Module:- SECURITY CONCEPT
(Sniffing and TCPdump)
Name:-Prithviraj Nikam

Lab Assignments:

Sniffing :-Sniffing is the process of monitoring and capturing all the packets passing through a given network using sniffing tools. It is a form of "tapping phone wires" and getting to know about the conversation. It is also called wiretapping applied to computer networks.

In other words, Sniffing allows you to see all sorts of traffic, both protected and unprotected. In the right conditions and with the right protocols in place, an attacking party may be able to gather information that can be used for further attacks or to cause other issues for the network or system owner.

What can be sniffed?

One can sniff the following sensitive information from a network −

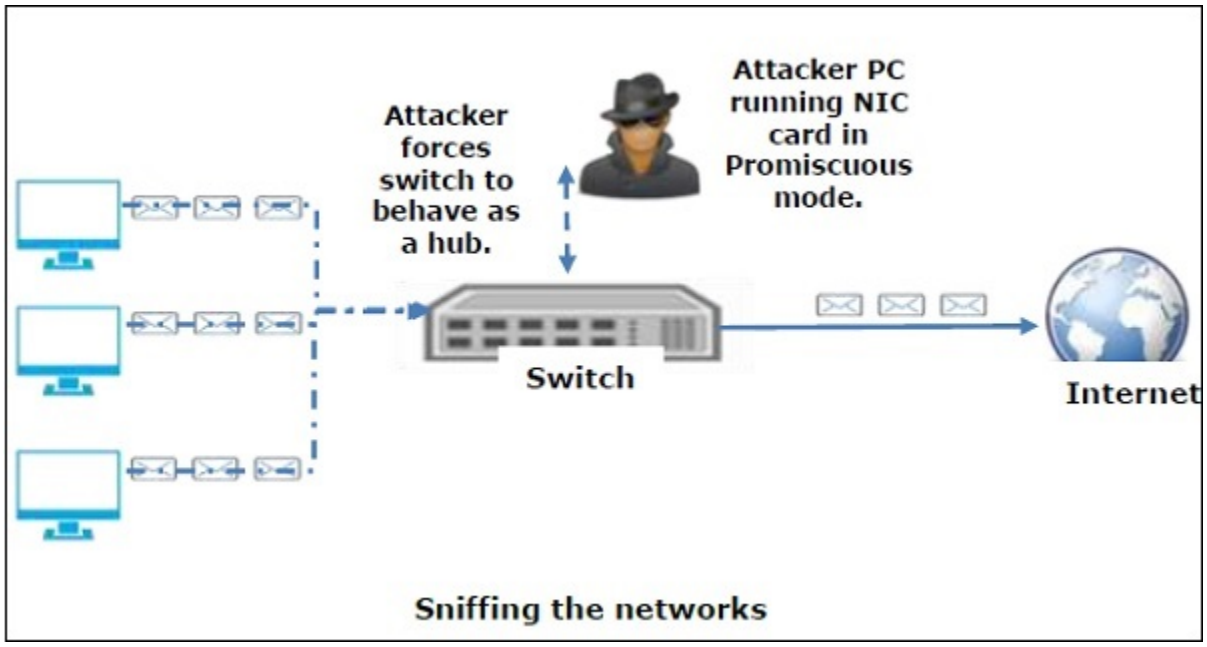
- Email traffic
- FTP passwords
- Web traffics
- Telnet passwords
- Router configuration
- Chat sessions
- DNS traffic

How it works

A sniffer normally turns the NIC of the system to the promiscuous mode so that it listens to all the data transmitted on its segment.

Promiscuous mode refers to the unique way of Ethernet hardware, in particular, network interface cards (NICs), that allows an NIC to receive all traffic on the network, even if it is not addressed to this NIC. By default, a NIC ignores all traffic that is not addressed to it, which is done by comparing the destination address of the Ethernet packet with the hardware address (a.k.a. MAC) of the device. While this makes perfect sense for networking, non-promiscuous mode makes it difficult

to use network monitoring and analysis software for diagnosing connectivity issues or traffic accounting.



Types of Sniffing

Sniffing can be either Active or Passive in nature.

Passive Sniffing

In passive sniffing, the traffic is locked but it is not altered in any way. Passive sniffing allows listening only. It works with Hub devices. On a hub device, the traffic is sent to all the ports. In a network that uses hubs to connect systems, all hosts on the network can see the traffic. Therefore, an attacker can easily capture traffic going through.

