University of Rome Tor Vergata
ICT and Internet Engineering

Network and System Defense

Alessandro Pellegrini, Angelo Tulumello

A.A. 2023/2024

Lecture 3: Ethernet LAN security

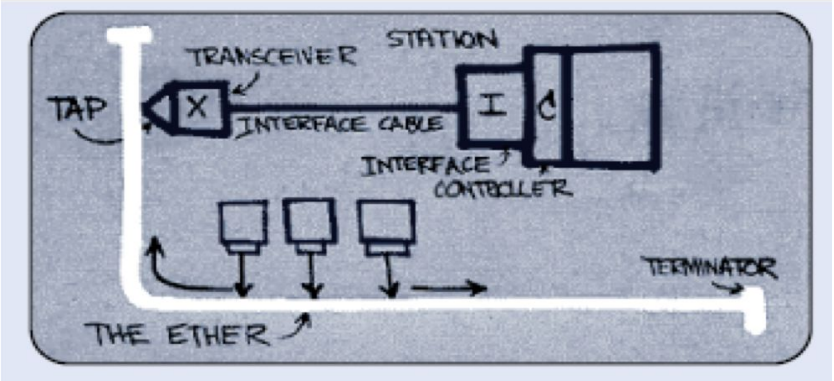
Angelo Tulumello

Slides by Marco Bonola

Ethernet LAN Recap

A little bit of history

- ***1976: Metcalfe, Bogs, et al. write "Ethernet: Distributed Packet-Switching for Local Computer Networks"***
  - Xerox patents the technology
- ***1979: Metcalfe leave Xerox and form 3COM***
  - He shepherds the idea of opening up Ethernet to others and get DEC, Intel, and Xerox to agree to commercialize Ethernet
- ***1980: DIX Ethernet Standard***
  - DIX = DEC-Intel-Xerox vendor consortium
  - Interoperable products from the three founding companies
- ***1982: Xerox relinquishes "Ethernet" trademark***
- ***1985: IEEE 802.3***
  - Ethernet becomes an IEEE 802 standard
    - 10 Mbps (10BASE5 thick coaxial)
    - 802.3 supplement a (1985):
  - 10BASE2 thin coax
  - Minor modifications vs DIX standard
  - Path towards worldwide interoperability



Ethernet standard: the world's FIRST open, multi-vendor standard!
Quoting Metcalfe: "***the invention of Ethernet as an open, non-proprietary, industry-standard local network was perhaps even more significant than the invention of Ethernet technology itself***"

IEEE 802 standards

IEEE 802 is a family of Institute of Electrical and Electronics Engineers (IEEE) standards for local area networks (LAN), personal area network (PAN), and metropolitan area networks (MAN)

The ***IEEE 802.1*** WG focuses on:

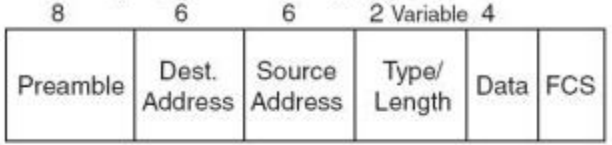
- 802 LAN/MAN architecture
- internetworking among 802 LANs, MANs and wide area networks
- 802 Link Security
- 802 overall network management
- protocol layers above the MAC and LLC layers
- LAN/MAN bridging and management

OSI	***IEEE***				
Data Link Layer	802.2 LLC				
	802.1d (Bridging)				
	802.3 CSMA/CD (Ethernet)	802.4 Token bus	802.5 Token Ring	802.11 Wireless LAN	….
Physical Layer	802.3 CSMA/CD (Ethernet)	802.4 Token bus	802.5 Token Ring	802.11 Wireless LAN	….

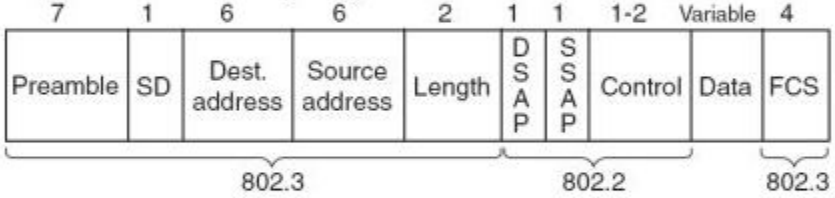
IEEE 802.3	Ethernet	10 Mbps
IEE802.3u	FastEthernet	100Mbps
IEEE802.3ab	Gigabit Ethernet	1000 Mbps
IEEE 802.3ae	10 Gigagbit Ethernet	10 Gbps
IEEE 802.3ba	100 Gigabit Ethernet	100 Gbps

The Ethernet Frame

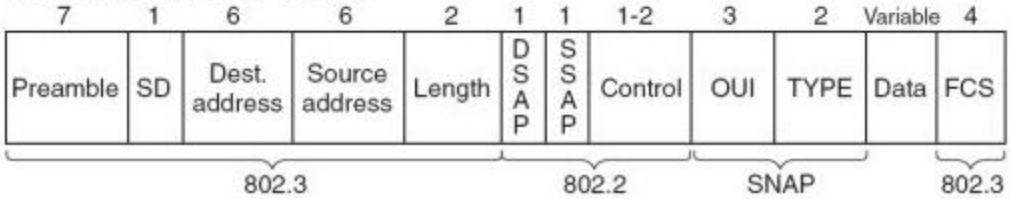
Ethernet (DIX) and Revised (1997) IEEE 802.3



Original IEEE Ethernet (802.3)



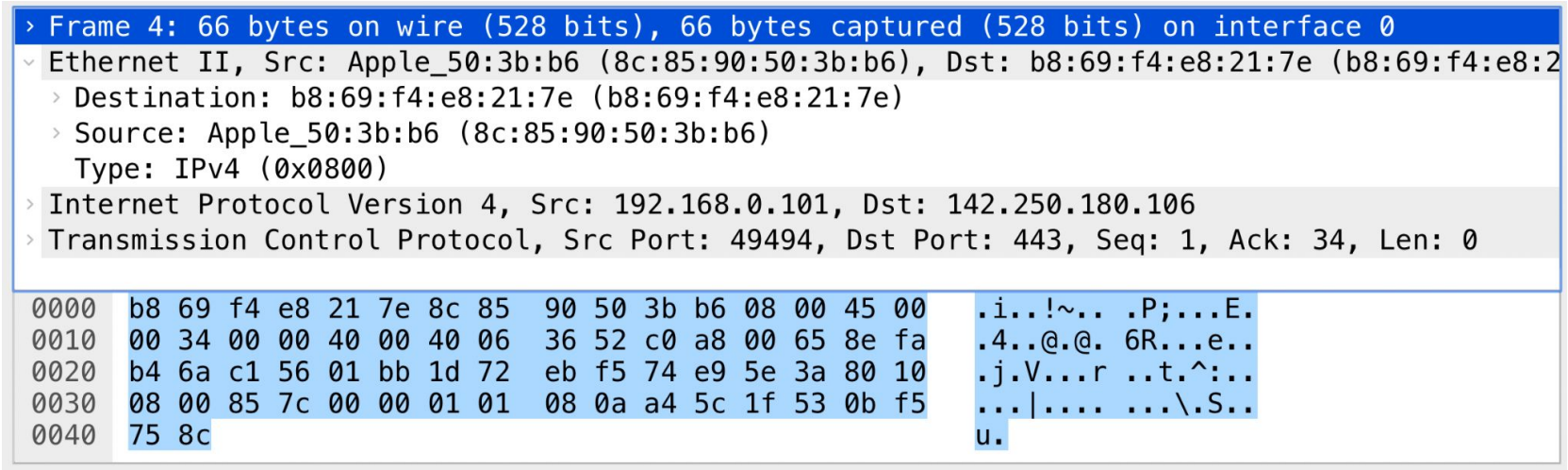
IEEE 802.3 with SNAP Header



Length or protocol?

