Sniffers

Chapter 9

What Is Sniffing?

Sniffers are a broad category that encompasses any utility that has the ability to perform a packet-capturing function.

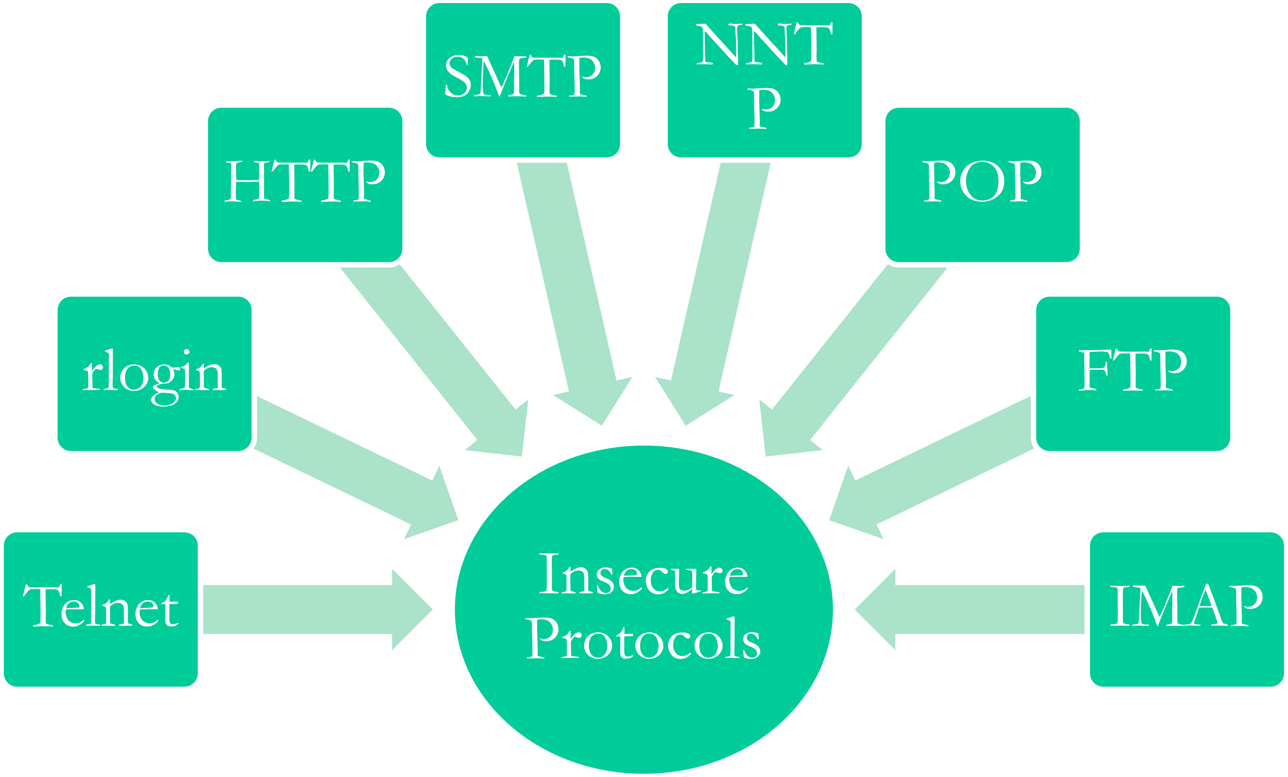
- **Is the act of viewing information as it flows over the network**
- **Can be performed with hardware or software**
- **Preys on vulnerable networks and protocols**
  - Passwords (from email, the Web, SMB, FTP, SQL, or Telnet)
  - Email text

Law Enforcement and Sniffing

Lawful interception (LI) is defined as legally accessing communications and network data such as telephone calls or email messages.

- Lawful interception is legally sanctioned access to network data
- Must have authority in pursuit of evidence or analysis
- Regulated by the law
- Sometimes called wiretapping

