1	OpenVAS Default
Target for immediate scan of IP 10.1.1.10	10.1.1.10	1	OpenVAS Default
Ubuntu	192.168.1.50	1	All IANA assigned TCP 2012-02-10

(Applied filter: rows=10 permission=any owner=any first=1 sort=name)

8. Leave the *Kali* window open to continue with the next task.

2.3 Create New User

1. Navigate to **Administration > Users**.

2. Add a new user by clicking on the **New User** icon.

3. Once redirected to the *New User* page, fill in the necessary fields:
    a. *Login Name*: **`Analyst1`**
    b. *Password*: **`password`**
    c. *Roles*: **User**
    d. *Host Access*: **Deny all and allow**:
        i. Type **192.168.1.50** into the whitespace.
    e. Click **Create User**.

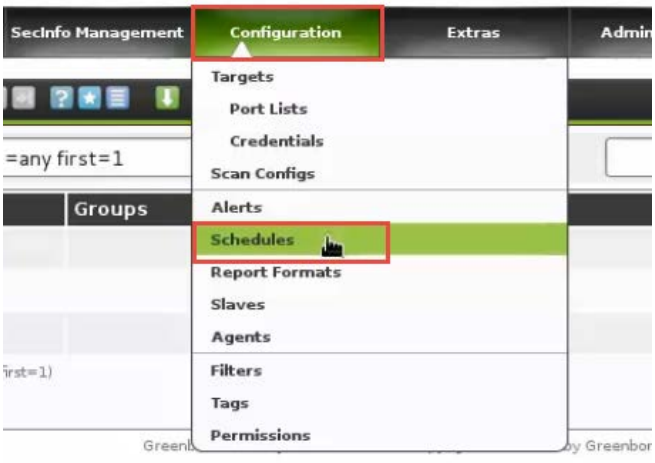
4. Navigate to **Administration > Users** and verify that the new *Analyst1* user has been successfully created.

Name	Roles	Groups
openvasadmin	Admin	
admin	Admin	
Analyst1	User	

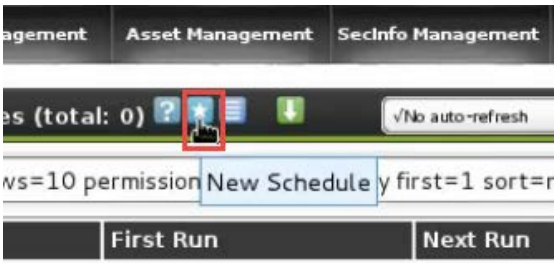
5. Leave the *Kali* window open to continue with the next task.

2.4 Create New Schedule

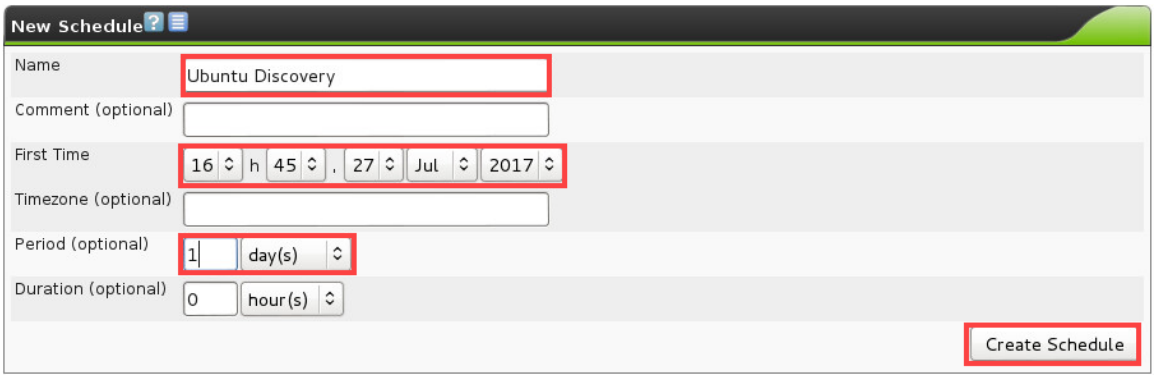
1. Navigate to **Configuration > Schedules**.



2. Create a new schedule by clicking on the **New Schedule** icon.



3. On the *New Schedule* page, fill in the necessary fields:
   a. *Name*: **Ubuntu Discovery**
   b. *First Time*: **Choose any desired time**
   c. *Period*: **1 day**
   d. Leave the rest to defaults.
   e. Click **Create Schedule**.