- In original ethernet: frame type
  - Used for demultiplexing upper layer proto
    - Eg: 0x0800=IP
- In 802.3: length OR type
  - If>1500 (more precisely, ≥ 0x0600 = 1536) → frame type
  - Else → LLC payload size (≤1500)
    - Demultiplexing provided by LLC
    - If <46, remaining octets are PAD (padding)

PCAP traces: legacy format VS LLC header



48 bit addresses

- Typically referred to as
  - Interface address
  - Hardware address
  - MAC address
  - “Ethernet” address (not properly!)
- ***First bit:***
  - 0 = physical address of an interface
    - Unicast address
  - 1 = group address
- ***Second bit:***
  - 0 = globally administered address
    - Assigned by the manufacturer
  - 1 = locally administered address
    - Can be configured through driver

First 24 bits: OUI
(Organization Unique Identifier)
(unique for each vendor)

Typically written in hex
e.g.: F0-11-00-4F-A2-1C

Each byte transmitted
from LSB to MSB

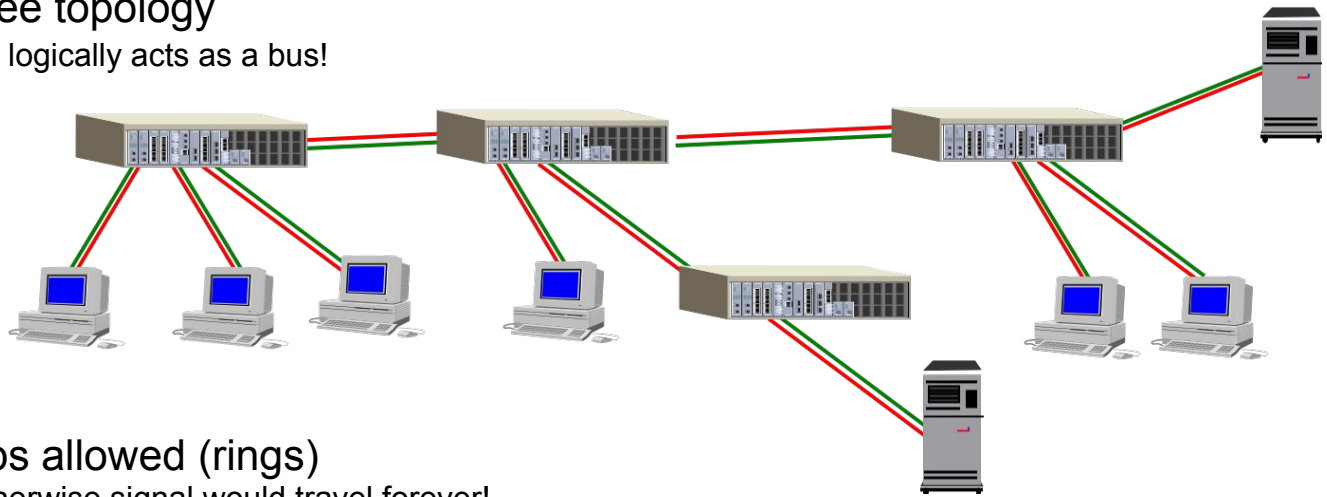
0000.1111.1000.1000.0000.0000.
1111.0010.0100.0101. 0011.1000
0F-11-00-F4-2A-1C

mcast addresses: start with 1
(first octet LSB!)

```
Destination: Apple_50:3b:b6 (8c:85:90:50:3b:b6)
  Address: Apple_50:3b:b6 (8c:85:90:50:3b:b6)
  .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
  .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
```

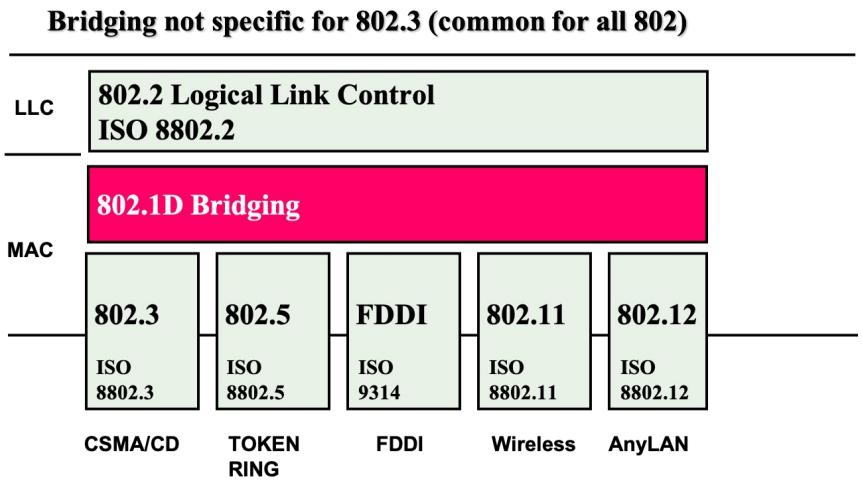
Multiport Repeaters (Hubs)

- Slang name: HUBS
- Essential for BASE-T and BASE-F
- Star / tree topology
  - But logically acts as a bus!



- No loops allowed (rings)
  - Otherwise signal would travel forever!
- Collision domain
  - Maximum propagation distance between end nodes

Bridge/Switches



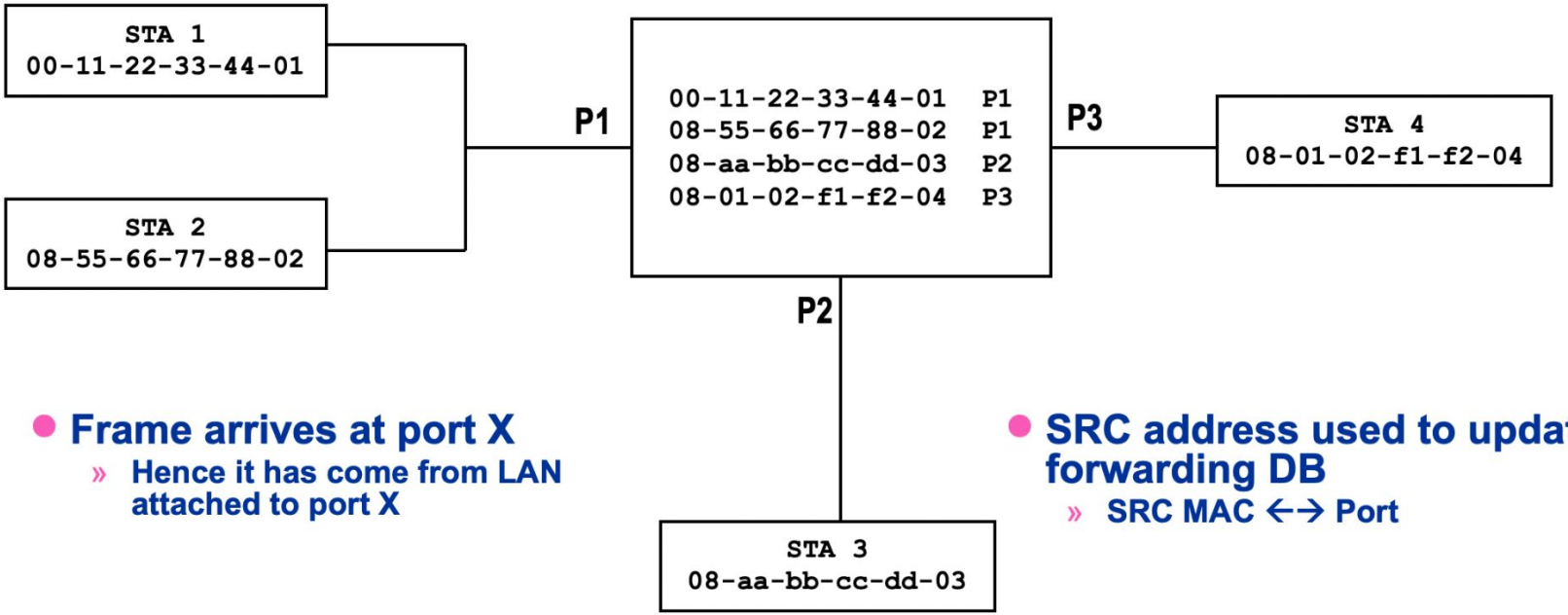
- ***Store & Forward:***
  - read frame (memorize into onboard buffer)
  - Check CRC
    - Discard frame if
      - CRC fails
      - too short (<64 bytes, “runt”)
      - too long
  - Look up destination into forwarding (switching) table
  - Forward packet to outgoing port
- ***Cut-through***
  - Just read first few bytes (until destination address)
  - Don’t do any check
  - Look up forwarding table and select destination
  - forward frame while receiving it

