CYBER CHALLENGE
CyberChallenge.IT

cini
Cybersecurity National Lab

SPONSOR PLATINUM

accenturesecurity

aizoon AUSTRALIA EUROPE USA
TECHNOLOGY CONSULTING

B5

EY Building a better working world

eni

exprivia | ITALTEL

IBM

KPMG

LEONARDO

NTT DATA
Trusted Global Innovator

NUMERA
SISTEMI E INFORMATICA S.p.A.

Telsy

SPONSOR GOLD

bip.

cisco

MONTE DEI PASCHI DI SIENA
BANCA DAL 1472

negg

NOVANEXT
connecting the future

pwc

SPONSOR SILVER

DiGi ONE
the leading digital company

ICT CYBER CONSULTING

Network analysis & monitoring

Francesco PALMIERI
Università di Salerno

fpalmieri@unisa.it
https://docenti.unisa.it/026587/home

CYBER CHALLENGE
CyberChallenge.IT

cini Cybersecurity National Lab

https://cybersecnatlab.it

License & Disclaimer

License Information

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Topics

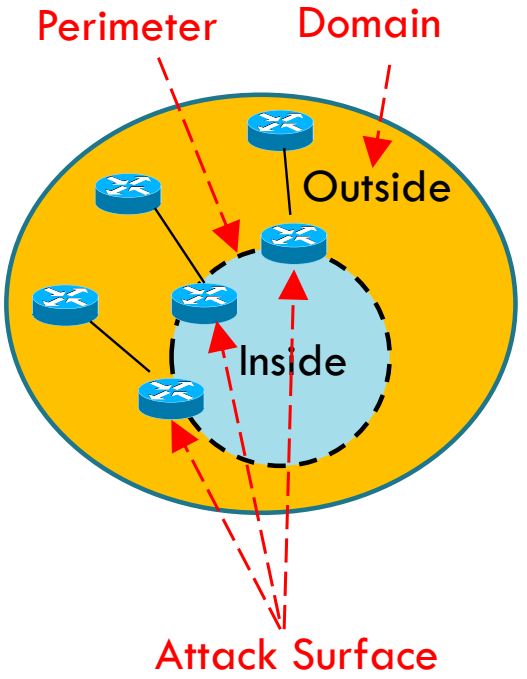
- Basic security architectural elements
- Traffic interception techniques
- Traffic Analysis tools and technologies
- Aggregated statistic traffic observations

Current Topic

- Basic security architectural elements
- Traffic interception techniques
- Traffic Analysis tools and technologies
- Aggregated statistic traffic observations

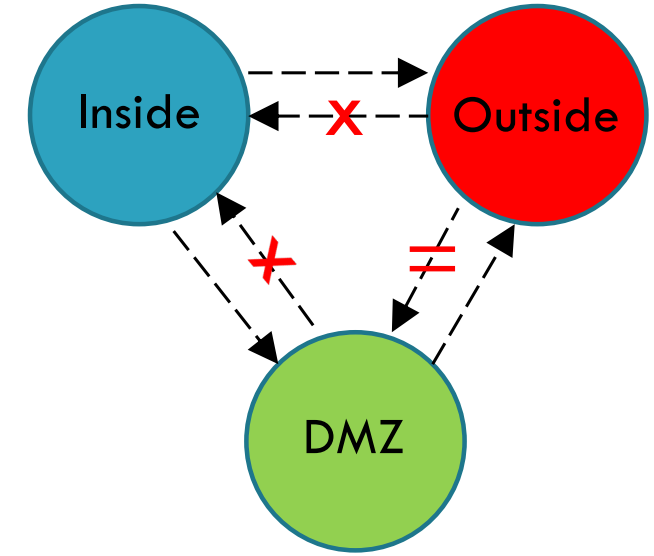
Domains, perimeter and attack surface

- A **security domain** is a set of entities/resources to be managed as a unique administration area according to a common security policy (security enforcement rules)
- A **security perimeter** is the secured boundary between the external and internal side of a security domain
  - e.g., an internal network and its public facing side, typically the Internet
  - The perimeter can be protected by several security devices
- The **attack surface** of a security domain is the sum of the different points ("attack vectors") where an unauthorized entity ("attacker") can try to enter data to or extract data or do any kind of unauthorized or hostile activity.
  - **Keeping the attack surface as small as possible** is a fundamental basic security measure

