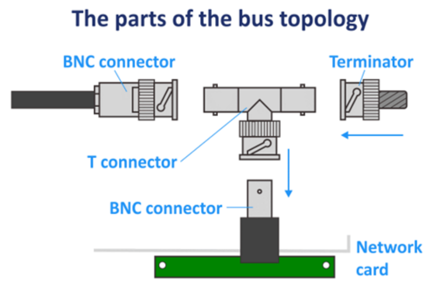
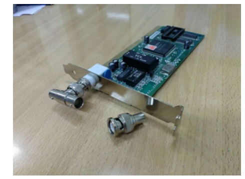
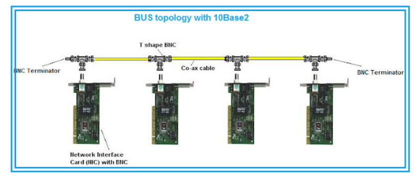
Packet sniffing lab

CalPolyPomona

LAN with T connectors (e.g., Eth 10Base2) – old times

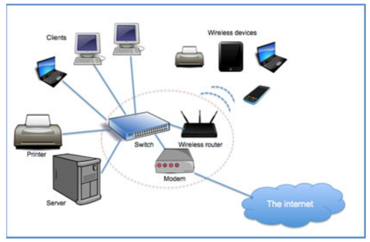


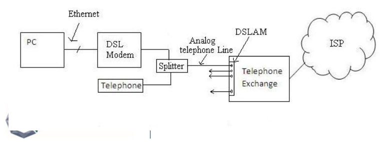


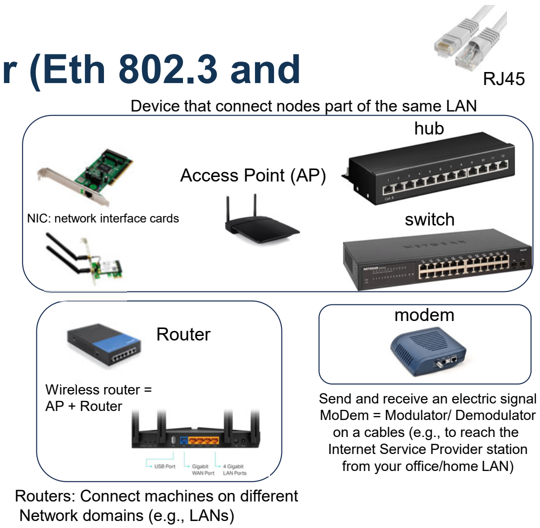


Each signal is transmitted and reaches all machines <u>physically</u>

LAN with RJ45 or air (Eth 802.3 and 802.11abg)

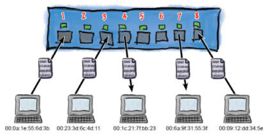






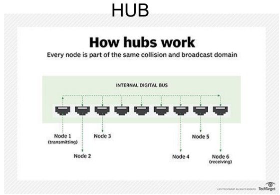
LAN connectors RJ45 and Ethernet connection

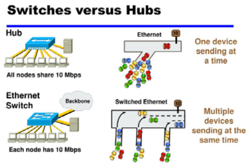
SWITCH

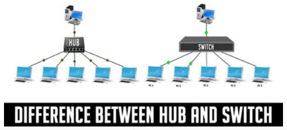


MAC address	Port
00:0a:1e:55:6d:3b	1
00:23:3d:6c:4d:11	3
00:1c:21:7f:bb:23	4
00:6a:9f:31:55:3f	7
00:09:12:dd:34:5e	8