Forwarding database

- Mapping between ***MAC addresses*** and ***ports***
  - Ports: module/port-#
- Static entries:
  - Configured by sysadmin
  - Permanent database
- Dynamic entries:
  - “Learned”
  - Expire after ageing process reaches upper value
  - E.g. 300 seconds
  - configurable

Dest MAC Address	Ports	Age
00-00-08-11-aa-01	1/1	1
00-b0-8d-13-1a-f1	1/7	4
a8-11-06-00-0b-b4	2/3	0
08-01-00-00-a7-64	2/4	1
00-ff-08-10-44-01	2/6	5

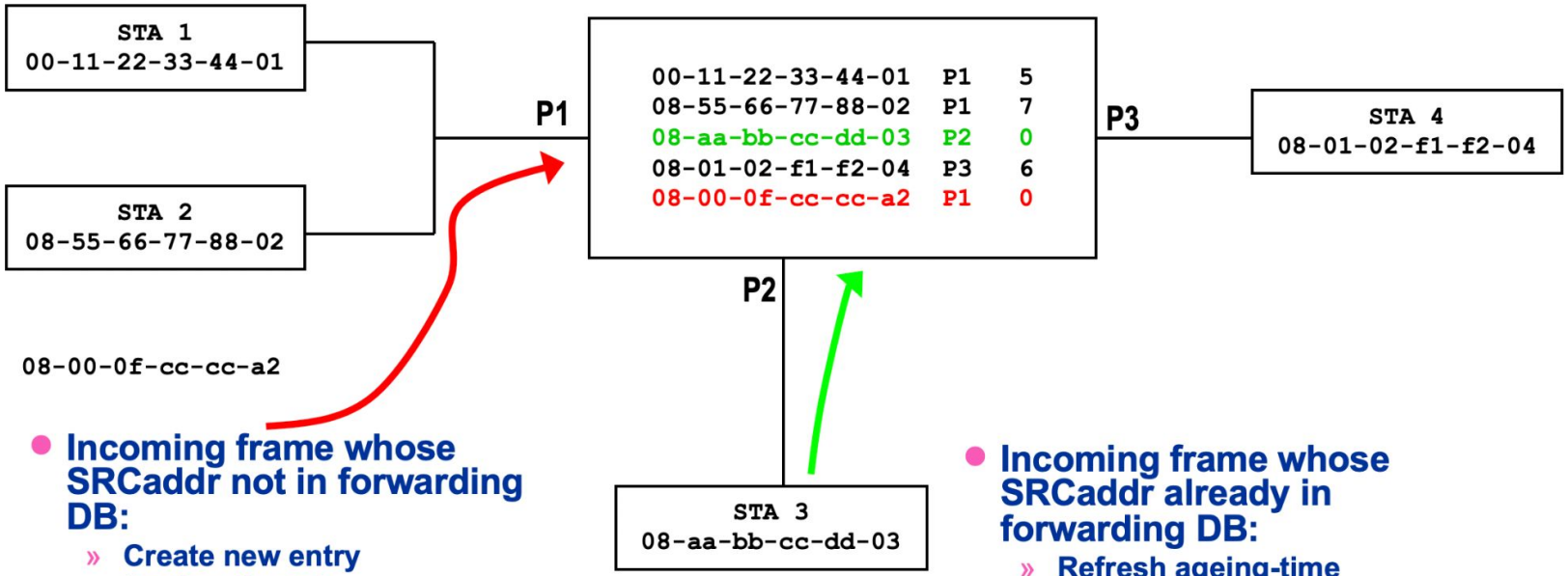
Address Learning (1)



- **Frame arrives at port X**
  - » **Hence it has come from LAN attached to port X**

- **SRC address used to update forwarding DB**
  - » **SRC MAC ←→ Port**

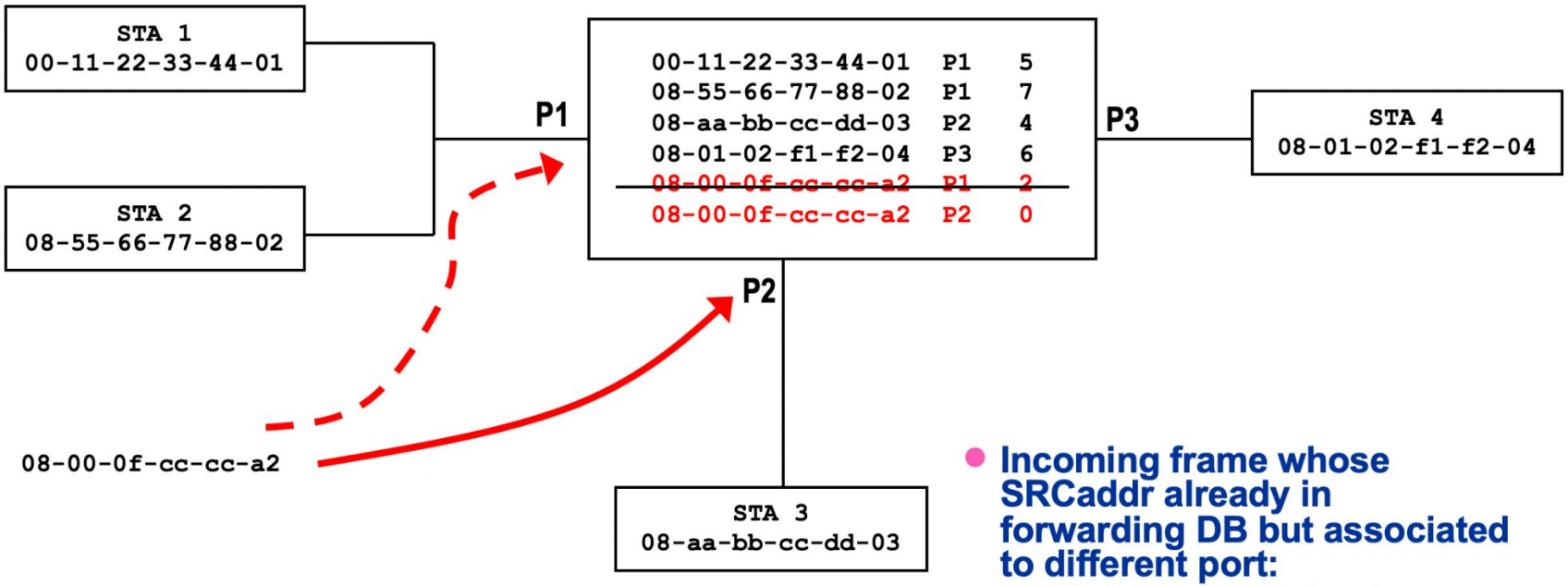
Address Learning (2)



- Incoming frame whose SRCaddr not in forwarding DB:
  - » Create new entry
  - » Ageing-time=0

- Incoming frame whose SRCaddr already in forwarding DB:
  - » Refresh ageing-time
  - » Ageing-time=0

Address Learning (3)



- **Incoming frame whose SRCaddr already in forwarding DB but associated to different port:**
  - **Update associated port**
  - **Refresh ageing time**

Spanning Tree Protocol (STP)

- Redundant links and mesh topologies are common in (large) switched LANs
- ***STP*** is a protocol defined in 802.1d to avoid loops
- STP is based on the flood of management frames from each port (Bridge (B)PDU: ID + cost)
  - *cost is incremented hop by hop*
- If the same BPDU is received on multiple port, only the port with the lowest cost is kept active (tie-break may be needed), the others are deactivated
  - ***loops avoided***
  - ***automatic reconfiguration upon failures***