The good news is that hubs are almost obsolete nowadays. Most modern networks use switches. Hence, passive sniffing is no more effective.

Active Sniffing

In active sniffing, the traffic is not only locked and monitored, but it may also be altered in some way as determined by the attack. Active sniffing is used to sniff a switch-based network. It involves injecting address resolution packets (ARP) into a target network to flood on the switch content addressable memory (CAM) table. CAM keeps track of which host is connected to which port.

Following are the Active Sniffing Techniques −

- MAC Flooding
- DHCP Attacks
- DNS Poisoning
- Spoofing Attacks
- ARP Poisoning

Protocols which are affected

Protocols such as the tried and true TCP/IP were never designed with security in mind and therefore do not offer much resistance to potential intruders. Several rules lend themselves to easy sniffing −

- **HTTP** − It is used to send information in the clear text without any encryption and thus a real target.
- **SMTP (Simple Mail Transfer Protocol)** − SMTP is basically utilized in the transfer of emails. This protocol is efficient, but it does not include any protection against sniffing.
- **NNTP (Network News Transfer Protocol)**− It is used for all types of communications, but its main drawback is that data and even passwords are sent over the network as clear text.
- **POP (Post Office Protocol)** − POP is strictly used to receive emails from the servers. This protocol does not include protection against sniffing because it can be trapped.
- **FTP (File Transfer Protocol)** − FTP is used to send and receive files, but it does not offer any security features. All the data is sent as clear text that can be easily sniffed.
- **IMAP (Internet Message Access Protocol)** − IMAP is the same as SMTP in its functions, but it is highly vulnerable to sniffing.
- **Telnet** − Telnet sends everything (usernames, passwords, keystrokes) over the network as clear text and hence, it can be easily sniffed.

Sniffing Tool

- **BetterCAP** − BetterCAP is a powerful, flexible and portable tool created to perform various types of MITM attacks against a network, manipulate HTTP, HTTPS and TCP traffic in real-time, sniff for credentials, and much more.
- **Ettercap** − 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.

- **Wireshark** − It is one of the most widely known and used packet sniffers. It offers a tremendous number of features designed to assist in the dissection and analysis of traffic.
- **Tcpdump** − It is a well-known command-line packet analyzer. It provides the ability to intercept and observe TCP/IP and other packets during transmission over the network. Available at www.tcpdump.org.
- **WinDump** − A Windows port of the popular Linux packet sniffer tcpdump, which is a command-line tool that is perfect for displaying header information.
- **OmniPeek** − Manufactured by WildPackets, OmniPeek is a commercial product that is the evolution of the product EtherPeek.
- **Dsniff** − A suite of tools designed to perform sniffing with different protocols with the intent of intercepting and revealing passwords. Dsniff is designed for Unix and Linux platforms and does not have a full equivalent on the Windows platform.
- **EtherApe** − It is a Linux/Unix tool designed to display graphically a system's incoming and outgoing connections.
- **MSN Sniffer** − It is a sniffing utility specifically designed for sniffing traffic generated by the MSN Messenger application.
- **NetWitness NextGen** − It includes a hardware-based sniffer, along with other features, designed to monitor and analyze all traffic on a network. This tool is used by the FBI and other law enforcement agencies.

TCPDUMP:-It is a well-known command-line packet analyzer.

Step-1:- List of Network Interfaces
 # tcpdump -D

```
File  Actions  Edit  View  Help
┌──(prithvi㉿kali)-[~]
└─$ tcpdump -D
1.eth0 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
7.dbus-system (D-Bus system bus) [none]
8.dbus-session (D-Bus session bus) [none]

┌──(prithvi㉿kali)-[~]
└─$
```