HUB

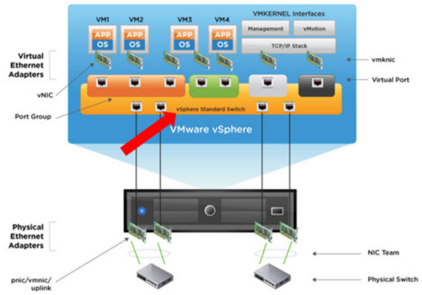






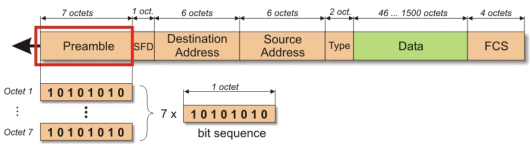
Hub	Switch
Hub is a broadcast device.	The switch is a multicast device.
Hub works in the physical layer of OSI Model.	The switch works in data link layer of OSI Model.
Hub sends data in the form of binary bits.	The switch sends data in the form of frames.
Transfers data to all the connected ports.	Transfers the data to the port for which it is addressed.
Hubs are connected to the system via the half-duplex connection.	Switches are connected to the system via the full-duplex connection.
Less expensive than the Switches.	More expensive than the Hubs.
The number of ports in hubs is between 4 and 24.	The number of ports in Switches is between 4 and 48.
Only one device can send data at a time.	Multiple devices can send data simultaneously at the same time.

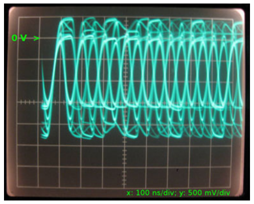
Virtual LANs with virtualization (VMware)



- Works in a similar way of physical devices but the software "recreates" emulated network components as seen before (Ethernet cards, Switches and routers)

Ethernet Frame



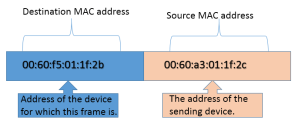


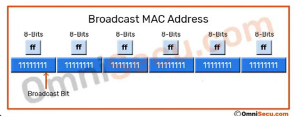
PREAMBLE – 10101010 ... 10101010 (ETHERNET 2)
8 BYTES 10101010 ... 10101011 (802.3)

Preamble changes from 802.3 (wired Ethernet) to WiFi 802.11abg, but the purpose is the same: each network card catches the preamble signal to understand when the ethernet frame starts

Destination and source address

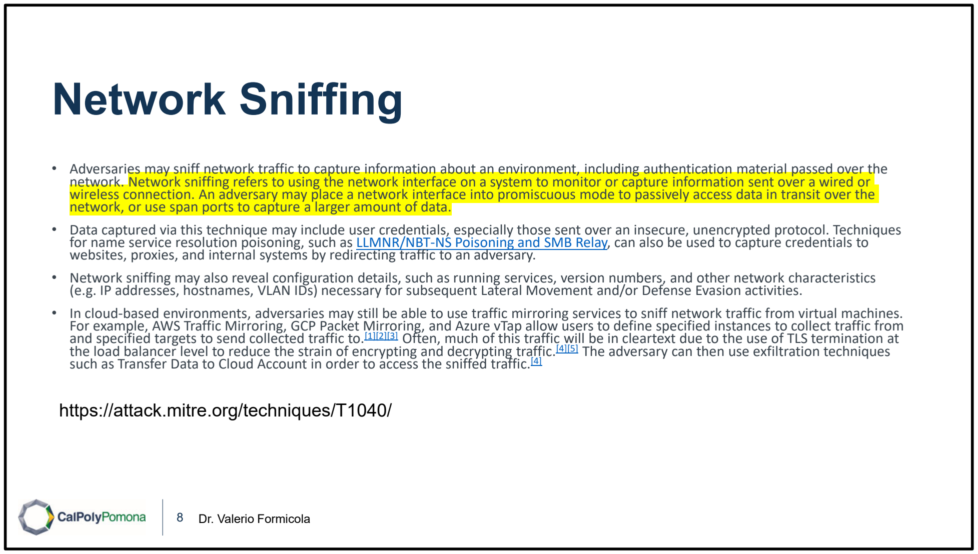
- Once the frame starts, the first thing arriving to the network card is the destination (physical) address, also known as **destination MAC address**





Default mode of operation for an Ethernet card ***(Unicast mode)***

If (destination_MAC_address == my_MAC_address)
    receive_all_frame
else
    skip_frame



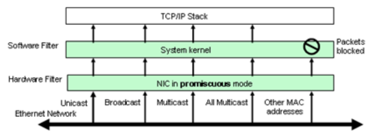
Denial of service is a form of attack on the availability of some service. In the context of computer and communications security, the focus is generally on network services that are attacked over their network connection. We distinguish this form of attack on availability from other attacks, such as the classic acts of god, that cause damage or destruction of IT infrastructure and consequent loss of service.