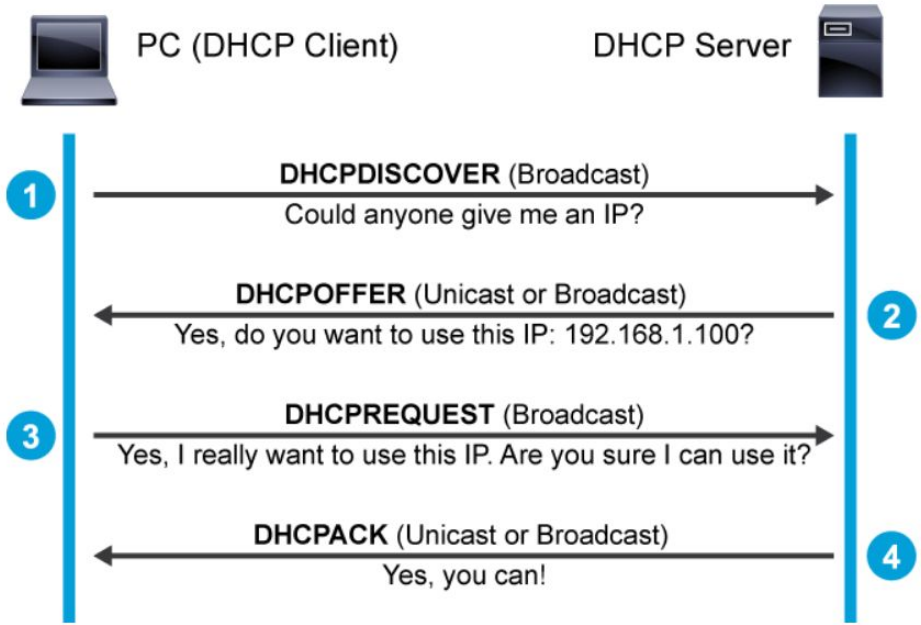
IPv4 adaptation protocols

- Two protocols are needed for the IP version 4 (IPv4) to operate over Ethernet
  - ***Dynamic Host Configuration Protocol (DHCP)***
    - DHCP is used to automatically configure IP-related parameters: IP address/subnet, default GW, DNS resolver, etc…
  - ***Address Resolution Protocol (ARP)***
    - ARP is used to dynamically resolve the MAC address associated with a destination IP address within the same LAN
- IPv6 has similar functions
  - Hosts are found with ***Neighbor Discovery Protocol (NDP)***
  - IPv6 routers are found by listening for multicast ***Router Advertisements***, from which a host can create its own IPv6 address and use Neighbor Discovery to verify its uniqueness
    - IPv6 address stateless autoconfiguration procedure
  - Optionally, ***DHCPv6*** can be used

DHCP basics

- 4 way handshake
  - Discover, Offer, Request, ACK
- Works with multiple DHCP servers on the same LAN (DHCP Release message)
- The Client broadcast (typically at startup) the discover and receive one or more offer from the Server(s) – (then the protocol continues, but we don’t care for now…)
- The Client can Renew (/Rebind) a lease for a previously assigned IP address
- 1 DHCP server for each LAN
- DHCP-Relays allow DHCP communication through routers

DHCP Initialization

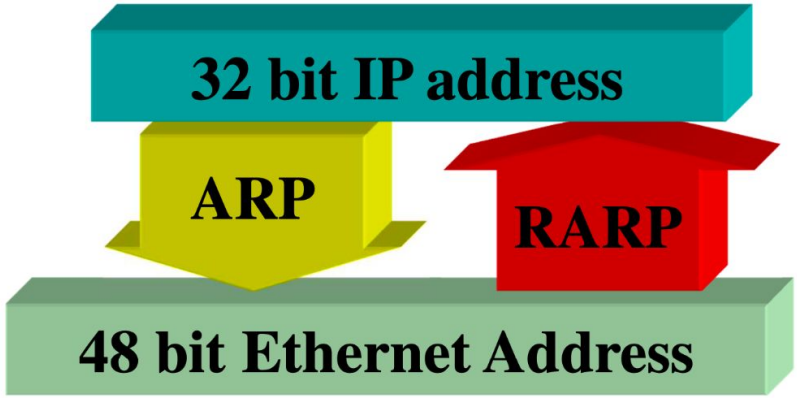


Note: Renewal and Rebinding are 2-way procedures...

ARP basics

- **Dynamic mapping**
  - » not a concern for application & user
  - » not a concern for system administrator!
- **Any network layer protocol**
  - » not IP-specific
- **supported protocol in datalink layer**
  - » not a datalink layer protocol !!!!
- **Need datalink with broadcasting capability**
  - » e.g. ethernet shared bus

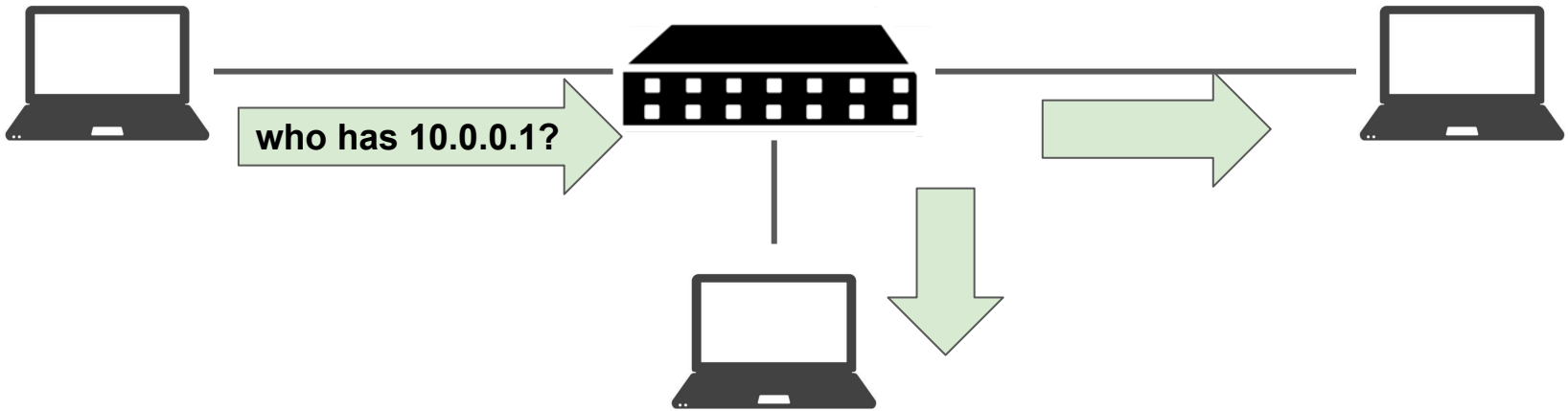
- **Note: ARP NOT STRICTLY NECESSARY!**
  - » May have manual IP←→MAC mapping
    - » Tedious, error prone, requires manual updating E.g. when attaching a new PC must touch all others



ARP: RFC 826
Here described for Ethernet, but valid for more general networks: designed for any datalink with broadcast capabilities