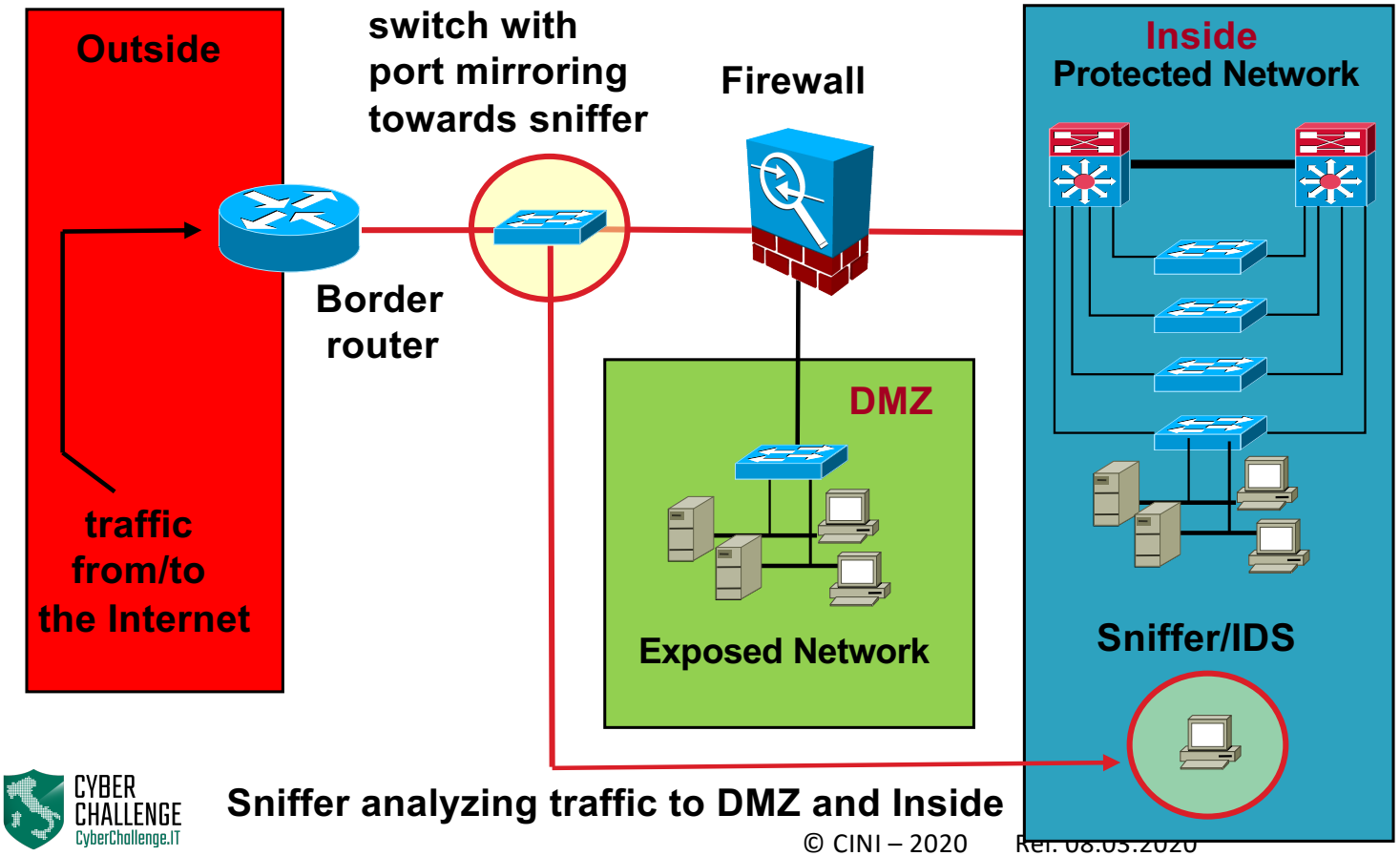


Security Domains

- Each security domain is assigned a **degree of trust** or **security level**
- Such value defines and characterizes its visibility rules (access rights) with respect to the others
  - A domain with a higher degree of trust can have full visibility than those with a lower degree
  - Vice versa, visibility is blocked unless specific exceptions (filtering / visibility rules) are defined
    - DMZ and INSIDE have full visibility of OUTSIDE
    - INSIDE has full visibility of DMZ
    - Any other access is not granted



Basic security architecture

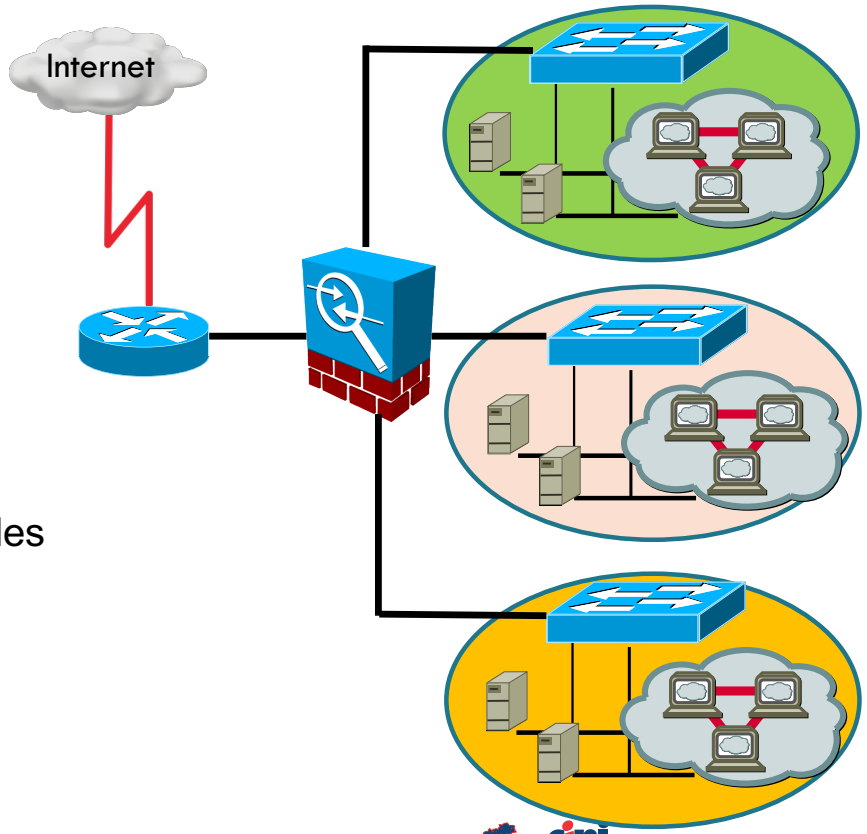


Sniffer analyzing traffic to DMZ and Inside

- In a common network architecture we have at least three domains:
  - **Outside** (all the world outside - the Internet): trust degree 0
  - **Inside** (the internal organization to be protected and hidden): degree of trust 100
  - **DMZ** (the set of internal machines that expose services outside): degree of trust 0 <x <100

Router, Firewall and Tapping Points

- A **router** is responsible for forwarding traffic between the internal network and the Internet
  - It is the first barrier or demarcation point,
  - often owned by the provider
- A **firewall** is a passive perimeter defense component that controls traffic flowing between two or more network segments associated to distinct **security domains**:
  - Separation of administratively different areas
  - Traffic filtering between different areas through visibility rules between domains (access control)
  - Mediation of access to specific applications
- A **tapping point** ensures traffic visibility and traffic monitoring



CYBER CHALLENGE
CyberChallenge.IT

 Rel. 08.03.2020

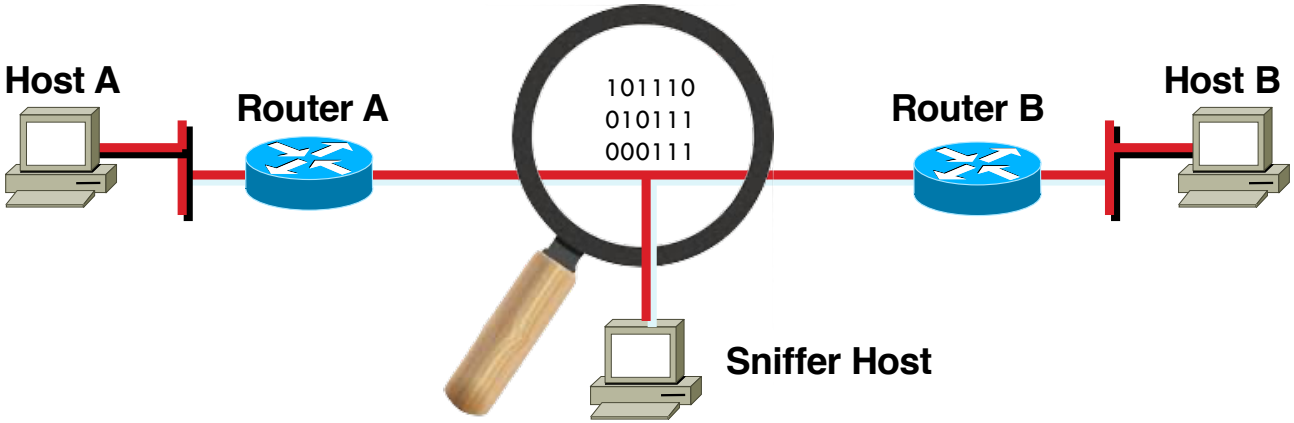
cini Cybersecurity National Lab

Current Topic

- Basic security architectural elements
- Traffic interception techniques
- Traffic Analysis tools and technologies
- Aggregated statistic traffic observations

Watching Traffic: Sniffing

- A sniffer is a software application that is capable of acquiring packets at the datalink level
- It is able to interpret clear information relating to level 2, 3 and 4 packet headers as well as application level protocols such as: FTP, HTTP, etc.
- A network adapter (NIC / TAP) programmed ad hoc (promiscuous mode) reads all packets in transit