4. Verify that the daily schedule has been successfully configured by navigating to **Configuration > Schedules**.

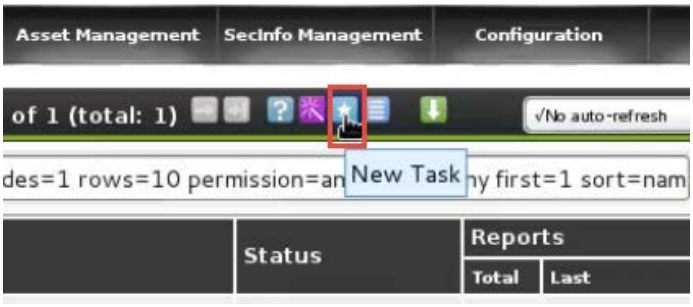
Name	First Run	Next Run	Period	Duration
Ubuntu Discovery	Thu Jul 27 16:45:00 2017 UTC	Sat Jul 28 16:45:00 2018 UTC	1 day	Entire Operation

5. Leave the *Kali* window open to continue with the next task.

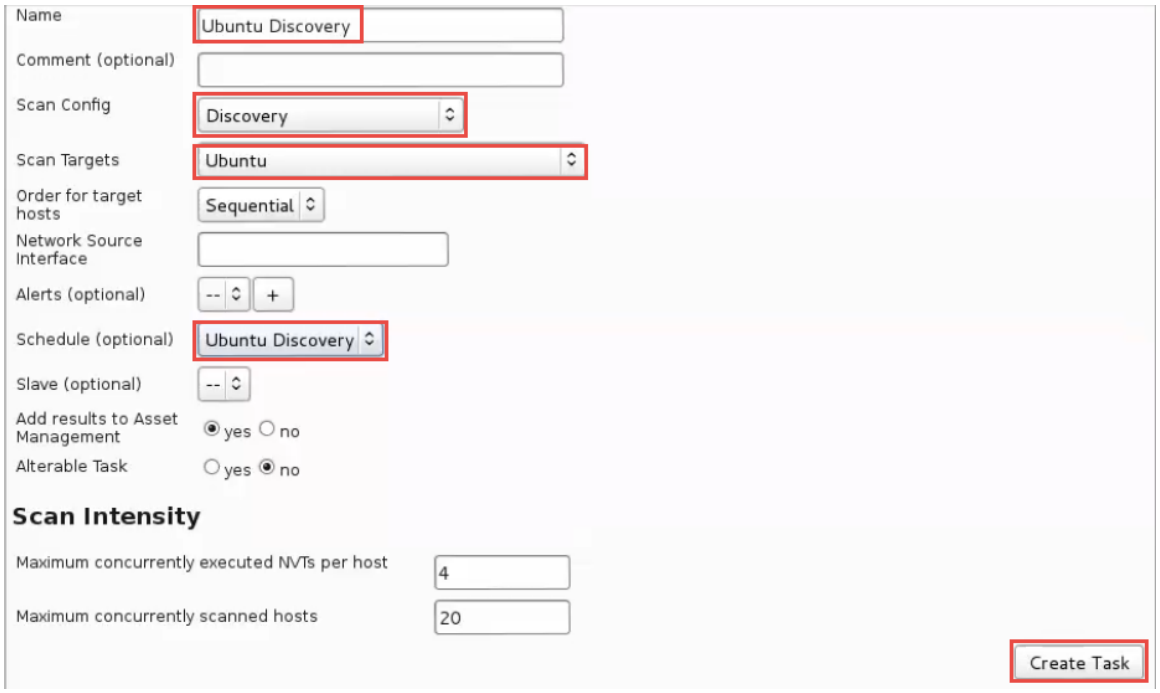
2.5 Schedule a New Task

1. Navigate to **Scan Management > Tasks**.

2. Click on the **New Task** icon.



3. Once redirected to the *New Task* page, fill in the necessary fields:
    a. *Name*: **Ubuntu Discovery**
    b. *Scan Config*: **Discovery**
    c. *Scan Targets*: **Ubuntu**
    d. *Schedule*: **Ubuntu Discovery**
    e. Leave the rest with defaults.
    f. Click **Create Task**.



4. Verify that the new daily task has been successfully created by navigating to **Scan Management > Tasks**.

Name	Status	Reports	
		Total	Last
Immediate scan of IP 10.1.1.10	Done	1 (1)	Jul 27 2018
Ubuntu Discovery	New		

5. Leave the *Kali* window open to continue with the next task.

2.6 Analyzing the Scan Report

1. While on the *Tasks* page, notice the name "*Immediate scan of IP 10.1.1.10*" should have a status of *Done*. If the scan is finished, click on the number "**1**" underneath the *Total Reports* column to view the scan report for *Immediate scan of IP 10.1.1.10*.

Name	Status	Reports		Severity
		Total	Last	
Immediate scan of IP 10.1.1.10	Done	1 (1)	Jul 27 2018	5.0 (Medium)
Ubuntu Discovery	New			

Please Note

If the scan is not finished, you may click on the status bar to view current scan results and proceed to *Step 4*. You may also choose to wait an additional 2-5 minutes for the scan to finish.

2. Once redirected to the *Reports* page, notice the *Scan Results* column. Click on the **date** given underneath the *Date* column to view the detailed scan report.

Date	Status	Task	Severity	Scan Results				
				High	Medium	Low	Log	False Pos.
Fri Jul 27 16:20:35 2018	Done	Immediate scan of IP 10.1.1.10	5.0 (Medium)	0	1	2	19	0

3. On the *Scan Reports* results page, analyze through the vulnerabilities found for host *10.1.1.10*.

Clicking on each vulnerability will display details about the vulnerability found along with associated *CVEs*, *CERTs*, and third-party resources explaining the vulnerability.

4. The lab is now complete; you may end the reservation.