Step-2:- Capture Traffic on single Interface
#sudo tcpdump -i eth0

```
┌──(prithvi㉿kali)-[~]
└─$ sudo tcpdump -i eth0
[sudo] password for prithvi:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
17:44:37.914604 IP 192.168.3.226.56628 > 239.255.255.250.1900: UDP, length 174
17:44:37.983361 IP 192.168.3.88.35917 > stuns.blr1.cdac.in.domain: 46256+ PTR? 250.255.255.23
17:44:38.342606 IP6 fe80::a00:27ff:fe06:b785 > fe80::2ae:9dff:fe00:ae9d: ICMP6, neighbor soli
17:44:38.917405 IP 192.168.3.226.56628 > 239.255.255.250.1900: UDP, length 174
17:44:39.107599 ARP, Request who-has 192.168.3.88 tell 172.31.158.14, length 46
17:44:39.107624 ARP, Reply 192.168.3.88 is-at 08:00:27:06:b7:85 (oui Unknown), length 28
17:44:39.648787 IP stuns.blr1.cdac.in.domain > 192.168.3.88.35917: 46256 NXDomain 0/1/0 (103)
17:44:39.649025 IP 192.168.3.88.41714 > stuns.blr1.cdac.in.domain: 39858+ PTR? 226.3.168.192.
17:44:39.658411 IP stuns.blr1.cdac.in.domain > 192.168.3.88.41714: 39858 NXDomain* 0/1/0 (99)
17:44:39.658608 IP 192.168.3.88.56578 > stuns.blr1.cdac.in.domain: 34947+ PTR? 3.1.168.192.in
17:44:39.661712 IP stuns.blr1.cdac.in.domain > 192.168.3.88.56578: 34947* 1/1/1 PTR stuns.blr
17:44:39.661937 IP 192.168.3.88.52122 > stuns.blr1.cdac.in.domain: 12091+ PTR? 88.3.168.192.i
17:44:39.677059 IP stuns.blr1.cdac.in.domain > 192.168.3.88.52122: 12091 NXDomain* 0/1/0 (98)
17:44:39.679154 IP 192.168.3.88.36778 > stuns.blr1.cdac.in.domain: 10410+ PTR? d.9.e.a.0.0.e.
17:44:39.687305 IP stuns.blr1.cdac.in.domain > 192.168.3.88.36778: 10410 NXDomain* 0/1/0 (139
17:44:39.687509 IP 192.168.3.88.57895 > stuns.blr1.cdac.in.domain: 20047+ PTR? 5.8.7.b.6.0.e.
17:44:39.691139 IP stuns.blr1.cdac.in.domain > 192.168.3.88.57895: 20047 NXDomain* 0/1/0 (139
```

Step-3:- Capture N number of Packets
 # Sudo tcpdump -i eth0 -c 10
Number of Packets

```
┌──(prithvi㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -c 10
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
17:46:40.955994 IP 192.168.3.226.59607 > 239.255.255.250.1900: UDP, length 174
17:46:41.025962 IP 192.168.3.88.33525 > stuns.blr1.cdac.in.domain: 42734+ PTR? 250.255.255.239.in-addr.arpa. (46)
17:46:41.037192 IP stuns.blr1.cdac.in.domain > 192.168.3.88.33525: 42734 NXDomain 0/1/0 (103)
17:46:41.037433 IP 192.168.3.88.47926 > stuns.blr1.cdac.in.domain: 51798+ PTR? 226.3.168.192.in-addr.arpa. (44)
17:46:41.059601 IP stuns.blr1.cdac.in.domain > 192.168.3.88.47926: 51798 NXDomain* 0/1/0 (99)
17:46:41.129897 IP 192.168.3.88.55336 > stuns.blr1.cdac.in.domain: 54411+ PTR? 3.1.168.192.in-addr.arpa. (42)
17:46:41.198498 IP stuns.blr1.cdac.in.domain > 192.168.3.88.55336: 54411* 1/1/1 PTR stuns.blr1.cdac.in. (104)
17:46:41.198679 IP 192.168.3.88.55984 > stuns.blr1.cdac.in.domain: 12349+ PTR? 88.3.168.192.in-addr.arpa. (43)
17:46:41.221780 IP6 fe80::a00:27ff:fe06:b785 > fe80::2ae:9dff:fe00:ae9d: ICMP6, neighbor solicitation, who has fe80::2ae:9dff:fe00:ae9d, length 32
17:46:41.243225 IP stuns.blr1.cdac.in.domain > 192.168.3.88.55984: 12349 NXDomain* 0/1/0 (98)
10 packets captured
14 packets received by filter
0 packets dropped by kernel
```

Step-4:- To save a Packet capture file .