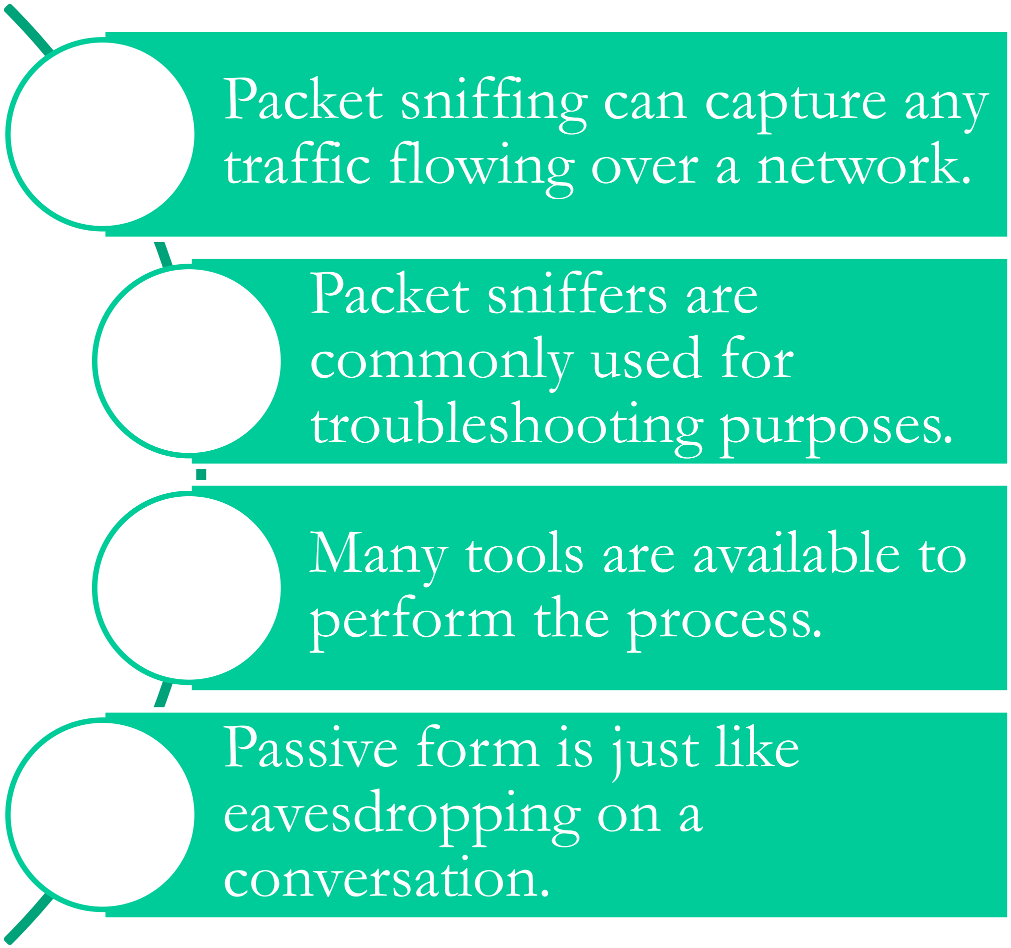
Vulnerable Protocols



How successful you are at the sniffing process depends on the relative and inherent insecurity of certain network protocols.

A Quick Overview

Packet sniffing, or packet analysis, is the process of capturing any data passed over the local network and looking for any information that may be useful.



Packet sniffing can capture any traffic flowing over a network.

Packet sniffers are commonly used for troubleshooting purposes.

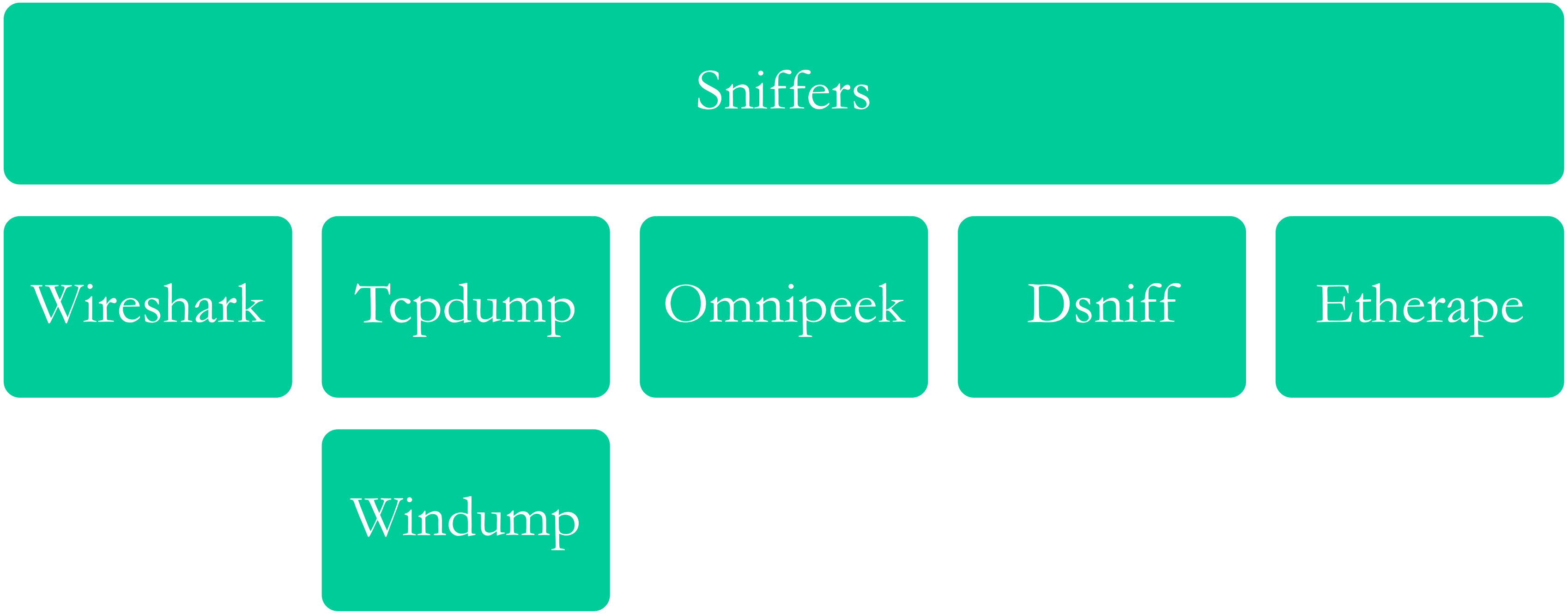
Many tools are available to perform the process.

Passive form is just like eavesdropping on a conversation.

What’s Required to Sniff?