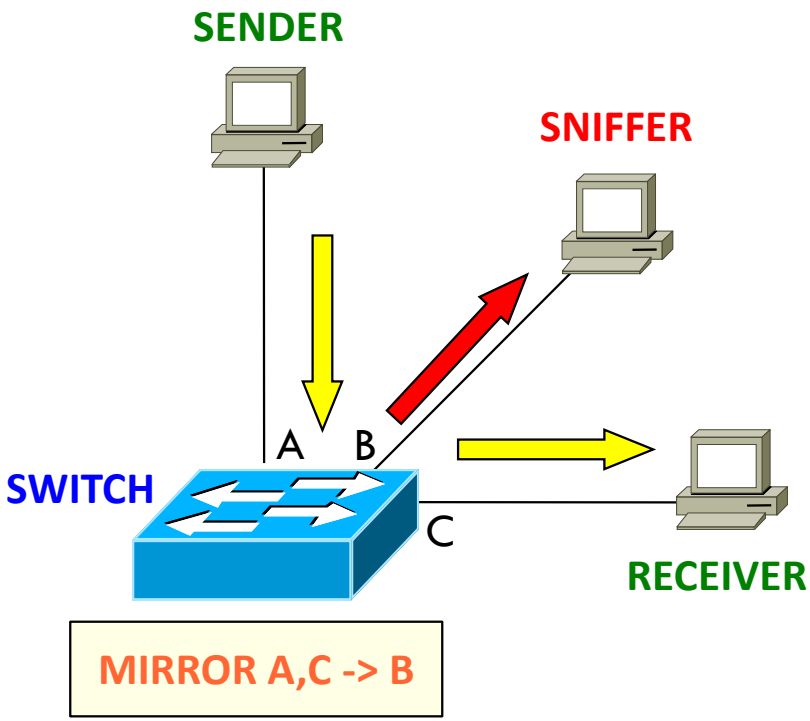
 Rel. 08.03.2020

Sniffing Applications

- **Automatic network analysis:** searching for specific patterns e.g. clear passwords and usernames: this is a common use for hackers / crackers;
- **Anomaly analysis**: in order to find out any problems within the networks, such as, why computer A cannot communicate with computer B;
- **Performance analysis**: to discover problems or bottlenecks in networks;
- **Detection of network intrusions**: so as to detect attacks or threats, as well as malicious activities in progress;
- **Recording of network traffic**: to create logs of network transactions available for subsequent "post-mortem" analysis.

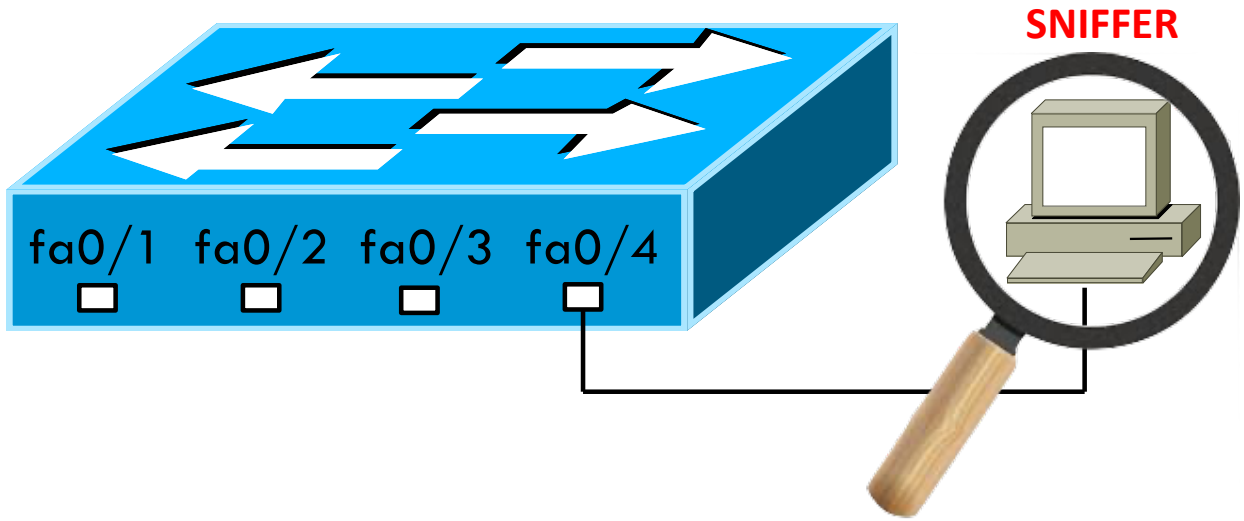
Sniffing on switched networks

- On switched networks, traffic is routed according to the MAC address + Port association, excluding terminals not interested in traffic
- Therefore a sniffer is only able to intercept the traffic destined for its hosting machine and the broadcast one
- The alternative is to configure the switch port to which the sniffer is connected in mirroring mode, from that moment it will replicate all the traffic received from specific ports on the sniffer port



Mirroring configuration

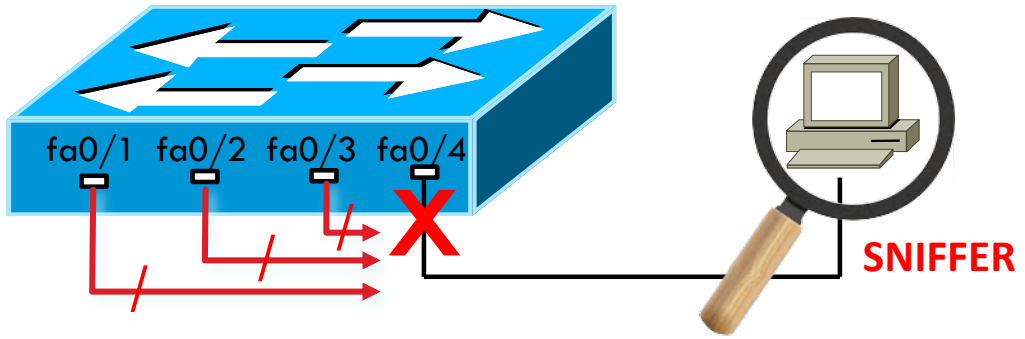
- Mirroring schemes:
  - 1 port to 1 port
  - Range of ports to 1 port
  - A whole VLAN to 1 port




```
Switch(config)#monitor session 1 source interface fa0/2
Switch(config)#monitor session 1 source interface fa0/1 - 3
Switch(config)#monitor session 1 source vlan 2
Switch(config)#monitor session 1 destination interface fa0/4
```

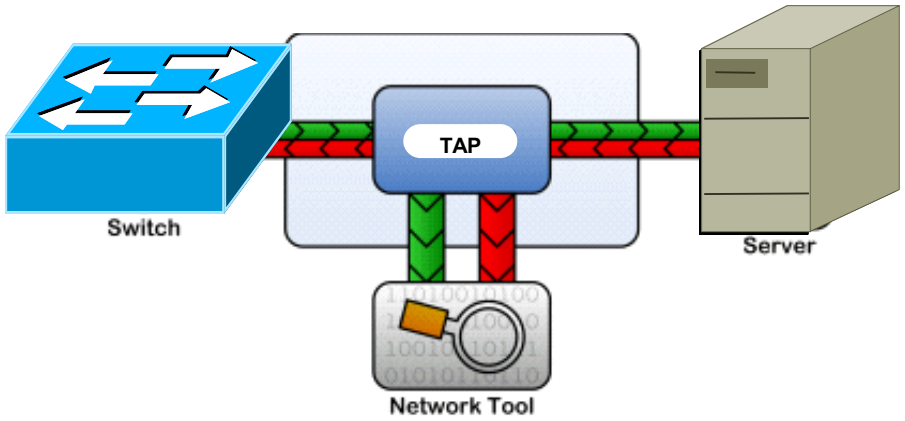
Sniffing without port mirroring

- In lack of port mirroring capabilities:
  - Use repeater devices (limited bands)
  - Use dedicated HW probes (TAP)
  - Perform traffic diversion through specific attacks (ARP Poisoning)