 # sudo tcpdump -i eth0 -c 10 -w abc.pcap

Number of Packets — **file_name.pcap**

```
prithvi@kali: ~
File  Actions  Edit  View  Help

┌──(prithvi㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -c 10 -w abc.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
10 packets captured
12 packets received by filter
0 packets dropped by kernel

┌──(prithvi㉿kali)-[~]
└─$
```

Step-5:- Read the pcap file

 # Sudo tcpdump -r abc.pcap

file_name.pcap

```
┌──(prithvi㉿kali)-[~]
└─$ sudo tcpdump -r abc.pcap
reading from file abc.pcap, link-type EN10MB (Ethernet), snapshot length 262144
17:55:35.994460 IP6 fe80::2ae:9dff:fe00:ae9d > fe80::a00:27ff:fe06:b785: ICMP6, neighbor solicitation, who has fe80::a00:27ff:fe06:b785, length 32
17:55:35.994506 IP6 fe80::a00:27ff:fe06:b785 > fe80::2ae:9dff:fe00:ae9d: ICMP6, neighbor advertisement, tgt is fe80::a00:27ff:fe06:b785, length 24
17:55:38.176751 IP 192.168.3.104.57367 > 239.255.255.250.1900: UDP, length 175
17:55:38.892945 ARP, Request who-has 192.168.3.88 tell 172.31.158.14, length 46
17:55:38.892970 ARP, Reply 192.168.3.88 is-at 08:00:27:06:b7:85 (oui Unknown), length 28
17:55:39.181789 IP 192.168.3.104.57367 > 239.255.255.250.1900: UDP, length 175
17:55:40.189736 IP 192.168.3.104.57367 > 239.255.255.250.1900: UDP, length 175
17:55:40.302939 IP6 fe80::1c0a:7ad7:1de1:de63.dhcpv6-client > ff02::1:2.dhcpv6-server: dhcp6 solicit
17:55:41.106072 IP6 fe80::2ae:9dff:fe00:ae9d > fe80::a00:27ff:fe06:b785: ICMP6, neighbor solicitation, who has fe80::a00:27ff:fe06:b785, length 32
17:55:41.106111 IP6 fe80::a00:27ff:fe06:b785 > fe80::2ae:9dff:fe00:ae9d: ICMP6, neighbor advertisement, tgt is fe80::a00:27ff:fe06:b785, length 24

┌──(prithvi㉿kali)-[~]
└─$
```

Step-6:- show full content of packet

 # sudo tcpdump -i eth0 -s0

```
┌──(prithvi㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -s0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
17:57:09.506715 IP 192.168.3.242.54897 > 239.255.255.250.1900: UDP, length 175
17:57:09.573988 IP 192.168.3.88.50025 > stuns.blr1.cdac.in.domain: 37206+ PTR? 250.255.255.239.in-addr.arpa. (46)
17:57:09.576298 IP stuns.blr1.cdac.in.domain > 192.168.3.88.50025: 37206 NXDomain 0/1/0 (103)
17:57:09.576490 IP 192.168.3.88.44967 > stuns.blr1.cdac.in.domain: 45479+ PTR? 242.3.168.192.in-addr.arpa. (44)
17:57:09.578804 IP stuns.blr1.cdac.in.domain > 192.168.3.88.44967: 45479 NXDomain* 0/1/0 (99)
17:57:09.677897 IP 192.168.3.88.55838 > stuns.blr1.cdac.in.domain: 15333+ PTR? 3.1.168.192.in-addr.arpa. (42)
17:57:09.681301 IP stuns.blr1.cdac.in.domain > 192.168.3.88.55838: 15333* 1/1/1 PTR stuns.blr1.cdac.in. (104)
17:57:09.681482 IP 192.168.3.88.44954 > stuns.blr1.cdac.in.domain: 37852+ PTR? 88.3.168.192.in-addr.arpa. (43)
17:57:09.683641 IP stuns.blr1.cdac.in.domain > 192.168.3.88.44954: 37852 NXDomain* 0/1/0 (98)
17:57:10.524876 IP 192.168.3.242.54897 > 239.255.255.250.1900: UDP, length 175
17:57:10.872856 ARP, Request who-has 192.168.3.88 tell 172.31.158.14, length 46
17:57:10.872883 ARP, Reply 192.168.3.88 is-at 08:00:27:06:b7:85 (oui Unknown), length 28
17:57:10.925914 IP 192.168.3.88.54869 > stuns.blr1.cdac.in.domain: 18460+ PTR? 14.158.31.172.in-addr.arpa. (44)
17:57:10.928437 IP stuns.blr1.cdac.in.domain > 192.168.3.88.54869: 18460 NXDomain* 0/1/0 (98)
^C
14 packets captured
14 packets received by filter
0 packets dropped by kernel
```