- Hardware in the form of network adapters
- Drive program or the core sniffing program
- Buffer to temporarily store the results of a sniff
- Packet analysis capability to interpret results

A Selection of Sniffing Tools



Sniffers

Wireshark

Tcpdump

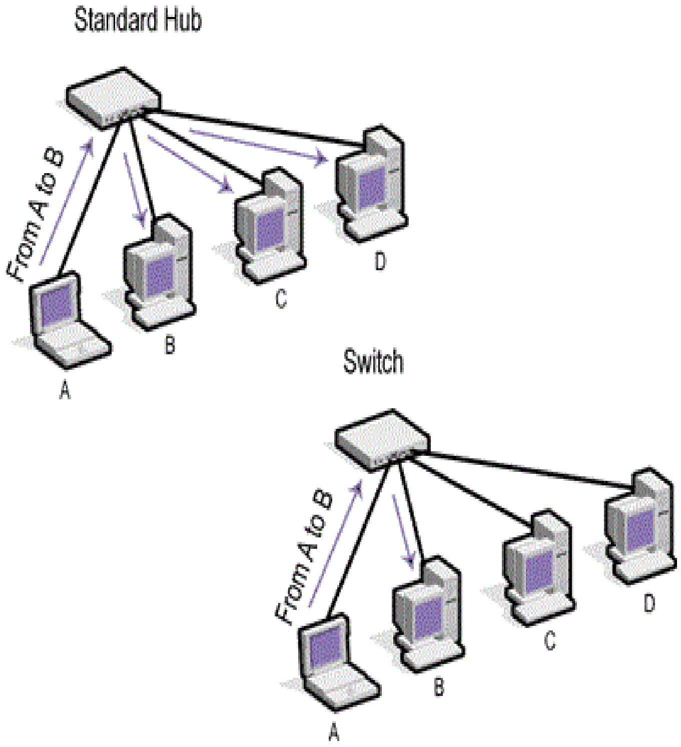
Omnipeek

Dsniff

Etherape

Windump

Types of Sniffing



Passive Sniffing

- Sniffing when a hub is present
- Restricted to a network segment
- Tends to be stealthier

Active Sniffing

- Sniffing when a switch is present
- Attempts to bypass switch
- Less stealthy

What Are Hubs?

- Central connection point for networks
- Broadcast traffic received out through every port
- Perform little to no filtering of traffic
- Slower and cheaper than switches
- Not common in modern networks

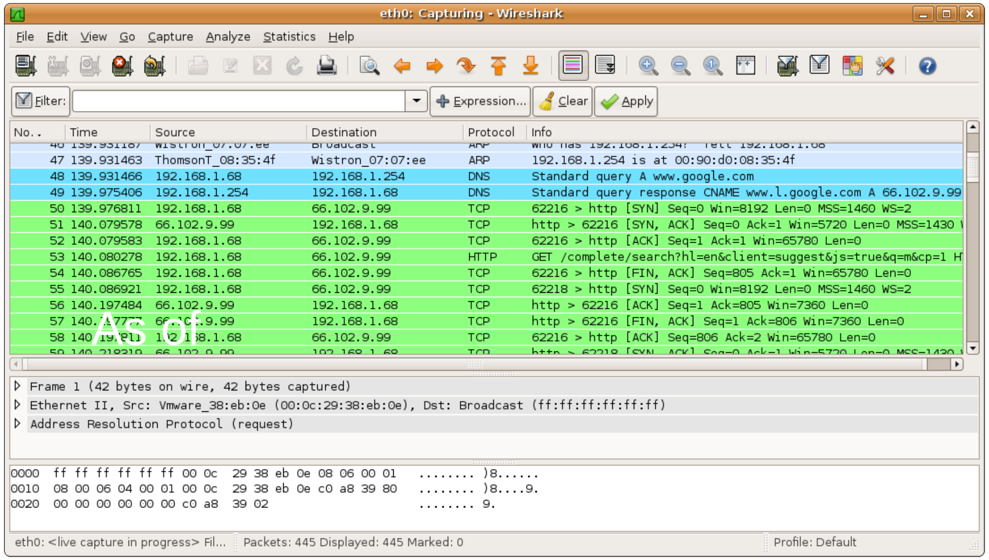
Network Switches

- **Switches**
  - Perform examination of each packet
  - Look at source and destination of each packet
  - Use information to direct traffic
  - Separate network into collision domains
  - Isolate network nodes from one another

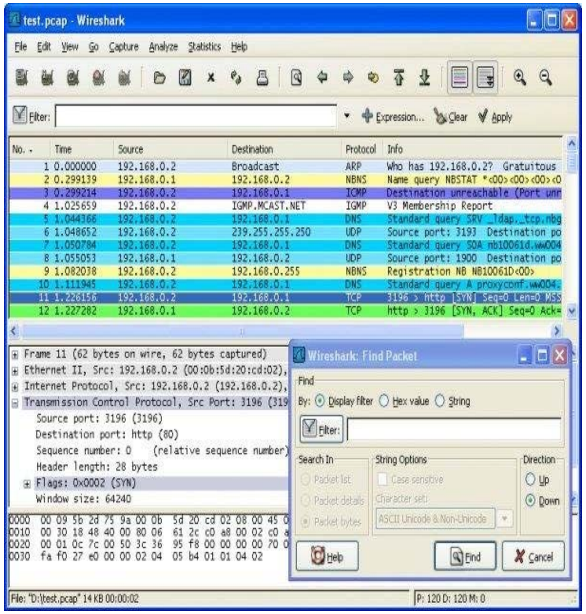
When a packet is received by the switch, the destination and source addresses and compares them to a table of network segments and addresses.