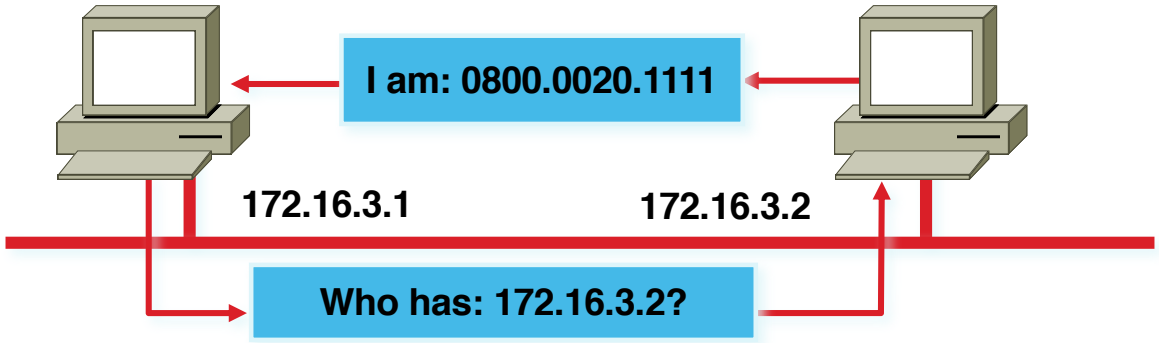
Traffic Access Port (TAP)

- HW solution that provides a copy of traffic on a section between 2 devices
- Requires no power supply
- 100% Visibility of Full Duplex Traffic including Errors or Anomalies at level 1 & 2
- Total isolation and safety of the sniffer
- It operates at level 1 and is very easy to install and manage (often transparent)
- It does not require specific configurations on switches or servers



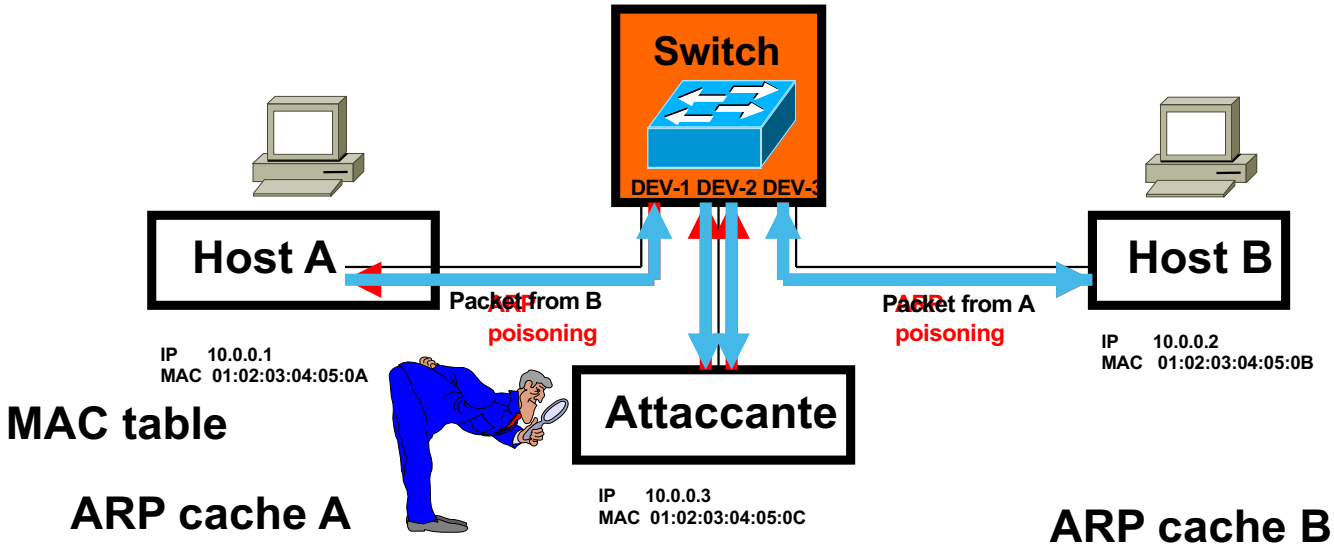
ARP Poisoning

- The Address Resolution Protocol (ARP) is concerned with mapping the 32 bits of IP address (version 4) into 48 bits of ETH address (MAC)
- Two main types of messages:
  - ARP request (request for IP address resolution)
  - ARP reply (reply containing an eth address)
- The replies are stored in the ARP CACHE, to limit traffic on the network



ARP poisoning

- Takes advantage of the stateless behavior of the protocol
- If the attacker sends an ARP reply (spoofed) to a host, this will save it in his ARP cache
- ARP replies are saved in cache even if they were not solicited (better performance at the expense of security)
- The cache entries are timed out, so the attacker must periodically "refresh"



MAC table

ARP cache A

DEV	MAC	STATE
DEV-1	01:02:03:04:05:0A	FORWARDING
DEV-3	01:02:03:04:05:0C	FORWARDING
DEV-2	01:02:03:04:05:0B	FORWARDING
...		

ARP cache B

IP	MAC
10.0.0.1	01:02:03:04:05:0C
10.0.0.3	01:02:03:04:05:0C

Example

- At startup A and B will have to exchange messages that allow their IP addresses to be associated with the physical Ethernet addresses, while the attacker will see only the packets:

```
16:38:36.501274 arp who-has 10.0.0.2 tell 10.0.0.1
16:38:36.509581 arp reply 10.0.0.2 is at 08:00:20:77:4d:db
```

- To intercept bidirectional communication, the program must be launched twice:

```
#./arpspoof –i eth0 –t 10.0.0.1 10.0.0.2
#./arpspoof –i eth0 –t 10.0.0.2 10.0.0.1
```

- In order for the packets to then return to the actual recipient, the attacker must send them back to the correct destination

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
```

Current Topic

- Basic security architectural elements
- Traffic interception techniques
- Traffic Analysis tools and technologies
- Aggregated statistic traffic observations

Tcpdump: a simple CLI-based sniffer

Sniffer: Software or hardware tool that by telying on promiscuous mode configuration captures and allows the analysis of all the packages that pass through a network segment

`tcpdump` : Sniffer public domain based on Berkeley packet filter (BPF)

Available for download: `ftp://ftp.ee.lbl.gov/tcpdump.tar.Z`

```
23:06:37 10.1.101.1 > 224.0.0.10: ip-proto-88 40 [tos 0xc0]
```

time **source IP** **dest IP** **protocol** **bytes** **type of srv**

Tcpdump: a simple CLI-based sniffer