 # sudo tcpdump -i eth0 -s0 -w abc.pcap

file_name.pcap

```
┌──(prithvi㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -s0 -w abc.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C9 packets captured
9 packets received by filter
0 packets dropped by kernel

┌──(prithvi㉿kali)-[~]
└─$
```

Step-7:- Capture packet more than

 # sudo tcpdump -i eth0 greater 20

Packet size

```
┌──(prithvi㉿kali)-[~]
└─$ sudo tcpdump -i eth0 greater 20
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:01:26.370416 ARP, Request who-has 192.168.3.88 tell 172.31.158.14, length 46
18:01:26.370416 ARP, Reply 192.168.3.88 is-at 08:00:27:06:b7:85 (oui Unknown), length 28
18:01:26.389949 IP 192.168.3.88.54343 > stuns.blr1.cdac.in.domain: 15714+ PTR? 88.3.168.192.in-addr.arpa. (43)
18:01:26.392966 IP stuns.blr1.cdac.in.domain > 192.168.3.88.54343: 15714 NXDomain* 0/1/0 (98)
18:01:26.393143 IP 192.168.3.88.37807 > stuns.blr1.cdac.in.domain: 56659+ PTR? 14.158.31.172.in-addr.arpa. (44)
18:01:26.395877 IP stuns.blr1.cdac.in.domain > 192.168.3.88.37807: 56659 NXDomain* 0/1/0 (98)
18:01:26.493927 IP 192.168.3.88.47095 > stuns.blr1.cdac.in.domain: 2345+ PTR? 3.1.168.192.in-addr.arpa. (42)
18:01:26.496041 IP stuns.blr1.cdac.in.domain > 192.168.3.88.47095: 2345* 1/1/1 PTR stuns.blr1.cdac.in. (104)
18:01:28.517738 IP6 fe80::a00:27ff:fe06:b785 > fe80::2ae:9dff:fe00:ae9d: ICMP6, neighbor solicitation, who has fe80::2ae:9dff:fe00:ae9d, length 32
18:01:28.573995 IP 192.168.3.88.40610 > stuns.blr1.cdac.in.domain: 53077+ PTR? d.9.e.a.0.0.e.f.f.f.d.9.e.a.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. (90)
18:01:28.585275 IP6 fe80::2ae:9dff:fe00:ae9d > fe80::a00:27ff:fe06:b785: ICMP6, neighbor solicitation, who has fe80::a00:27ff:fe06:b785, length 32
18:01:28.585309 IP6 fe80::a00:27ff:fe06:b785 > fe80::2ae:9dff:fe00:ae9d: ICMP6, neighbor advertisement, tgt is fe80::a00:27ff:fe06:b785, length 24
18:01:28.595008 IP stuns.blr1.cdac.in.domain > 192.168.3.88.40610: 53077 NXDomain* 0/1/0 (139)
18:01:28.595191 IP 192.168.3.88.46668 > stuns.blr1.cdac.in.domain: 46505+ PTR? 5.8.7.b.6.0.e.f.f.f.7.2.0.0.a.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. (90)
18:01:28.597048 IP stuns.blr1.cdac.in.domain > 192.168.3.88.46668: 46505 NXDomain* 0/1/0 (139)
18:01:29.541640 IP6 fe80::a00:27ff:fe06:b785 > fe80::2ae:9dff:fe00:ae9d: ICMP6, neighbor solicitation, who has fe80::2ae:9dff:fe00:ae9d, length 32
^C
16 packets captured
16 packets received by filter
0 packets dropped by kernel

┌──(prithvi㉿kali)-[~]
└─$
```