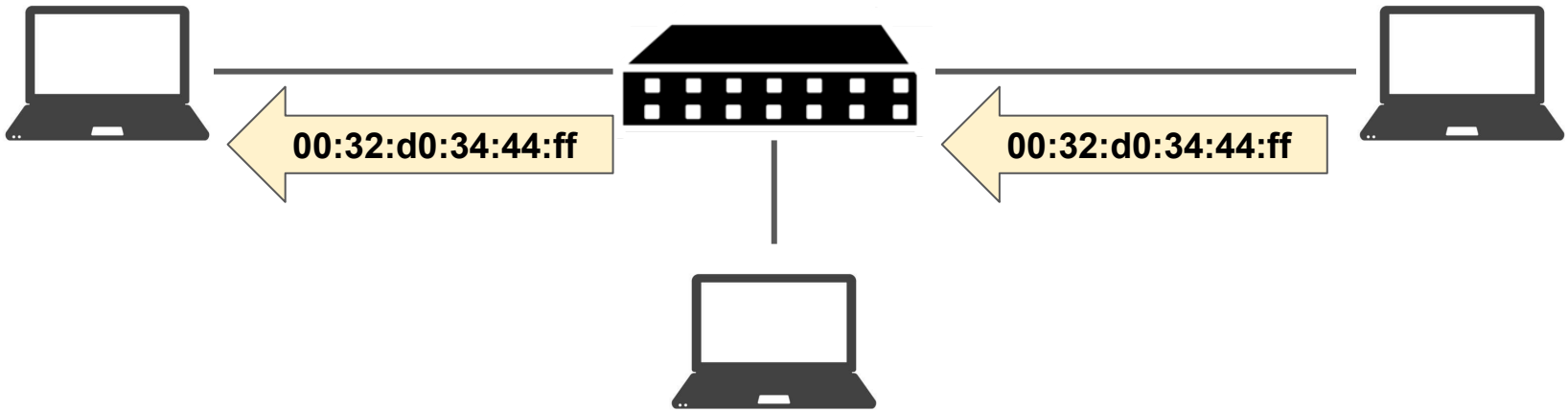
Address Resolution Protocol basics

- ARP is a simple request response protocol
- Request are sent broadcast

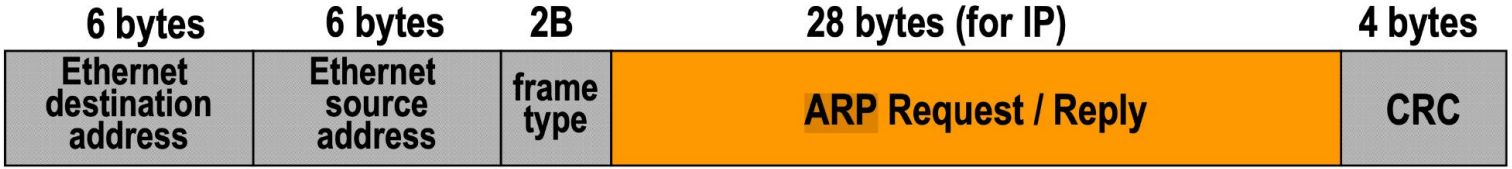


Address Resolution Protocol basics

- ARP is a simple request response protocol
- Request are sent broadcast
- Responses are unicast



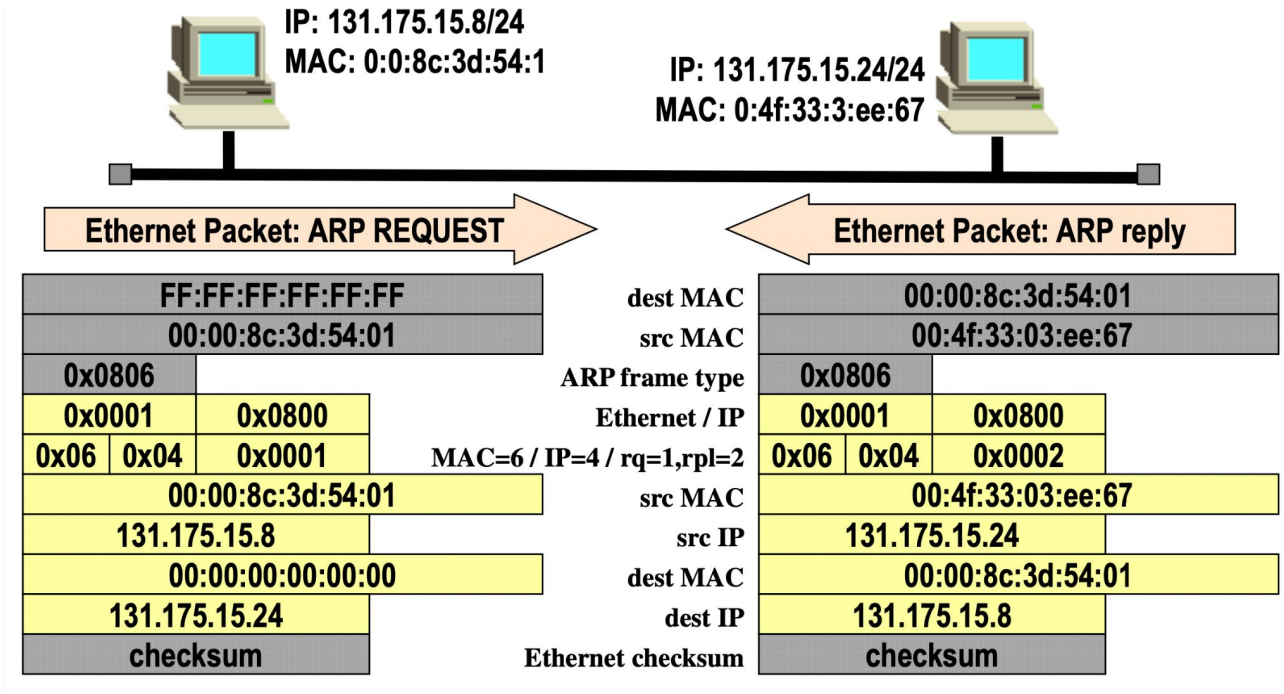
ARP Encapsulation in Ethernet Frame



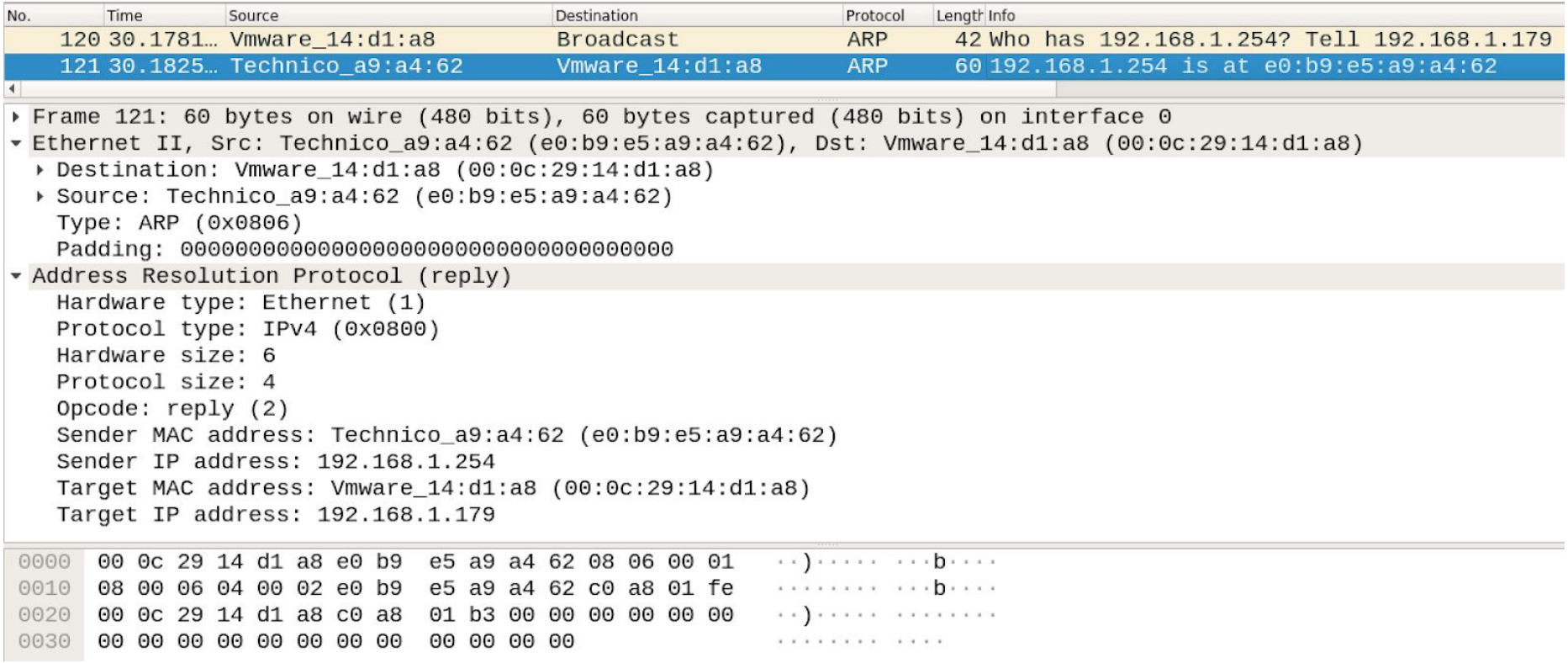
- **Ethernet Destination Address**
  - » ff:ff:ff:ff:ff:ff (broadcast) for ARP request
- **Ethernet Source Address**
  - » of ARP requester
- **Frame Type**
  - » ARP request/reply: 0x0806
  - » RARP request/reply: 0x8035
  - » IP datagram: 0x0800

Protocol demultiplexing codes!