```
08:08:16.155 spoofed.target.net.7 > 172.31.203.17.chargen: udp
```

timestamp | src IP | src port | dst IP | dst port | protocol

- hosts can be referenced by name or IP address
- the ports can be specified by number or name of the service
- to specify a range of values, specific bytes must be pointed to

Tcpdump: filtering expression

- Expressions define the criteria with which to choose what has to be displayed.
- Expressions consist of one or more primitives preceded by "qualifiers".

Source or destination host:	`host spoofed.target.net`
Destination network 172.31.x.x:	`dst net 172.31`
Destination networks 172.16 - 172.31:	`dst net 172 and (ip[17]>15) and (ip[17]<32)`
Source port 7:	`src port 7`
Destination port 19:	`dst port chargen`
Source port < 20:	`udp[0:2] < 20`
Destination port <20:	`udp[2:2] < 20`

Tcpdump: common qualifiers

- Type: host, net e port
  - Es. ‘`host 155.185.54.156`’, ‘`port 22`’, ecc.
- Dir: src, dst, src or dst
  - Es. ‘`src 155.185.54.156`’
- Proto: ether, fddi, tr, ip, ip6, arp, rarp, decnet, tcp and udp
  - Es. ‘`tcp port 21`‘, ‘`arp net 155.185.54`’

Packet sniffing example

```
# tcpdump 'port 23'

10.6.1.9.4548 > 10.6.1.2.23: S 2115515278:2115515278(0) win 32120 <mss 1460,
    nop,nop,sackOK,nop,wscale 0> (DF)

10.6.1.2.23 > 10.6.1.9.4548: S 1220480853:1220480853(0)
ack 2115515279 win 32120 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)