Wireshark

- As of this writing, Wireshark reigns supreme as perhaps the best sniffer on the market.
- Wireshark has been around for quite a while, and it has proven its worth time and time again.
- Wireshark is natively available on Windows, Mac OS X, and Linux.



```
No. .  Time        Source             Destination        Protocol  Info
    47 139.931463  ThomsonT_08:35:4f  Wistron_07:07:ee   ARP       192.168.1.254 is at 00:90:d0:08:35:4f
    48 139.931466  192.168.1.68       192.168.1.254      DNS       Standard query A www.google.com
    49 139.975406  192.168.1.254      192.168.1.68       DNS       Standard query response CNAME www.l.google.com A 66.102.9.99
    50 139.976811  192.168.1.68       66.102.9.99        TCP       62216 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
    51 140.079578  66.102.9.99        192.168.1.68       TCP       http > 62216 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
    52 140.079583  192.168.1.68       66.102.9.99        TCP       62216 > http [ACK] Seq=1 Ack=1 Win=65780 Len=0
    53 140.080278  192.168.1.68       66.102.9.99        HTTP      GET /complete/search?hl=en&client=suggest&js=true&q=m&cp=1 H
    54 140.086765  192.168.1.68       66.102.9.99        TCP       62216 > http [FIN, ACK] Seq=805 Ack=1 Win=65780 Len=0
    55 140.086921  192.168.1.68       66.102.9.99        TCP       62218 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
    56 140.197484  66.102.9.99        192.168.1.68       TCP       http > 62216 [ACK] Seq=1 Ack=805 Win=7360 Len=0
    57 140.[illegible]  66.[illegible].9.99   192.168.1.68       TCP       http > 62216 [FIN, ACK] Seq=1 Ack=806 Win=7360 Len=0
    58 140.[illegible]  1[illegible]8.1.68      66.102.9.99        TCP       62216 > http [ACK] Seq=806 Ack=2 Win=65780 Len=0

▷ Frame 1 (42 bytes on wire, 42 bytes captured)
▷ Ethernet II, Src: Vmware_38:eb:0e (00:0c:29:38:eb:0e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 00 0c  29 38 eb 0e 08 06 00 01   ........ )8......
0010  08 00 06 04 00 01 00 0c  29 38 eb 0e c0 a8 39 80   ........ )8....9.
0020  00 00 00 00 00 00 c0 a8  39 02                     ........ 9.

eth0: <live capture in progress> Fil...   Packets: 445 Displayed: 445 Marked: 0   Profile: Default
```



```
No. -  Time      Source       Destination      Protocol  Info
    1  0.000000  192.168.0.2  Broadcast        ARP       Who has 192.168.0.2?  Gratuitous
    2  0.299139  192.168.0.1  192.168.0.2      NBNS      Name query NBSTAT *<00><00><00><0
    3  0.299214  192.168.0.2  192.168.0.1      ICMP      Destination unreachable (Port unr
    4  1.025659  192.168.0.2  IGMP.MCAST.NET   IGMP      V3 Membership Report
    5  1.044366  192.168.0.2  192.168.0.1      DNS       Standard query SRV _ldap._tcp.nbg
    6  1.048652  192.168.0.2  239.255.255.250  UDP       Source port: 3193  Destination po
    7  1.050784  192.168.0.2  192.168.0.1      DNS       Standard query SOA nb10061d.ww004
    8  1.055053  192.168.0.1  192.168.0.2      UDP       Source port: 1900  Destination po
    9  1.082038  192.168.0.2  192.168.0.255    NBNS      Registration NB NB10061D<00>
   10  1.111945  192.168.0.2  192.168.0.1      DNS       Standard query A proxyconf.ww004.
   11  1.226156  192.168.0.2  192.168.0.1      TCP       3196 > http [SYN] Seq=0 Len=0 MSS
   12  1.227282  192.168.0.1  192.168.0.2      TCP       http > 3196 [SYN, ACK] Seq=0 Ack=

⊞ Frame 11 (62 bytes on wire, 62 bytes captured)
⊞ Ethernet II, Src: 192.168.0.2 (00:0b:5d:20:cd:02),
⊞ Internet Protocol, Src: 192.168.0.2 (192.168.0.2),
⊟ Transmission Control Protocol, Src Port: 3196 (319
    Source port: 3196 (3196)
    Destination port: http (80)
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊞ Flags: 0x0002 (SYN)
    Window size: 64240

0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 0
0010  00 30 18 48 40 00 80 06  61 2c c0 a8 00 02 c0 a
0020  00 01 0c 7c 00 50 3c 36  95 f8 00 00 00 00 70 0
0030  fa f0 27 e0 00 00 02 04  05 b4 01 01 04 02

File: "D:\test.pcap" 14 KB 00:00:02    P: 120 D: 120 M: 0
```

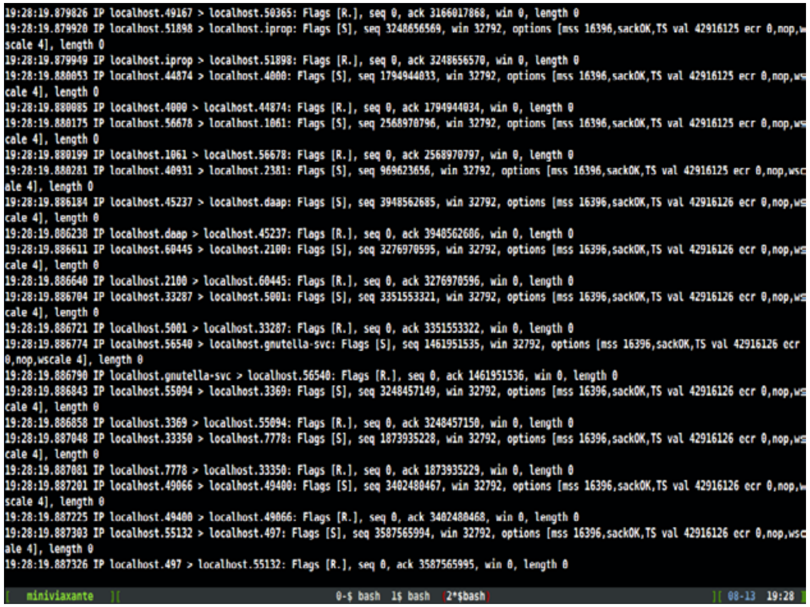
By: ● Display filter ○ Hex value ○ String