ARP request/reply example



ARP reply capture



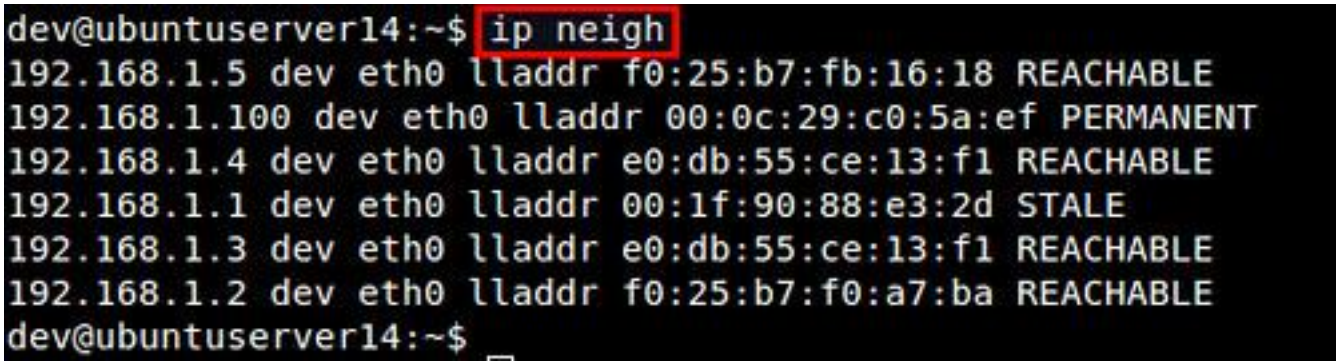
No.	Time	Source	Destination	Protocol	Length	Info
120	30.1781…	Vmware_14:d1:a8	Broadcast	ARP	42	Who has 192.168.1.254? Tell 192.168.1.179
121	30.1825…	Technico_a9:a4:62	Vmware_14:d1:a8	ARP	60	192.168.1.254 is at e0:b9:e5:a9:a4:62

```
▸ Frame 121: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▾ Ethernet II, Src: Technico_a9:a4:62 (e0:b9:e5:a9:a4:62), Dst: Vmware_14:d1:a8 (00:0c:29:14:d1:a8)
  ▸ Destination: Vmware_14:d1:a8 (00:0c:29:14:d1:a8)
  ▸ Source: Technico_a9:a4:62 (e0:b9:e5:a9:a4:62)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Technico_a9:a4:62 (e0:b9:e5:a9:a4:62)
    Sender IP address: 192.168.1.254
    Target MAC address: Vmware_14:d1:a8 (00:0c:29:14:d1:a8)
    Target IP address: 192.168.1.179
```

```
0000  00 0c 29 14 d1 a8 e0 b9  e5 a9 a4 62 08 06 00 01   ··)····· ···b····
0010  08 00 06 04 00 02 e0 b9  e5 a9 a4 62 c0 a8 01 fe   ········ ···b····
0020  00 0c 29 14 d1 a8 c0 a8  01 b3 00 00 00 00 00 00   ··)····· ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····
```

ARP cache

- Avoids arp request for every IP datagram!
  - Entry lifetime defaults to 20min
    - deleted if not used in this time
    - 3 minutes for “incomplete” cache entries (i.e. arp requests to non existent host)
  - it may be changed in some implementations
    - in particularly stable (or dynamic) environments



Virtual LANs (VLANs)

- ***VLANs*** are used to separate a physical network into several logical networks
- Each switch in the network keeps a table associating its ports with the various ***VLAN identifiers*** in use
- The motivation for the VLAN mechanism is to ***increase efficiency*** by limiting the size of the broadcast domain, but it is used also for ***security purposes***
- ***Hosts in different VLANs can not send frames to each other directly***
- When multiple switches are deployed in a VLAN environment, a VLAN ID tagging mechanism is required (e.g. CISCO ISL, 802.3q)

VLANs requires a deeper recap. We'll get back to the VLAN mechanisms in the next class

Ethernet LAN Vulnerabilities

Based on: Kiravuo, Timo, Mikko Sarela, and Jukka Manner. "A survey of Ethernet LAN security." IEEE Communications Surveys & Tutorials 15.3 (2013): 1477-1491.

Security Threats

- Ethernet's (in)security is fundamentally tied to its ***self-configuring nature***
  - features like MAC table learning, STP and ARP together with the underlying broadcasting mechanism are key vulnerabilities
- The basis for attacks is gaining access to the target Ethernet segment
  - The attacker may be an insider with full access rights, may have found an Ethernet connection in a public space, or may have taken control of a workstation using a malware application, or other methods …
- The attacker may utilize the network access for
  - learning about the private network topology and the network traffic for use in a later attack
  - gaining control over switches, routers, or servers in the LAN
  - eavesdropping
  - manipulating information
  - disrupting the availability of the network.
- ***Let's consider the following threat categories:***
  - ***Network and System Access***
  - ***Traffic Confidentiality***
  - ***Traffic Integrity***
  - ***Denial of Service***

Network and System Access (1)

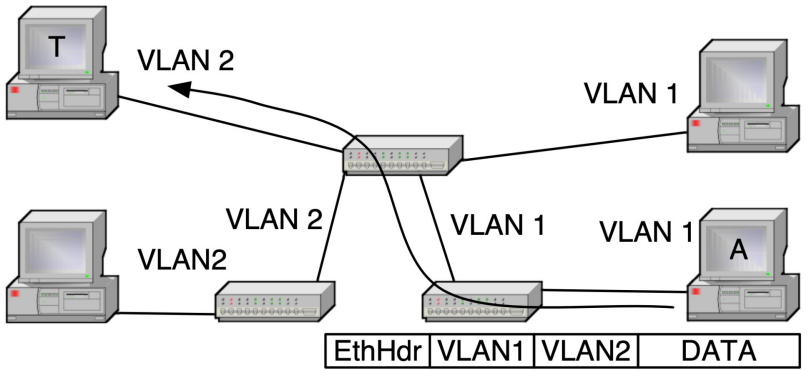
- Access to the network is a prerequisite for attacks and a necessity for all types of attackers
  - Access can be achieved by connecting equipment to the network or by gaining control of existing resources
- ***Unauthorized Joins***: anybody can connect to an Ethernet segment by gaining access to an unconnected port on a switch, by gaining
  - physical access to the switch (if the port is enabled)
  - access to a wall socket (if the port is enabled)
  - removing the cable from a computer and plugging it into another computer
  - plugging in a switch between the existing computer and the socket
- ***Unauthorized Expansion of the Network***
  - The architecture of the Ethernet allows users to expand the network by installing their own switches or wireless access points, which in turn allows other people join the network

Network and System Access (2)

- ***VLAN Join***
  - If a switch listens for VLAN management protocols on host ports, a host can act as a switch and join all VLANs.
- ***VLAN Tagging and Hopping***
  - An attacker can create Ethernet frames that have a VLAN tag and thus inject frames to VLANs to which they are not supposed to have access.