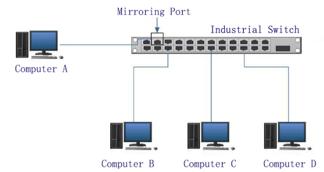
NIST SP 800-61 (*Computer Security Incident Handling Guide* , August 2012) defines denial-of-service (DoS) attack as follows:

A **denial of service (DoS)** is an action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources such as central processing units (CPU), memory, bandwidth, and disk space.

How do we capture all the traffic on a LAN?

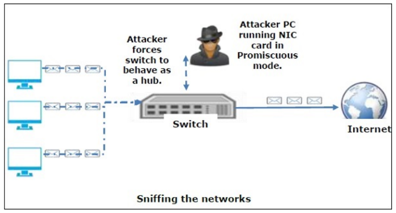
1) Set your network card in "Promiscuous mode"
   - It makes sure your NIC doesn't skip frames not intended for the interface MAC address, .e., observe anything "on the wire"
   - Note: wireshark should set it for you, if you use that as a sniffing tool





2) Connect your computer to a **HUB** or to the **mirror port** of a network switch
   - Network switches implement port segmentation which doesn't allow your machine to receive all network traffic, i.e., packets not for your MAC address; that's why they have a **mirror port**
   - Switch vendors tell you which port is for mirroring, or you can set it from the switch configuration interface

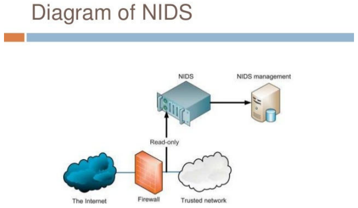
Network Sniffing: malicious vs good intent



Sniffing based attack, ways to obtain (alternatives):

1. The attacker physically connect to a hub
2. The attacker connects to the managing port of the switch and removes the segmentation features, transforming it into a hub
3. The attacker access a host which is able to see all network traffic (e.g., a Network Intrusion Detection System, right picture here)

Sniffing is a security solution for data-in-transit

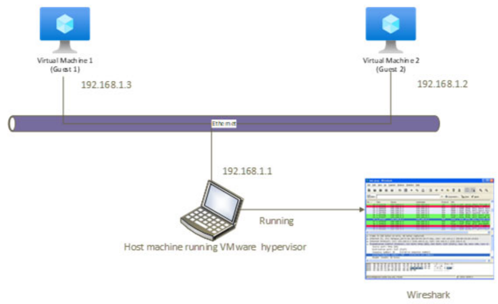


Network-based Intrusion Detection System
N-IDS

LAN sniffing in VMware

Set up

- Live…
- Note: IP addresses might be different



Example of protocol stack

FIGURE 1: INTERNET PROTOCOL STACK

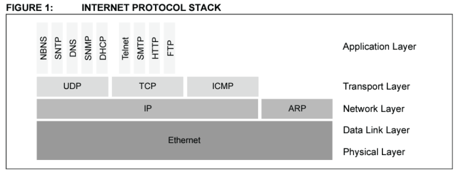
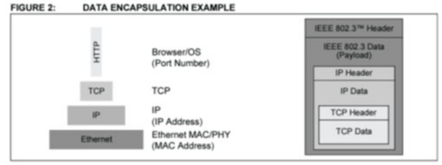
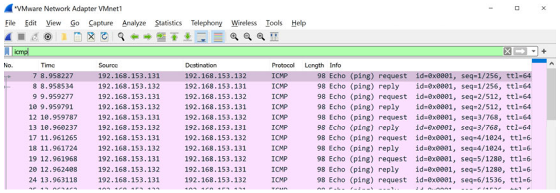