Filter:

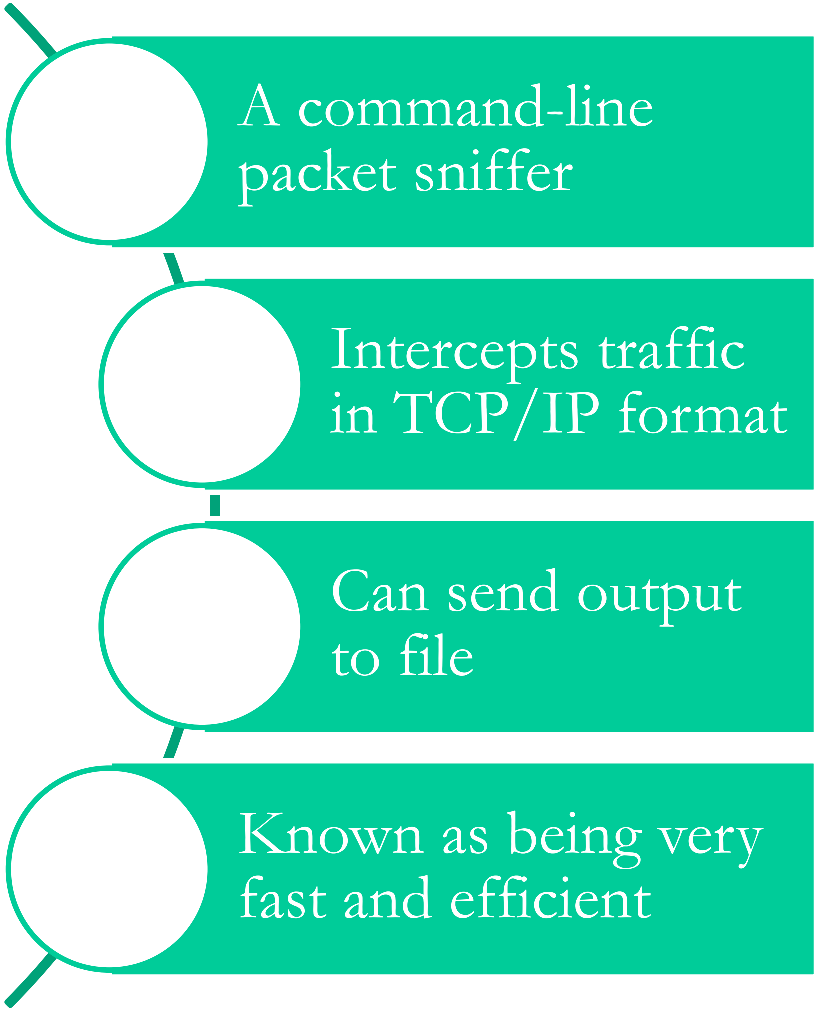
Direction: ○ Up ● Down

tcpdump

tcpdump is an open source network utility that is freely available under the BSD license.