"Double tagging" attack

1. The attacker creates a frame which has the target host's MAC address as the recipient and contains a VLAN 1 tag followed by a VLAN 2 tag
2. The switch strips the tag off and pushes the frame to the trunk link of VLAN 1, where the receiving switch notices the second tag and processes the frame as belonging to the target VLAN
3. NO return traffic capability, but additional spoofing can do this
4. Various attacks can be performed over the unidirectional flow



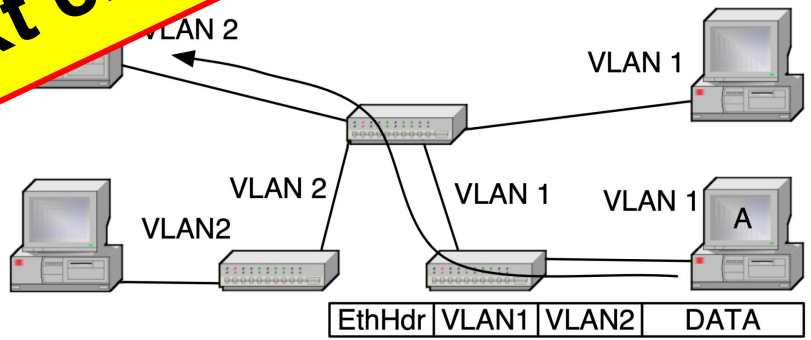
Network and System Access (2)

- ***VLAN Join***
  - If a switch listens for VLAN management protocols on host port… …and join all VLANs.
- ***VLAN Tagging and Hopping***
  - An attacker can create Ethernet frames th… …which they are not supposed to have access.

"Double tagging" attack

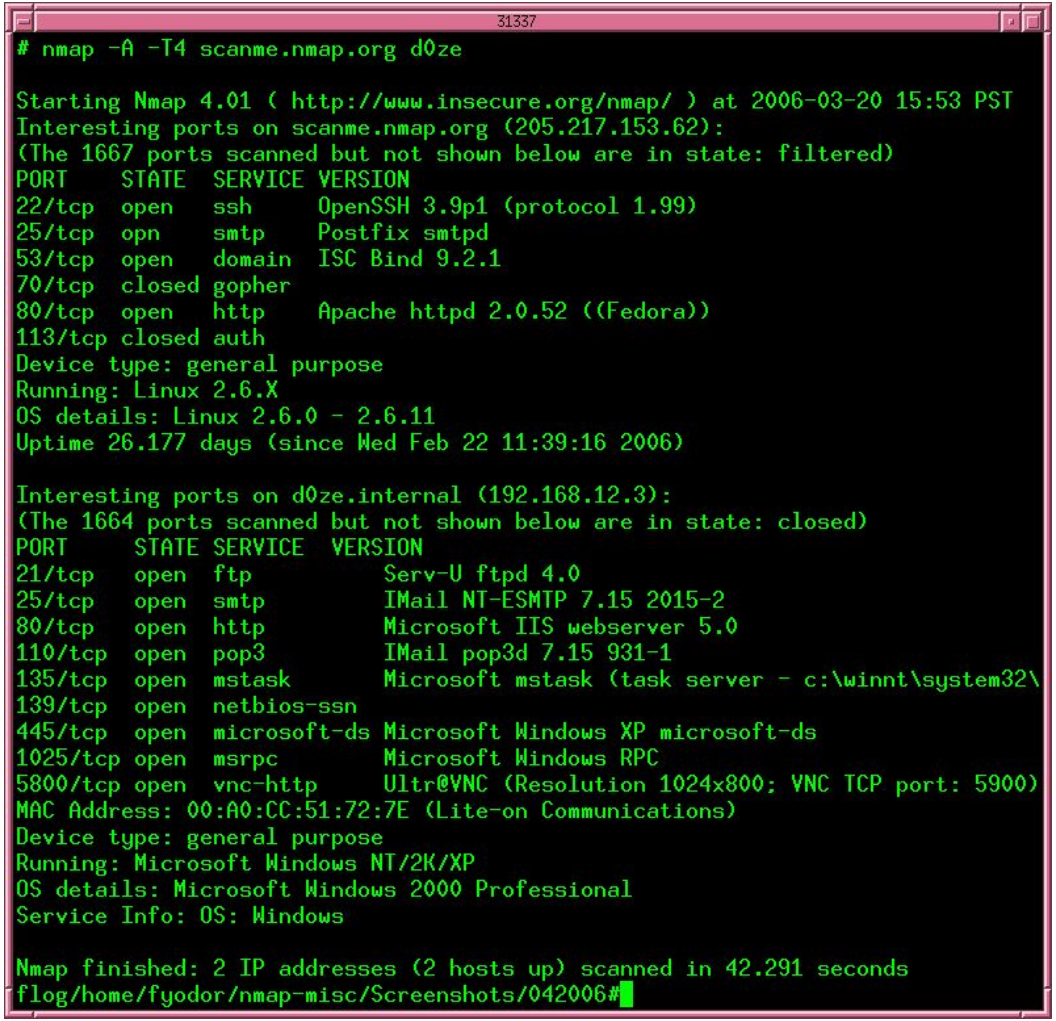
1. The attacker creates a fr… address as the rec… by a VLAN 2…
2. The switc… link of VLA… tag and proc… …second …target VLAN
3. NO return trai… …spoofing can do this
4. Various attacks… …over the unidirectional flow

We'll get back to the VLAN mechanisms and vulnerabilities in the next class



Network and System Access (3)

- ***Remote Access to the LAN:*** Access to an Ethernet segment can be achieved ***by gaining higher layer access to a host*** on the segment
- ***Topology and Vulnerability Discovery:*** An attacker can ***probe the network to find hosts and services in them*** by sending messages and analyzing the replies
  - ***Broadcast ARP*** requests reveal the IP addresses in use and servers or gateways to which other hosts connect to
  - The IP address range in use can be detected from this or the information ***can be requested from the DHCP server***
  - ***This scanning process can be very detailed*** and will reveal plenty of information on hosts and their software, including operating systems, services, and versions, which leads to the identification of potential vulnerabilities


31337
nmap -A -T4 scanme.nmap.org d0ze
Starting Nmap 4.01 (http://www.insecure.org/nmap/) at 2006-03-20 15:53 PST
Interesting ports on scanme.nmap.org (205.217.153.62):
(The 1667 ports scanned but not shown below are in state: filtered)
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 3.9p1 (protocol 1.99)
25/tcp opn smtp Postfix smtpd
53/tcp open domain ISC Bind 9.2.1
70/tcp closed gopher
80/tcp open http Apache httpd 2.0.52 ((Fedora))
113/tcp closed auth
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.0 - 2.6.11
Uptime 26.177 days (since Wed Feb 22 11:39:16 2006)
Interesting ports on d0ze.internal (192.168.12.3):
(The 1664 ports scanned but not shown below are in state: closed)
PORT STATE SERVICE VERSION
21/tcp open ftp Serv-U ftpd 4.0
25/tcp open smtp IMail NT-ESMTP 7.15 2015-2
80/tcp open http Microsoft IIS webserver 5.0
110/tcp open pop3 IMail pop3d 7.15 931-1
135/tcp open mstask Microsoft mstask (task server - c:\winnt\system32\
139/tcp open netbios-ssn
445/tcp open microsoft-ds Microsoft Windows XP microsoft-ds
1025/tcp open msrpc Microsoft Windows RPC
5800/tcp open vnc-http Ultr@VNC (Resolution 1024x800; VNC TCP port: 5900)
MAC Address: 00:A0:CC:51:72:7E (Lite-on Communications)
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows 2000 Professional
Service Info: OS: Windows
Nmap finished: 2 IP addresses (2 hosts up) scanned in 42.291 seconds
flog/home/fyodor/nmap-misc/Screenshots/042006#

Network and System Access (4)

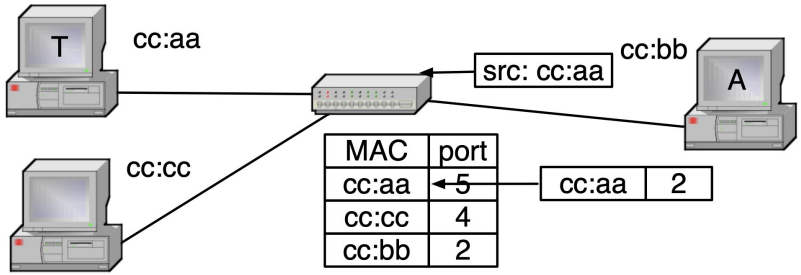
- ***Break-Ins:*** An attacker can use the ***Ethernet network as a medium to attack other hosts and switches on the network***.
  - These attacks typically target vulnerabilities on **higher layer network software**, like the TCP/IP stack and especially server applications. They can lead to the **capture of a host or a switch**, which can be used for further attacks.
- ***Switch Control:*** switches are usually shipped with default or no passwords and the password can usually be physically reset. ***If an attacker gains control of a switch, traffic can be rerouted by switching links down, claiming the STP root by rising the priority of the switch or DoS selected links***.
  - However, as a switch is not a general purpose computer (not always true…), ***its software limits the attacker's ability to eavesdrop*** on the traffic or generate spoofed frames; control of a workstation is needed for these attacks.
  - In cooperation with a connected host the switch can be used to turn on mirroring for eavesdropping and, depending on what management protocols are operational on the network, potentially gain access to any VLAN in use.