Step-8:- Capture packet less than

 # sudo tcpdump -i eth0 less 100

Packet size

```
┌──(prithvi㉿kali)-[~]
└─$ sudo tcpdump -i eth0 less 100
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:02:12.358014 ARP, Request who-has 192.168.3.88 tell 172.31.158.14, length 46
18:02:12.358038 ARP, Reply 192.168.3.88 is-at 08:00:27:06:b7:85 (oui Unknown), length 28
18:02:12.401946 IP 192.168.3.88.44367 > stuns.blr1.cdac.in.domain: 46726+ PTR? 88.3.168.192.in-ad
18:02:12.404399 IP 192.168.3.88.41189 > stuns.blr1.cdac.in.domain: 57814+ PTR? 14.158.31.172.in-a
18:02:12.505926 IP 192.168.3.88.39238 > stuns.blr1.cdac.in.domain: 40598+ PTR? 3.1.168.192.in-add
18:02:14.573975 IP6 fe80::2ae:9dff:fe00:ae9d > fe80::a00:27ff:fe06:b785: ICMP6, neighbor solicita
18:02:14.574018 IP6 fe80::a00:27ff:fe06:b785 > fe80::2ae:9dff:fe00:ae9d: ICMP6, neighbor advertis
18:02:17.413740 ARP, Request who-has 192.168.3.1 tell 192.168.3.88, length 28
18:02:17.460687 ARP, Reply 192.168.3.1 is-at ec:9b:8b:02:24:38 (oui Unknown), length 46
18:02:17.471383 ARP, Request who-has 192.168.3.88 tell 172.31.158.14, length 46
18:02:17.471418 ARP, Reply 192.168.3.88 is-at 08:00:27:06:b7:85 (oui Unknown), length 28
18:02:17.497770 IP 192.168.3.88.48710 > stuns.blr1.cdac.in.domain: 50875+ PTR? 1.3.168.192.in-add
18:02:17.872579 IP 54.191.251.76.https > 192.168.3.88.55590: Flags [P.], seq 582275590:582275621,
44724], length 31
18:02:17.913916 IP 192.168.3.88.41557 > stuns.blr1.cdac.in.domain: 52607+ PTR? 76.251.191.54.in-a
18:02:18.124844 IP 54.191.251.76.https > 192.168.3.88.55590: Flags [.], ack 36, win 119, options
18:02:18.461837 IP stuns.blr1.cdac.in.domain > 192.168.3.88.41557: 52607 ServFail 0/0/0 (44)
18:02:18.462111 IP 192.168.3.88.37054 > stuns.blr1.cdac.in.domain: 52607+ PTR? 76.251.191.54.in-a
18:02:18.463827 IP stuns.blr1.cdac.in.domain > 192.168.3.88.37054: 52607 ServFail 0/0/0 (44)
^C
18 packets captured
18 packets received by filter
0 packets dropped by kernel
```

Step-9:- To Capture packet based on the source

 # sudo tcpdump src 192.168.3.88

System IP

```
┌──(prithvi㉿kali)-[~]
└─$ sudo tcpdump src 192.168.3.88
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:30:03.175759 ARP, Reply 192.168.3.88 is-at 08:00:27:06:b7:85 (oui Unknown), length 28
18:30:03.261970 IP 192.168.3.88.57693 > stuns.blr1.cdac.in.domain: 28945+ PTR? 88.3.168.192.in-addr.arpa. (43)
18:30:03.366997 IP 192.168.3.88.47170 > stuns.blr1.cdac.in.domain: 21109+ PTR? 3.1.168.192.in-addr.arpa. (42)
18:30:08.287238 ARP, Reply 192.168.3.88 is-at 08:00:27:06:b7:85 (oui Unknown), length 28
18:30:08.325727 ARP, Request who-has 192.168.3.1 tell 192.168.3.88, length 28
18:30:08.357891 IP 192.168.3.88.40974 > stuns.blr1.cdac.in.domain: 31347+ PTR? 1.3.168.192.in-addr.arpa. (42)
^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel

┌──(prithvi㉿kali)-[~]
└─$
```