10.6.1.9.4548 > 10.6.1.2.23: . ack 1220480854 win 32120 (DF)
```

Wireshark

- Wireshark is a sophisticated new generation packet sniffer
- It has filtering functions and allows you to observe all traffic on a network.
- Identify encapsulations and recognize all individual fields.
- For the capture it has no code of its own, but uses libpcap / WinPcap.
- It is open source and compatible with Unix and Windows systems.

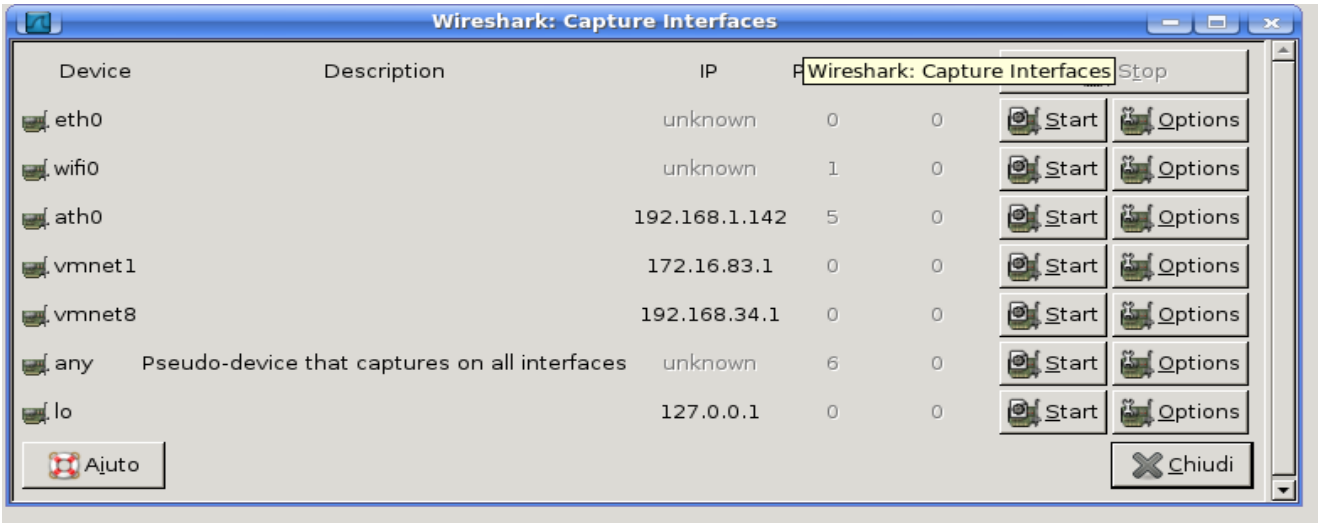
[Wikipedia: http://it.wikipedia.org/wiki/Wireshark *]*

Official site: http://www.wireshark.org/

Wireshark

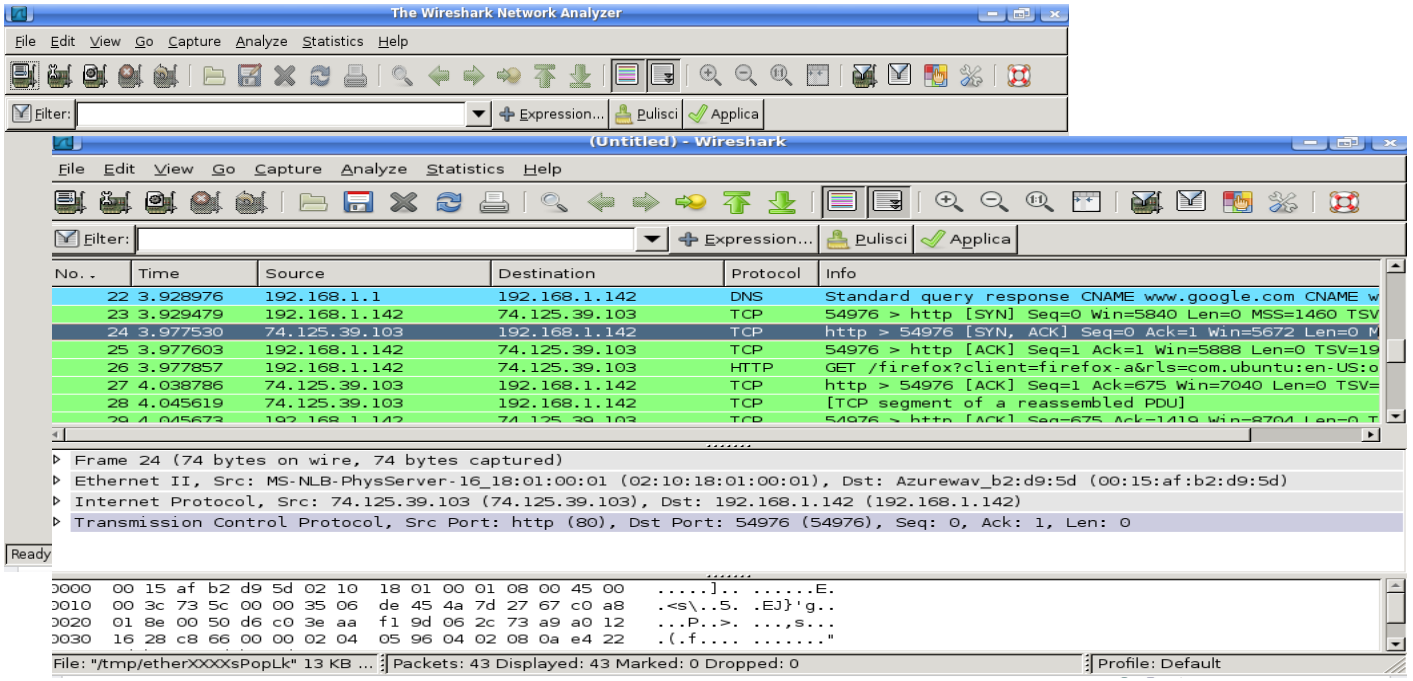
- We start our first capture by clicking on the first button from the left:

- the interface selection screen will open from which we can see all the interfaces on which we can operate, change the options for each interface or simply start capturing packages directly with the default options.



Wireshark

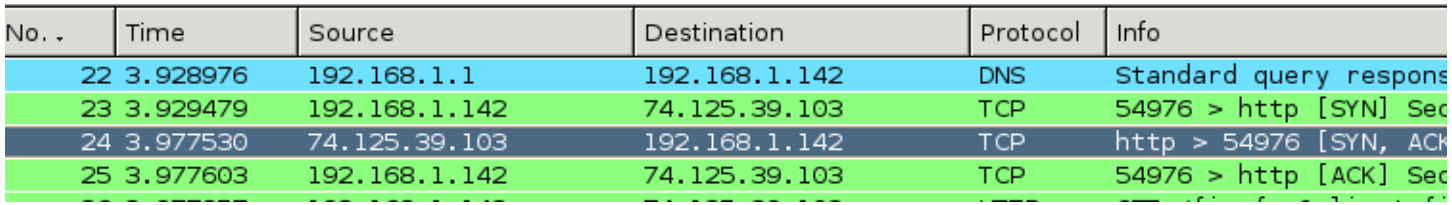
- Once the capture has started, the screen will change from a gray uniform, to a division into three sections
- Several lines and data will be presented, corresponding to the captured traffic info from different perspectives:
  - Detalink header
  - Network header
  - Transport header
  - Payload content



 Rel. 08.03.2020

Wireshark

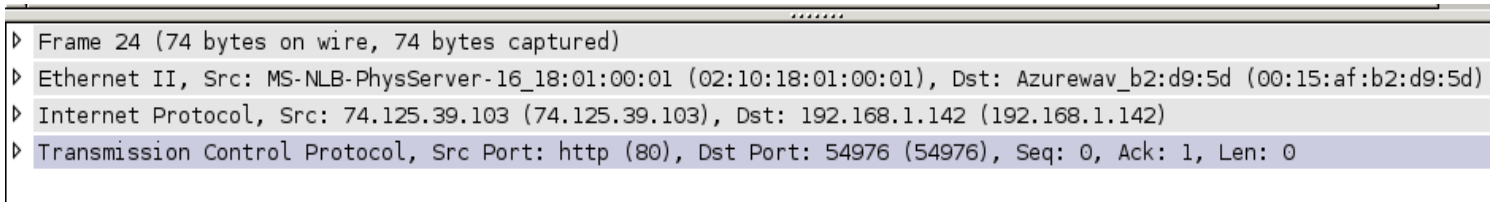
- The screen is divided into three sections, in the first section (summary), the top one, we have a further subdivision into columns, they represent (from left to right):



No.	Time	Source	Destination	Protocol	Info
22	3.928976	192.168.1.1	192.168.1.142	DNS	Standard query respons
23	3.929479	192.168.1.142	74.125.39.103	TCP	54976 > http [SYN] Seq
24	3.977530	74.125.39.103	192.168.1.142	TCP	http > 54976 [SYN, ACK
25	3.977603	192.168.1.142	74.125.39.103	TCP	54976 > http [ACK] Seq

  - the progressive number of the package
  - the time between the start of the capture and the arrival of the package
  - who generated the package (mac address or ip address)
  - who is the recipient of the packet (mac address, IP, broadcast)
  - the protocol used