FIGURE 2: DATA ENCAPSULATION EXAMPLE



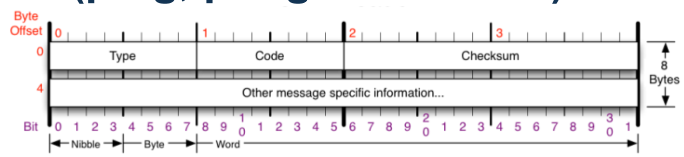
Exercise 1 – sniff a ping pong

1. Run Wireshark on Host machine
2. Ping VM1 with VM2
3. Observe Ping between VM1 and VM2
   - ICMP Echo Request and Reply
   - Apply display filter: icmp
4. Inspect fields



https://wiki.wireshark.org/Internet_Control_Message_Protocol

ICMP (ping, pong and others)



ICMP Message Types

Type	Code/Name
0	Echo Reply
3	Destination Unreachable
	0 Net Unreachable
	1 Host Unreachable
	2 Protocol Unreachable
	3 Port Unreachable
	4 Fragmentation required, and DF set
	5 Source Route Failed
	6 Destination Network Unknown
	7 Destination Host Unknown
	8 Source Host Isolated
	9 Network Administratively Prohibited
	10 Host Administratively Prohibited
	11 Network Unreachable for TOS

Type	Code/Name
3	Destination Unreachable (continued)
	12 Host Unreachable for TOS
	13 Communication Administratively Prohibited
4	Source Quench
5	Redirect
	0 Redirect Datagram for the Network
	1 Redirect Datagram for the Host
	2 Redirect Datagram for the TOS & Network
	3 Redirect Datagram for the TOS & Host
8	Echo
9	Router Advertisement
10	Router Selection

Type	Code/Name
11	Time Exceded
	0 TTL Exceeded
	1 Fragment Reassembly Time Exceeded
12	Parameter Problem
	0 Pointer Problem
	1 Missing a Required Operand
	2 Bad Length
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
30	Traceroute

Checksum

Checksum of ICMP header

RFC 792

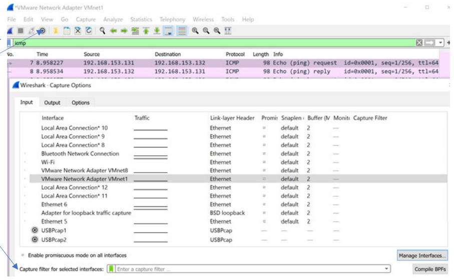
Please refer to RFC 792 for the Internet Control Message protocol (ICMP) specification.

Observation: Capture filter Vs Display filter

- **Capture filter** is a filter that drops data not matching the filter rule
- **Display filter** is a filter that simply shows some packets matching the filter in the visual interface. Other packets might still be captured

Display filter set up

Capture filter set up



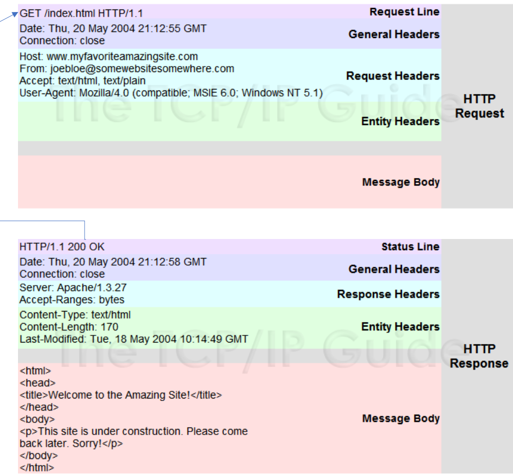
Exercise 2 – sniff a netcat

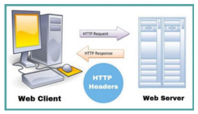
1. Establish a kept-alive client-server communication between VM1 and VM2 using netcat
   1. Server:
      nc –l –p 2000 -k
   2. Client :
      nc *server_IP_address* 2000
2. Execute a network sniffing with wireshark from host machine
3. Send some messages from Netcat client towards Netcat server
4. Stop the capture
5. Analyze the network caputres:
   - Visualize src and dst IP addresses from the packet capture on the host machine
   - Visualize what is the network transport protocol
   - Visualize if the destination port on the server is corresponding to the destination port on the captured packets (sniffed packets)
   - Visualize the content of your messages sent from client and server in Wireshark