Step-10:- You want to see packet based on the port

#sudo tcpdump -i eth0 port 53

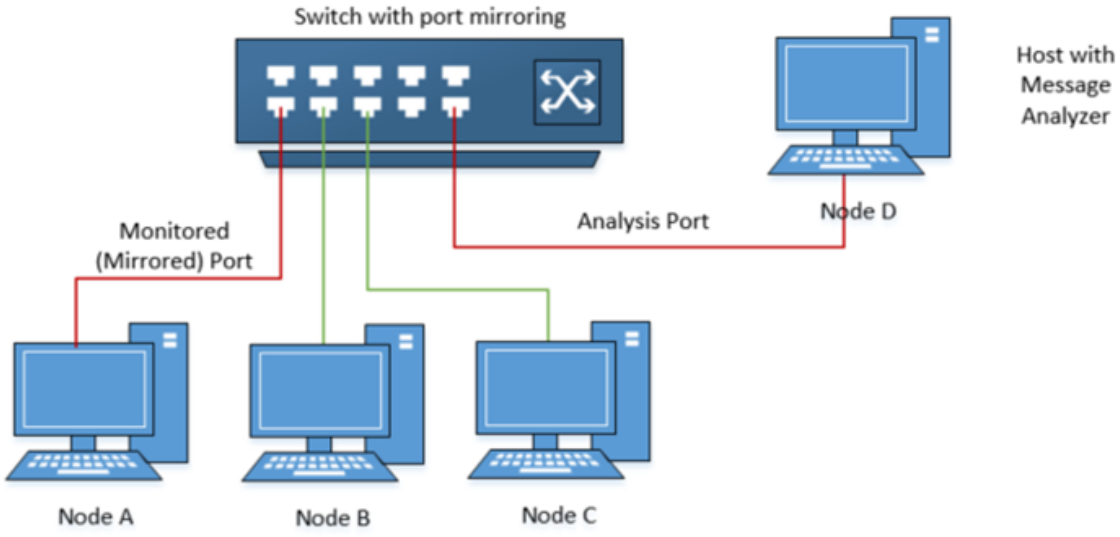
Port no

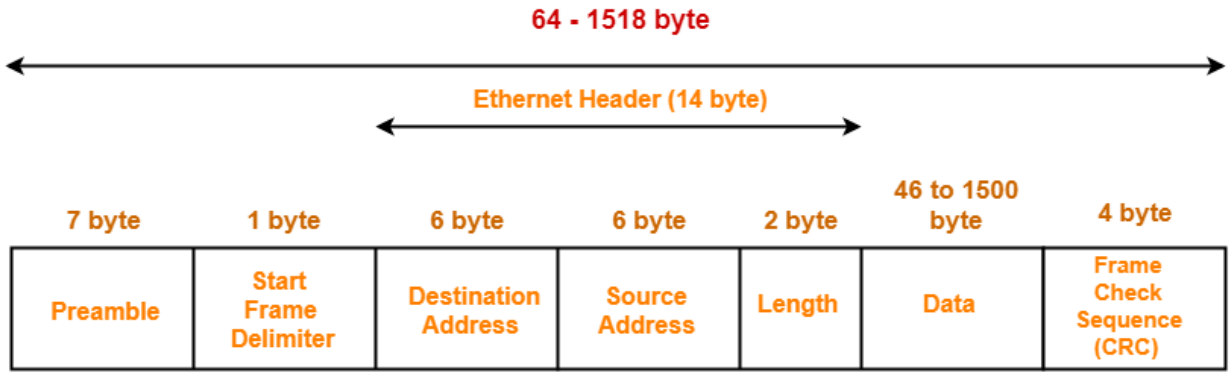
```
┌──(prithvi㉿kali)-[~]
└─$ sudo tcpdump -i eth0 port 53
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel

┌──(prithvi㉿kali)-[~]
└─$
```

Port mirroring and monitoring overview

- Port mirroring is a method of monitoring network traffic that forwards a copy of each incoming or outgoing packet from one port on a network switch to another port where the packet can be analyzed.
- Port mirroring can be used as a diagnostic tool or debugging feature, especially for preventing attacks. Port mirroring can be managed locally or remotely.
- You can configure port mirroring, by assigning a port (known as the Monitor port), from which the packets are copied and sent to a destination port (known as the Mirror port).
- All packets received on the Monitor port or issued from it, are forwarded to the second port. You next attach a protocol analyzer on the mirror port to monitor each segment separately.
- The analyzer captures and evaluates the data without affecting the client on the original port.
- The mirror port may be a port on the same switch with an attached **RMON(Remote Network Monitoring)** probe, a port on a different switch in the same hub, or the switch processor.





IEEE 802.3 Ethernet Frame Format