  - in this section we can select a single row to better explore trafic

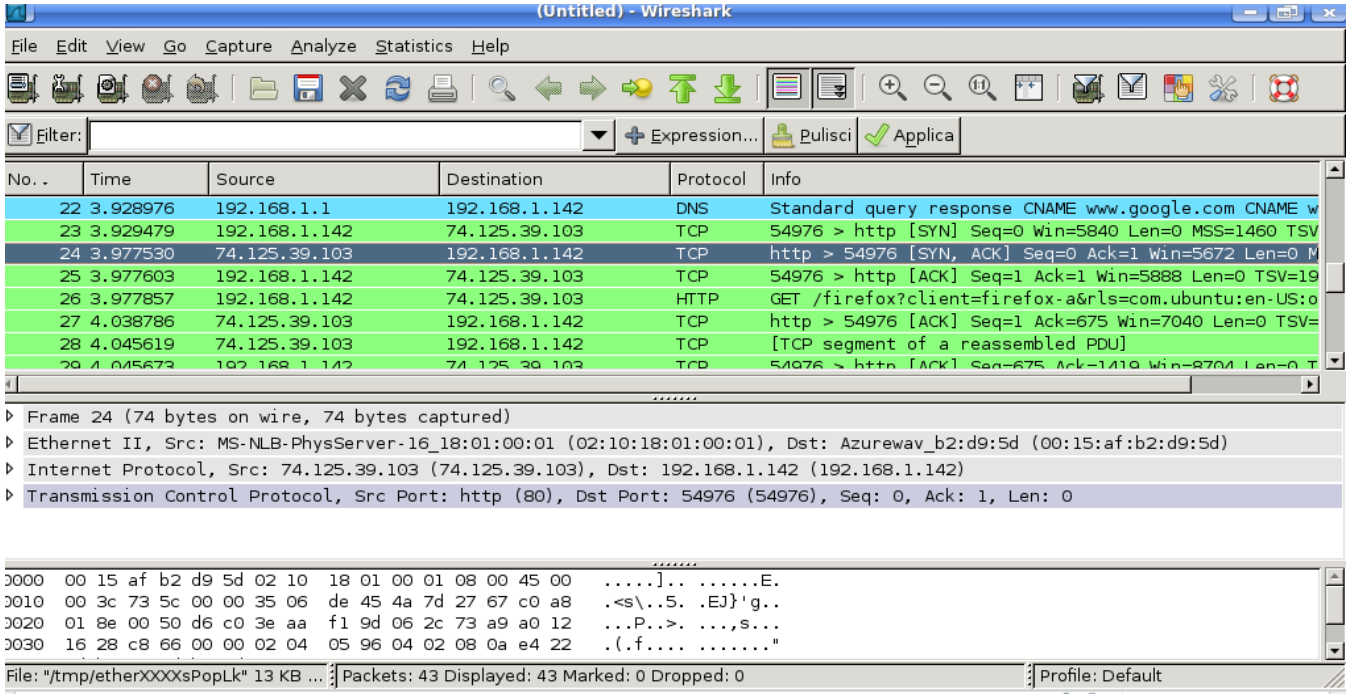
Wireshark



- The second section (protocol) details the data associated to the row selected in the first section. Here we can therefore better see
  - the type of frame,
  - the protocol from which the frame originates,
  - the source and recipient mac address in extended form,
  - any payload of the frame and other useful data, always organized according to hierarchies that can be inspected by clicking on the sign ▷ on the side

Wireshark

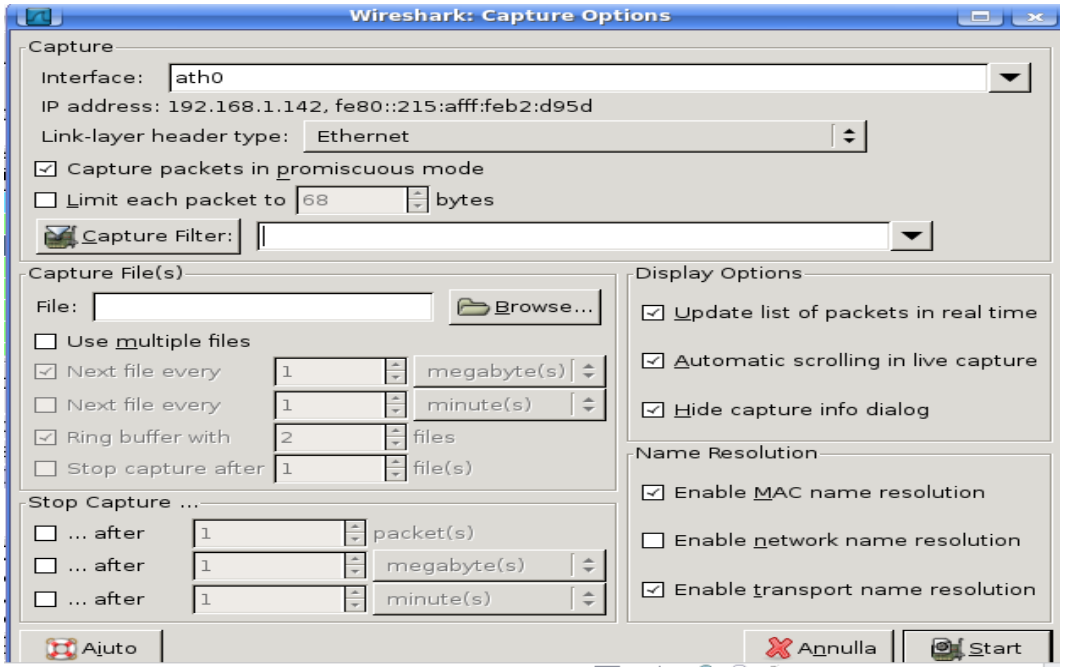
- In the third section (data) instead we see the native frame, in hex and ascii format, as it is acquired by the capture driver directly on the ethernet interface
- Here it is possible to highlight the bytes content associated to the previously selected sections



 Rel. 08.03.2020

Wireshark: filters usage

- There are two types of filters:
  - visualization filters
    - They allow you to view only what is defined by the filter
  - capture filters
    - They only capture packets that meet the criteria defined by the filter



Wireshark: filters usage

- The visualization filters, besides being very useful, are also facilitated in the use by the existence of numerous preset filters (Filter button in the filter bar)
- There is also a support facility that allows us to write new filters with a few clicks, accessible the "+ Expression" button,
- As we write in the filter row, it will change color based on the correctness of what we are writing.
- Once the filter is selected, we just have to apply it.
- To remove a filter just click on the "clean" button

Wireshark: example filtering options

- Here are some common filtering expressions:
  - `eth.addr == ff:ff:ff:ff:ff:ff`
  - `ip.addr == 192.168.1.7`
  - `ip.src == 192.168.1.17 and ip.dst == 192.168.1.19`
  - `ip.addr 192.168.1 and pop`
  - `ip.addr 192.168.1 and messenger`

Wireshark: example filtering options

- If instead we want to filter all packets that do not come / go to an IP, then the opposite of:

```
ip.addr==192.168.1.1
```

- we would be tempted to use:

```
ip.addr!=192.168.1.1
```

- which however will not filter anything! The right expression to use is:

```
!(ip.addr == 192.168.0.1)
```

- The difference between the two lines, even if semantically correct, is that, by using the operator ! = We ask to delete the lines where we have the IP address indicated, but without specifying whether in the source or recipient field.

Current Topic

- Basic security architectural elements
- Traffic interception techniques
- Traffic Analysis tools and technologies
- Aggregated statistic traffic observations

SNMP-based traffic observation

- It is possible to monitor aggregate statistical traffic data of a network through the SNMP protocol
- In the following example, a query is made to a specific element (MIB object) associated with an interface, obtaining information on incoming and outgoing traffic volumes

```
% snmpwalk -v2c -c test 10.106.65.131 1.3.6.1.2.1.2.2.1.16.7 IF-MIB::ifOutOctets.7
= Counter32: 1874894
% snmpwalk -v2c -c test 10.106.65.131 1.3.6.1.2.1.2.2.1.10.7 IF-MIB::ifInOctets.7
= Counter32: 2275304
```

- Observing how traffic volumes vary over time can provide us with information of great interest for the security of a network

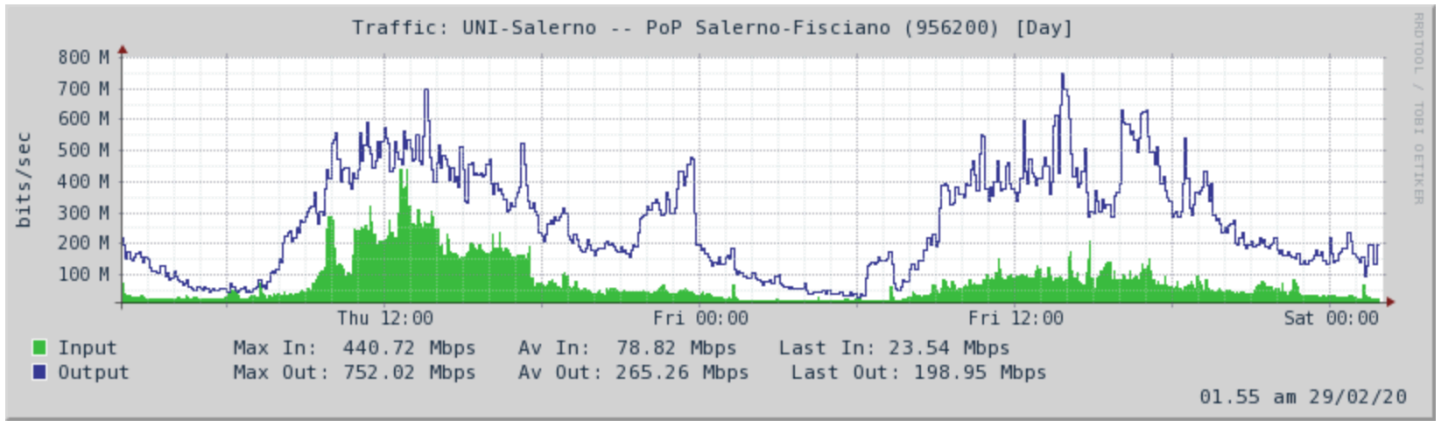
SNMP-based traffic observation

- Tools such as **MRTG** or **CACTI** are responsible for automatically collecting the SNMP bandwidth usage statistics of all the interfaces of the devices present on the network.
- The interface traffic counters are read every 5 minutes (time-driven SNMP reading via cron) and saved on a log file (1 logfile / interface) so that we can obtain:
  - A graphic representation of the throughput
  - A load map allowing us to visualize at at glance the Load Level of all network devices

SNMP-based traffic observation

UNI-Salerno -- PoP Salerno-Fisciano (956200) close

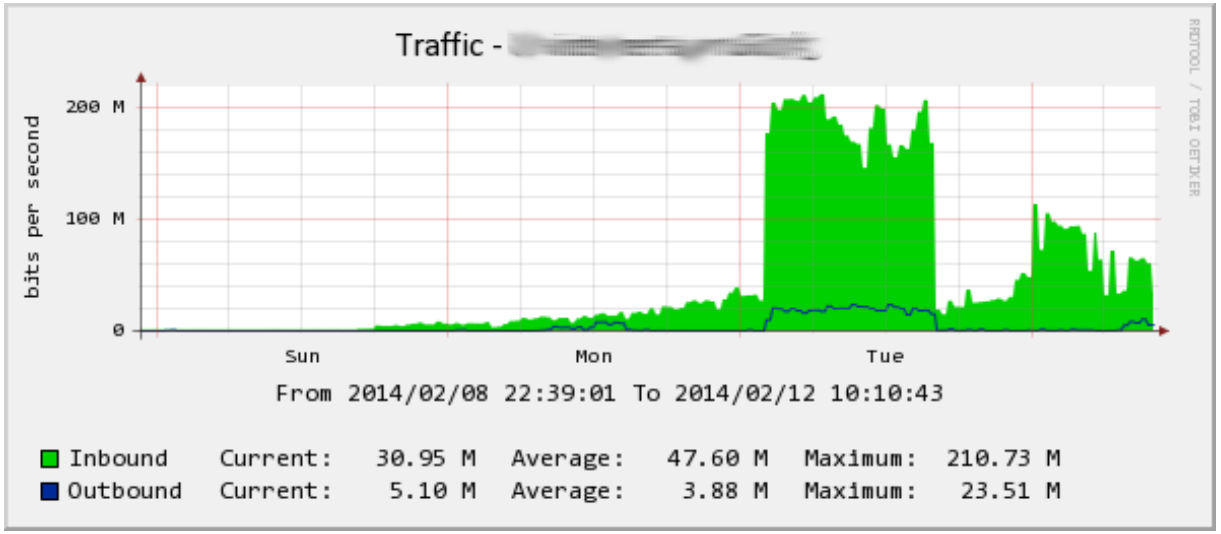
Link name	Use	BW	Side A	Side B	Target options
UNI-Salerno -- PoP Salerno-Fisciano	access	10,00 Gbps	UNI-Salerno 193.204.219.202	PoP Salerno-Fisciano rx1.sa.garr.net (MX480) irb.200 193.204.219.201	



- By observing the traffic trends over time, you can get an idea of the "normal" behavior of a network
- You can easily spot outliers!!

Automatic attack identification

- It is easy to recognize "volumetric" attacks by identifying sustained traffic plafonds that go beyond the behavior normally observed at specific times



- This activity can be easily automated through simple monitoring functions associated with MRTG or CACTI that generate alarms (mail, SMS, etc.) on the basis of exceeding specific traffic thresholds

Acquiring Network and Traffic Information

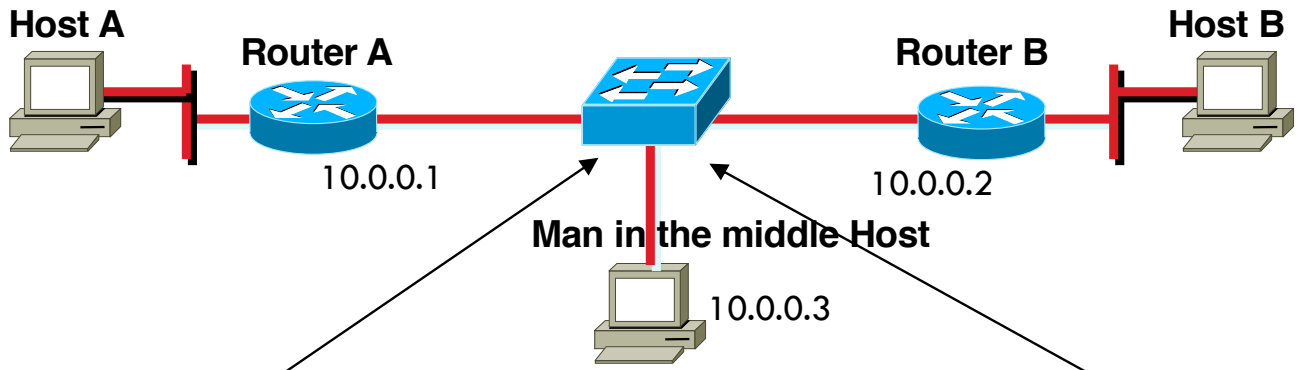
cini Cybersecurity National Lab

Francesco Palmieri
Università di Salerno

fpalmieri@unisa.it
https://docenti.unisa.it/026587/home

Traffic capture through arp poisoning

- Traffic flowing between two networks must be intercepted by a third component (Man in the middle) first through an ARP spoofing attack on the 2 routers and analyzed with tcpdump to capture and examine ftp traffic and HTTP urls

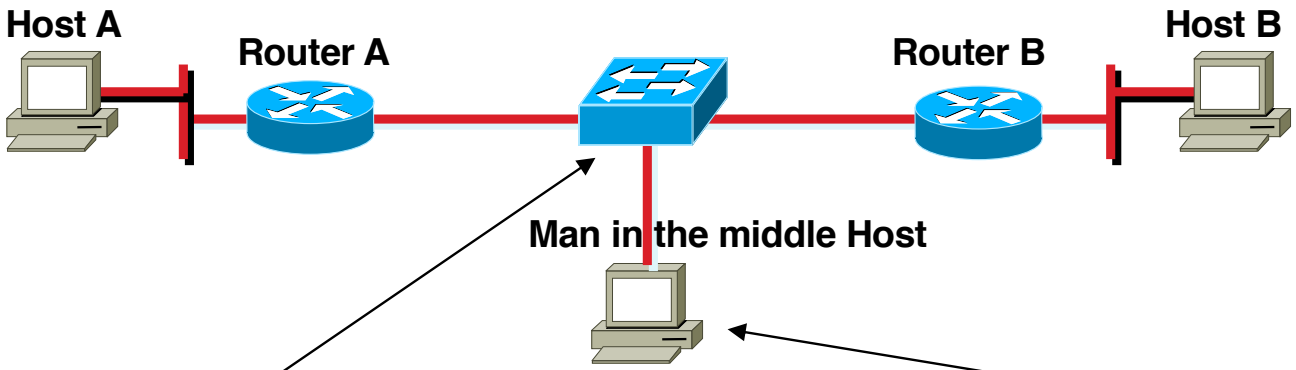



```
echo '1' > /proc/sys/net/ipv4/ip_forward
cat /proc/sys/net/ipv4/ip_forward
```

```
arpspoof -i eth0 -t 10.0.0.1 10.0.0.2 2> /dev/null &
arpspoof -i eth0 -t 10.0.0.2 10.0.0.1 2> /dev/null &
```

Traffic capture through port mirroring

- Traffic flowing between two networks must be intercepted by a third component (Man in the middle) first through the configuration of port mirroring on the link switch and analyzed with tcpdump to capture and examine ftp traffic and HTTP url




```
monitor session 1 source interface fa 1/0 - 2
monitor session 1 destination interface fa 1/15
```

```
tcpdump -n -i eth0 -s 65535 -X
```

CYBER CHALLENGE
CyberChallenge.IT

cini
Cybersecurity National Lab

SPONSOR PLATINUM

accenturesecurity

aizoon AUSTRALIA EUROPE USA
TECHNOLOGY CONSULTING

B5

EY Building a better working world

eni

exprivia | ITALTEL

IBM

KPMG

LEONARDO

NTT DATA
Trusted Global Innovator

NUMERA
SISTEMI E INFORMATICA S.p.A.

Telsy

SPONSOR GOLD

bip.

cisco

MONTE DEI PASCHI DI SIENA
BANCA DAL 1472

negg

NOVANEXT
connecting the future

pwc

SPONSOR SILVER

DiGi ONE
the leading digital company

ICT CYBER CONSULTING