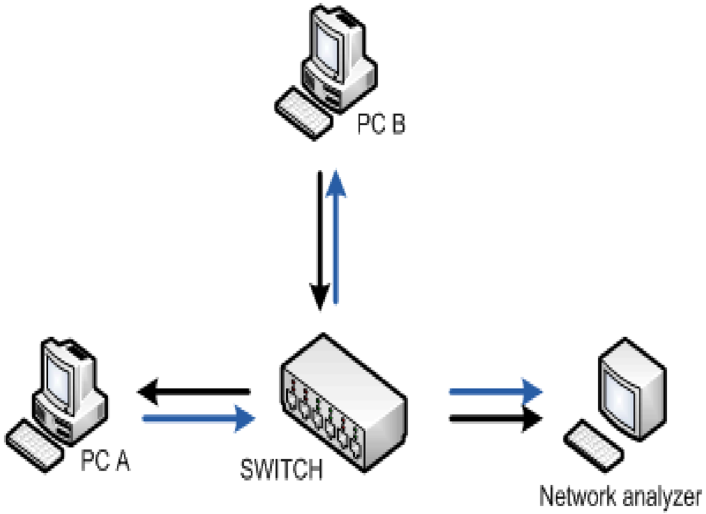
```
19:28:19.879826 IP localhost.49167 > localhost.50365: Flags [R.], seq 0, ack 3166017868, win 0, length 0
19:28:19.879920 IP localhost.51898 > localhost.iprop: Flags [S], seq 3248656569, win 32792, options [mss 16396,sackOK,TS val 42916125 ecr 0,nop,w
scale 4], length 0
19:28:19.879949 IP localhost.iprop > localhost.51898: Flags [R.], seq 0, ack 3248656570, win 0, length 0
19:28:19.880053 IP localhost.44874 > localhost.4000: Flags [S], seq 1794944033, win 32792, options [mss 16396,sackOK,TS val 42916125 ecr 0,nop,ws
cale 4], length 0
19:28:19.880085 IP localhost.4000 > localhost.44874: Flags [R.], seq 0, ack 1794944034, win 0, length 0
19:28:19.880175 IP localhost.56678 > localhost.1061: Flags [S], seq 2568970796, win 32792, options [mss 16396,sackOK,TS val 42916125 ecr 0,nop,ws
cale 4], length 0
19:28:19.880199 IP localhost.1061 > localhost.56678: Flags [R.], seq 0, ack 2568970797, win 0, length 0
19:28:19.880281 IP localhost.40931 > localhost.2381: Flags [S], seq 969623656, win 32792, options [mss 16396,sackOK,TS val 42916125 ecr 0,nop,wsc
ale 4], length 0
19:28:19.886184 IP localhost.45237 > localhost.daap: Flags [S], seq 3948562685, win 32792, options [mss 16396,sackOK,TS val 42916126 ecr 0,nop,ws
cale 4], length 0
19:28:19.886238 IP localhost.daap > localhost.45237: Flags [R.], seq 0, ack 3948562686, win 0, length 0
19:28:19.886611 IP localhost.60445 > localhost.2100: Flags [S], seq 3276970595, win 32792, options [mss 16396,sackOK,TS val 42916126 ecr 0,nop,ws
cale 4], length 0
19:28:19.886640 IP localhost.2100 > localhost.60445: Flags [R.], seq 0, ack 3276970596, win 0, length 0
19:28:19.886704 IP localhost.33287 > localhost.5001: Flags [S], seq 3351553321, win 32792, options [mss 16396,sackOK,TS val 42916126 ecr 0,nop,ws
cale 4], length 0
19:28:19.886721 IP localhost.5001 > localhost.33287: Flags [R.], seq 0, ack 3351553322, win 0, length 0
19:28:19.886774 IP localhost.56540 > localhost.gnutella-svc: Flags [S], seq 1461951535, win 32792, options [mss 16396,sackOK,TS val 42916126 ecr
0,nop,wscale 4], length 0
19:28:19.886790 IP localhost.gnutella-svc > localhost.56540: Flags [R.], seq 0, ack 1461951536, win 0, length 0
19:28:19.886843 IP localhost.55094 > localhost.3369: Flags [S], seq 3248457149, win 32792, options [mss 16396,sackOK,TS val 42916126 ecr 0,nop,ws
cale 4], length 0
19:28:19.886858 IP localhost.3369 > localhost.55094: Flags [R.], seq 0, ack 3248457150, win 0, length 0
19:28:19.887048 IP localhost.33350 > localhost.7778: Flags [S], seq 1873935228, win 32792, options [mss 16396,sackOK,TS val 42916126 ecr 0,nop,ws
cale 4], length 0
19:28:19.887081 IP localhost.7778 > localhost.33350: Flags [R.], seq 0, ack 1873935229, win 0, length 0
19:28:19.887201 IP localhost.49066 > localhost.49400: Flags [S], seq 3402480467, win 32792, options [mss 16396,sackOK,TS val 42916126 ecr 0,nop,w
scale 4], length 0
19:28:19.887225 IP localhost.49400 > localhost.49066: Flags [R.], seq 0, ack 3402480468, win 0, length 0
19:28:19.887303 IP localhost.55132 > localhost.497: Flags [S], seq 3587565994, win 32792, options [mss 16396,sackOK,TS val 42916126 ecr 0,nop,wsc
ale 4], length 0
19:28:19.887326 IP localhost.497 > localhost.55132: Flags [R.], seq 0, ack 3587565995, win 0, length 0

[   miniviaxante   ][          0-$ bash  1$ bash  (2*$bash)          ][ 08-13  19:28 ]
```



- A command-line packet sniffer
- Intercepts traffic in TCP/IP format
- Can send output to file
- Known as being very fast and efficient

Active Sniffing Close-Up

When sniffing is performed in a switched network, it is called active sniffing.



Active sniffing means the network has a switch instead of a hub.

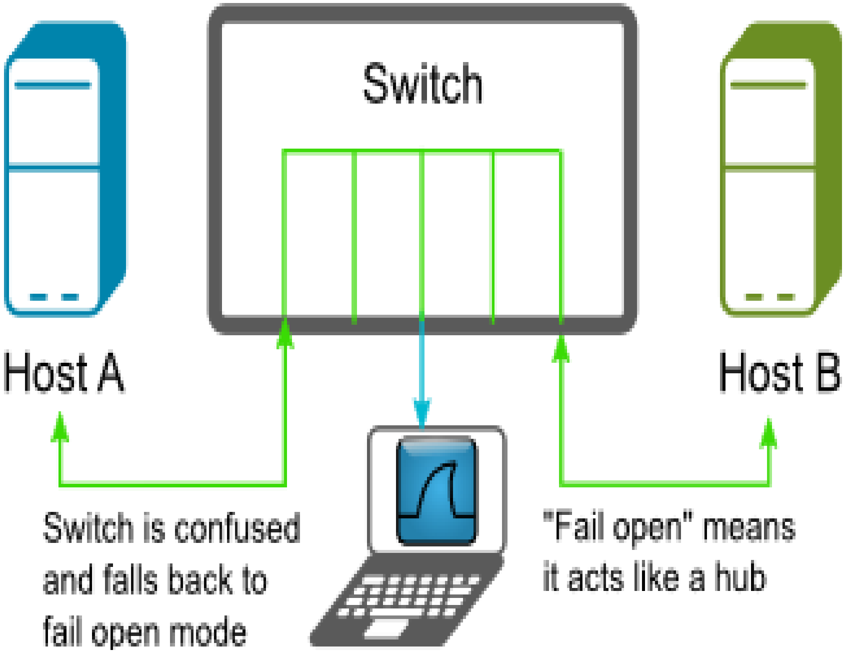
The switch actively regulates traffic.

The switch uses Address Resolution Protocol (ARP) to direct traffic.

The switch maintains an ARP table in memory to track MAC addresses.