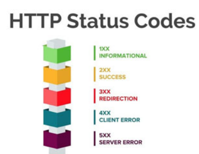
Exercise 3 – sniff an http communication

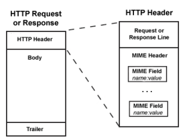
- Install apache webserver on one of the VMs
  - sudo apt -y install apache2
- Connect from the browser to http://IP_address_web_server
- Change the content of the webserver to a random page:
  - echo '<!doctype html><html><body><h1>Hello World!</h1></body></html>' | sudo tee /var/www/html/index.html
- Execute a network capture from Wireshark and observe the communication pattern
  - Identify the HTTP packet with "Hello World!"

HTTP messages

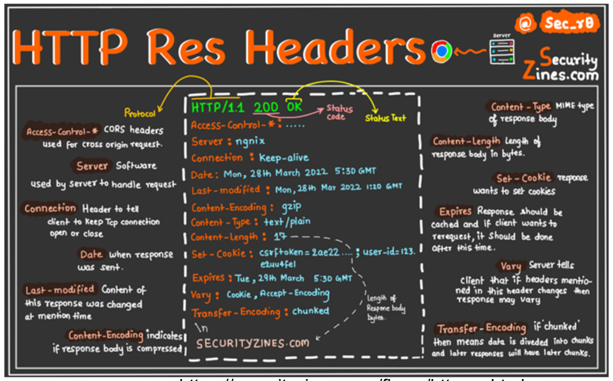








More header fields



https://securityzines.com/flyers/httpres.html

Exercise 4 – sniff using tcpdump

- A command line tool that uses same library of wireshark (libpcap or winpcap)
  - Remember to be sudoer
  - Cheatsheet: https://cdn.comparitech.com/wp-content/uploads/2019/06/tcpdump-cheat-sheet-1.jpg.webp
- Check available interfaces and their names:
  - tcpdump –D
- Command line for sniffing on any interfaces (or specify one) and stop after 5 packets:
  - tcpdump –i any –c5
- disable name resolution by using the option -n and port resolution with –nn:
  - tcpdump –i any –c5 –n –nn
- Filtering packets (e.g., only icmp packets):
  - sudo tcpdump -i any -c5 icmp
- Quite mode (less packet details):
  - sudo tcpdump -i any –q
- Capture http packets and also translate in ASCII format:
  - sudo tcpdump -i any -A port 80
- Save the capture on a file (pcap format) or read it:
  - sudo tcpdump -w capture.pcap
  - sudo tcpdump -r capture.pcap

Exercise 5 – build a sniffer in Python Scapy

- https://scapy.readthedocs.io/en/latest/introduction.html
- Install python3 (if not installed)
- Install scapy module:
  - sudo apt-get install python3-scapy
- First step, create a text file with py extinction
  - touch sniffer.py
- Second, make it executable (for all users, add executable permission on the file):
  - chmod a+x sniffer.py
  - Not necessary but makes life easier to run the script without calling python
- Then, we edit and try it (& will leave the gedit process in background, so you can continue use the current shell to test the python code):
  - gedit sniffer.py &

Exercise 6

- Write a Scapy program that:
  - captures only HTTP data and ICMP packets towards one of your virtual machines
  - saves the pcap files in a local folder where is the sniffer running
  - Bonus: capture and save pcap files from your host machine (e.g., Windows). The capture packets are related only to one of your guest virtual machines
    - For the bonus, you need to have python running on your host