MAC Flooding

A switch keeps track of MAC addresses received by writing them to a content addressable memory (CAM) table. If a switch is flooded with MAC addresses, it may easily overwhelm the switch's ability to write to its own CAM table.

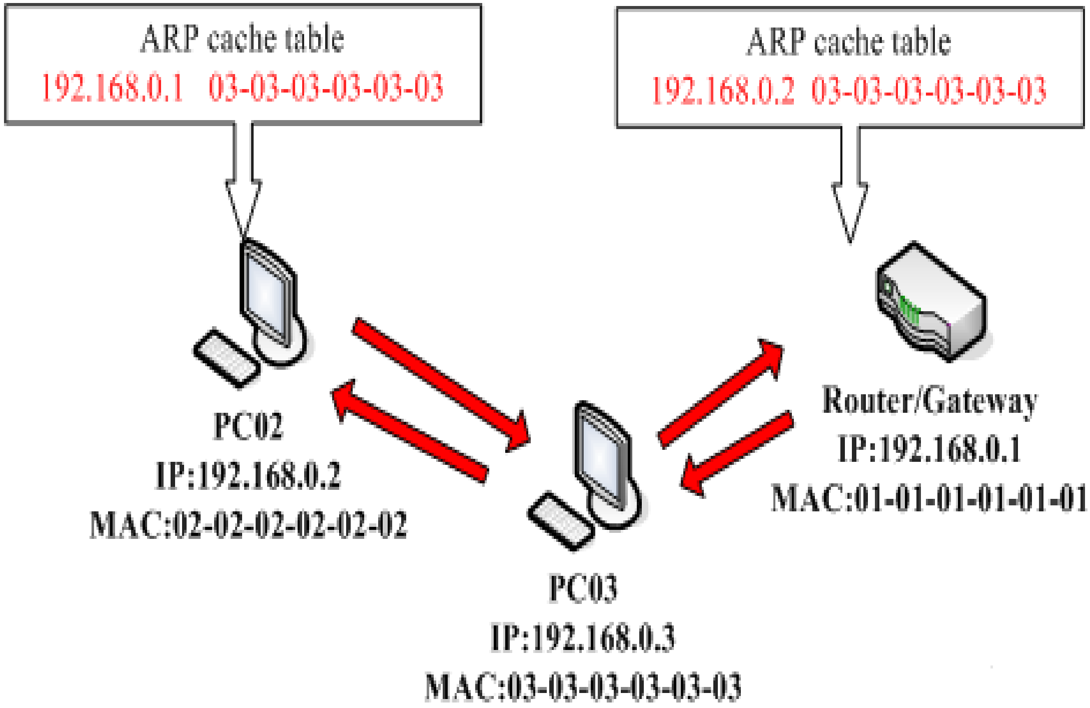

Switch
Host A
Host B
Switch is confused and falls back to fail open mode
"Fail open" means it acts like a hub

Involves flooding the switch with numerous requests

Overloads the CAM table in the switch

Causes switch to fail and act like a hub

ARP Spoofing



ARP cache table
192.168.0.1 03-03-03-03-03-03

ARP cache table
192.168.0.2 03-03-03-03-03-03

PC02
IP:192.168.0.2
MAC:02-02-02-02-02-02

Router/Gateway
IP:192.168.0.1
MAC:01-01-01-01-01-01

PC03
IP:192.168.0.3
MAC:03-03-03-03-03-03

Denial of service

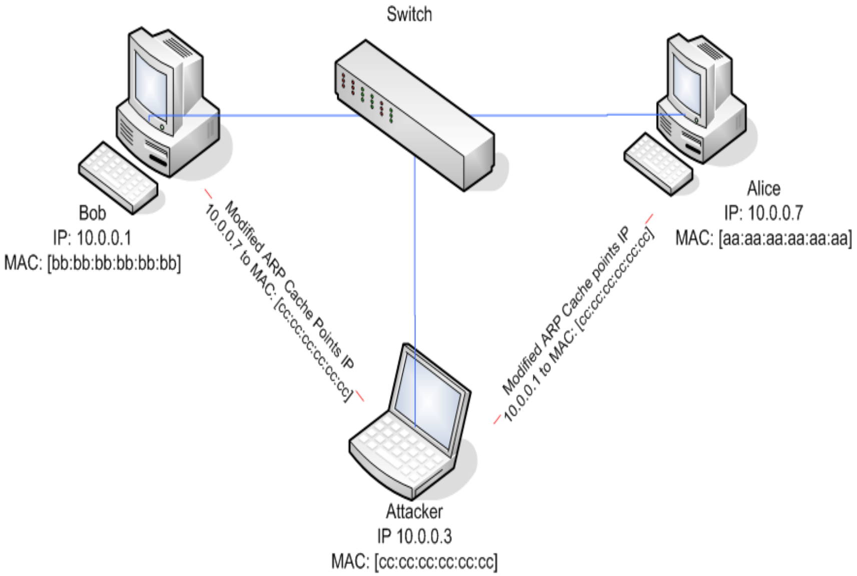
Man-in-the-middle/sniffing

MAC flooding

The ARP protocol is a simple and efficient protocol, but one drawback is its lack of authentication, and as a result, there is no way to verify the IP to MAC address mapping.

MAC Spoofing

MAC spoofing is a simple concept in which an attacker (or pen tester) changes their MAC address to the MAC address of an existing authenticated machine already on the network.

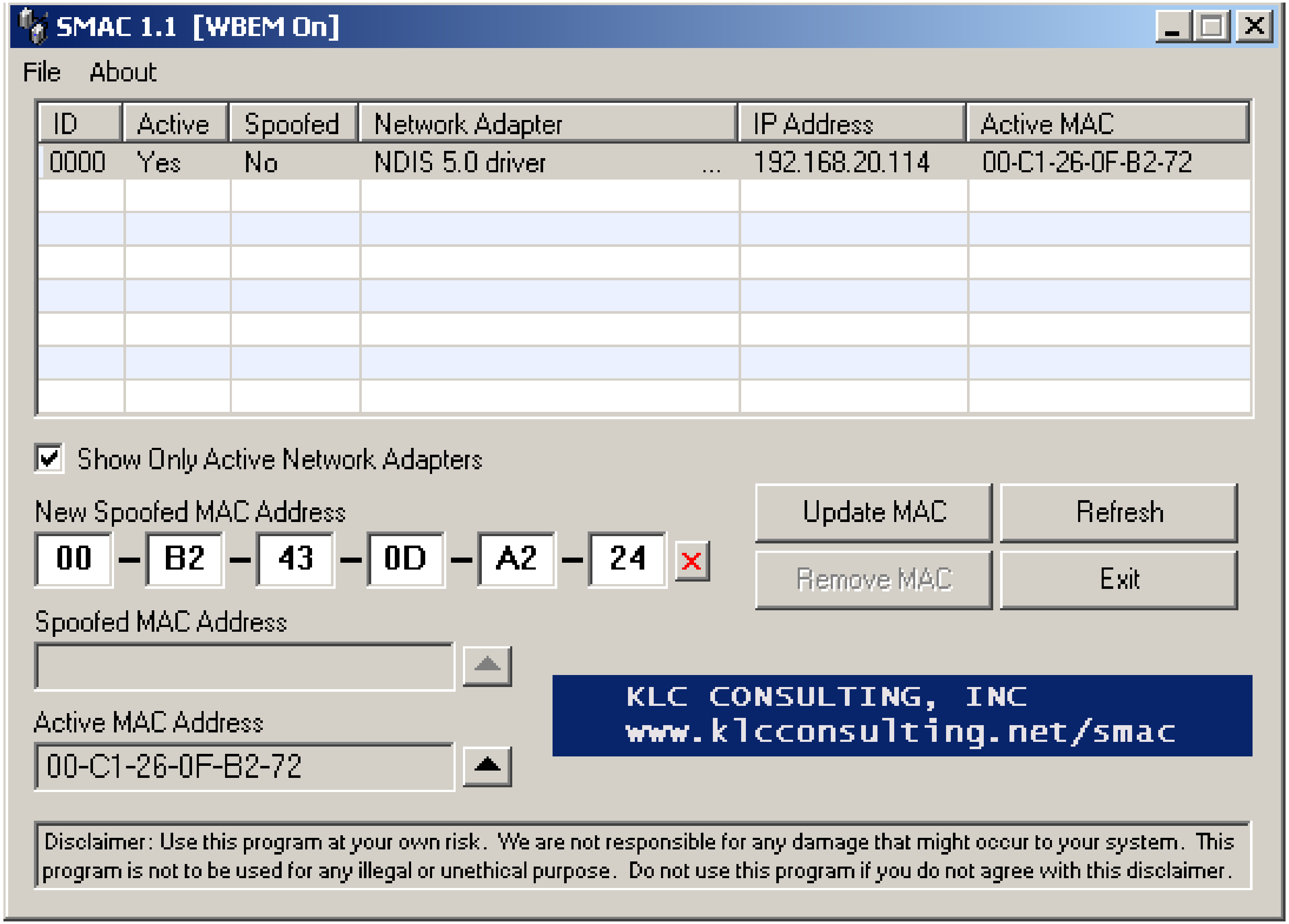

Switch
Bob
IP: 10.0.0.1
MAC: [bb:bb:bb:bb:bb:bb]
Alice
IP: 10.0.0.7
MAC: [aa:aa:aa:aa:aa:aa]
Modified ARP Cache Points IP 10.0.0.7 to MAC: [cc:cc:cc:cc:cc:cc]
Modified ARP Cache points IP 10.0.0.1 to MAC: [cc:cc:cc:cc:cc:cc]
Attacker
IP 10.0.0.3
MAC: [cc:cc:cc:cc:cc:cc]

Fakes the MAC address of an existing client

Allows a system to impersonate another

Can allow for the bypass of any mechanism that uses a MAC address to control traffic

SMAC



ID	Active	Spoofed	Network Adapter	IP Address	Active MAC
0000	Yes	No	NDIS 5.0 driver ...	192.168.20.114	00-C1-26-0F-B2-72

☑ Show Only Active Network Adapters

New Spoofed MAC Address

00 – B2 – 43 – 0D – A2 – 24 ☒

Spoofed MAC Address

Active MAC Address

00-C1-26-0F-B2-72

Disclaimer: Use this program at your own risk. We are not responsible for any damage that might occur to your system. This program is not to be used for any illegal or unethical purpose. Do not use this program if you do not agree with this disclaimer.

Sniffing Countermeasures

Use a hardware-switched network for the most sensitive portions of your network.

Implement IP DHCP snooping on switches to prevent ARP poisoning and spoofing attacks.

Implement policies preventing promiscuous mode on network adapters.

Be careful when deploying wireless access points, knowing that all traffic on the wireless network is subject to sniffing.

Encrypt your sensitive traffic using an encrypting protocol such as SSH or IPsec.

Summary

- Sniffing allows the interception of network traffic.
- Sniffing targets vulnerable or insecure network protocols.
- Sniffing uses packet sniffers to gather traffic.
- Sniffing comes in active and passive modes.
- Sniffing can be impacted by hubs and switches.