Traffic Confidentiality (1)

- Traffic on the network can be useful in itself and also serve the attacker in search of targets. An attacker gains information being transmitted, but also authentication information like passwords and network topology information that can be used for further purposes.
- The original co-axial Ethernet was an easily eavesdroppable bus, where every station received every frame.
- Modern bridged Ethernet **filters most of the traffic** and a host receives only its own traffic, broadcasts, and random frames flooded at the switch after a MAC table timeout.
- ***Passive eavesdropping is possible if an attacker can attach a listening device to a cable connecting a host to a switch or between two switches***. Traffic between hosts can be captured this way. Equipment exists for passively tapping into electrical or optical cabling, or a switch, or multiport computer. Passive eavesdropping is fairly difficult to detect.
- If a switch does not know where to forward a frame, it floods it out of all of its ports. With software ***an attacker can easily generate enough frames with random addresses to overwrite an entire MAC table and make the switch flood all data frames to all ports for eavesdropping***
- On most switches this MAC flooding attack affects all VLANs, even if the attack originates within one VLAN.

Traffic Confidentiality (2)

- Sending a frame with a forged sender address ***overwrites the correct entry in the MAC table*** and redirects traffic to the attacker
  - overloaded switch (forwarding DB full) "hub" mode
  - more later on...
- This MAC spoofing attack becomes more useful, if the real owner of the MAC address can be disabled or is known to be off-line
  - Otherwise a race condition exists between the two hosts and traffic will flipflop between them. If the real host can be made to go off-line on demand, the spoofing host may not only receive traffic intended to the target host, but take over existing sessions of higher layer protocols



- Many switches have a port mirroring feature to support diagnostics or intrusion detection systems. If the attacker has control of a switch, this may be activated.

Note on MAC address spoofing

- MAC can be changed in locally generated packets as well as in packets generated by other stations
  - ethernet layer does not implement any secure integrity check (only a CRC for TX errors)
  - ethernet does not implement any authentication mechanism that binds the MAC address in the packets to the MAC address configured on the NIC
- ***Locally generated packets***
  - ***1: change the MAC address of the NIC***
    - `# ip link set dev $interface address XX:XX:XX:XX:XX:XX`
  - ***2: raw socket programming***
    - `PF_INET, PF_PACKET`
    - high level libraries like `python scapy`
  - ***3: in-kernel programming***
    - e.g. eBPF/XDP
- ***"intercepted" packets***
  - MAC implementations silently discard frames addressed to other MAC address (except for multicast Ethernet address)
  - We can configure the NIC into promiscuous mode (i.e. to not perform any mac-based filtering at firmware level)
  - All further non-Ethernet processing is up to your application. Anyway OS Kernel usually filters these packets. Still need for low level socket programming

```
#...includes and defines omitted ...
int main() {
	int sockFd = 0, retValue = 0;
	char buffer[BUFFER_LEN]={0}, dummyBuf[50]={0};
	struct sockaddr_ll destAddr;
	short int etherTypeT = htons(0x8200);
	unsigned char localMac[6] = {0x00, 0x08, 0xA1, 0x8E, 0xE4, 0x52};
	unsigned char destMac[6] = {0x00, 0x17, 0x9A, 0xB3, 0x9E, 0x16};
	memset(&destAddr, 0, sizeof(struct sockaddr_ll));
	if((sockFd = socket(PF_PACKET, SOCK_RAW, htons(ETH_P_ALL))) < 0) {
		printf("ERROR! socket() call failed (Error No: %d \"%s\").\n", errno, strerror(errno));
		exit(1);
	}
	destAddr.sll_family = htons(PF_PACKET);
	destAddr.sll_protocol = htons(ETH_P_ALL);
	destAddr.sll_halen = 6;
	destAddr.sll_ifindex = 2;
	memcpy(&(destAddr.sll_addr), destMac, MAC_ADDR_LEN);

	memcpy(buffer, localMac, MAC_ADDR_LEN);
	memcpy((buffer+MAC_ADDR_LEN), destMac, MAC_ADDR_LEN);
	memcpy((buffer+(2*MAC_ADDR_LEN)), &(etherTypeT), sizeof(etherTypeT));
	memset(dummyBuf, 0xa0, sizeof(dummyBuf));
	memcpy((buffer+ETHERTYPE_LEN+(2*MAC_ADDR_LEN)), dummyBuf, 50);

	if((retValue = sendto(sockFd, buffer, 64, 0, (struct sockaddr *)&(destAddr), sizeof(struct sockaddr_ll))) < 0) {
		printf("ERROR! sendto() call failed (Error No: %d \"%s\").\n", errno, strerror(errno));
		exit(1);
	}
	return(0);
}
```

Traffic Integrity (1): MAC flooding

- In a typical ***MAC flooding attack***, a switch is fed many Ethernet frames, each containing different source MAC addresses, by the attacker. The intention is to consume the limited memory set aside in the switch to store the MAC address table
- The effect of this attack may vary across implementations, ***however the desired effect (by the attacker) is to force legitimate MAC addresses out of the MAC address table, causing significant quantities of incoming frames to be flooded out on all ports.*** It is from this flooding behavior that the MAC flooding attack gets its name
- After launching a successful MAC flooding attack, a malicious user can use a packet analyzer to capture sensitive data being transmitted between other computers, which would not be accessible when the switch is operating normally
- The attacker may also ***follow up with an ARP spoofing attack*** which will allow them to retain access to privileged data after switches recover from the initial MAC flooding attack

Traffic Integrity (2): ARP and DHCP Poisoning

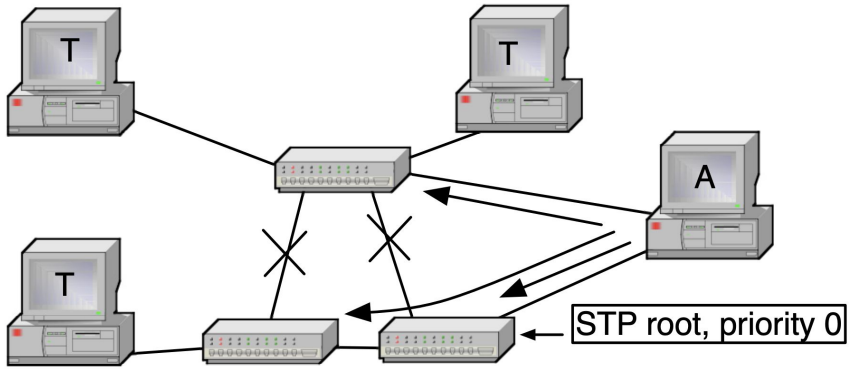
- ARP is a stateless protocol and most operating systems will ***accept ARP replies even when not requested***
  - hosts tend to send these gratuitously whenever a link goes temporarily down
- This enables a host to capture traffic intended for another host just by sending an ARP message to the sender with the intended receiver's IP address and the attacker's MAC address
- In a similar way, an attacker can ***detect broadcast DHCP server requests and race the server to reply them first***
  - upon success the attacker can assign a gateway (router) and DNS servers to the target host, along with its IP address, and control the host's traffic at will.

ARP poisoning

- Based on the transmission of malicious unsolicited ARP responses proposing a wrong mapping between IP addresses and MAC addresses
- Some OSes trust such responses even without a pending request
- Other OSes requires that a request for that address is pending
- In both cases is pretty easy to implement this attack
  - with third party programs (e.g. ettercap)
  - with ad-hoc scripts/programs
- Consequences:
  - it is easy to impersonificate a victim in the same LAN at IP level
  - it is easy to realize a Man in the Middle attack

Traffic Integrity (3): Man in The Middle

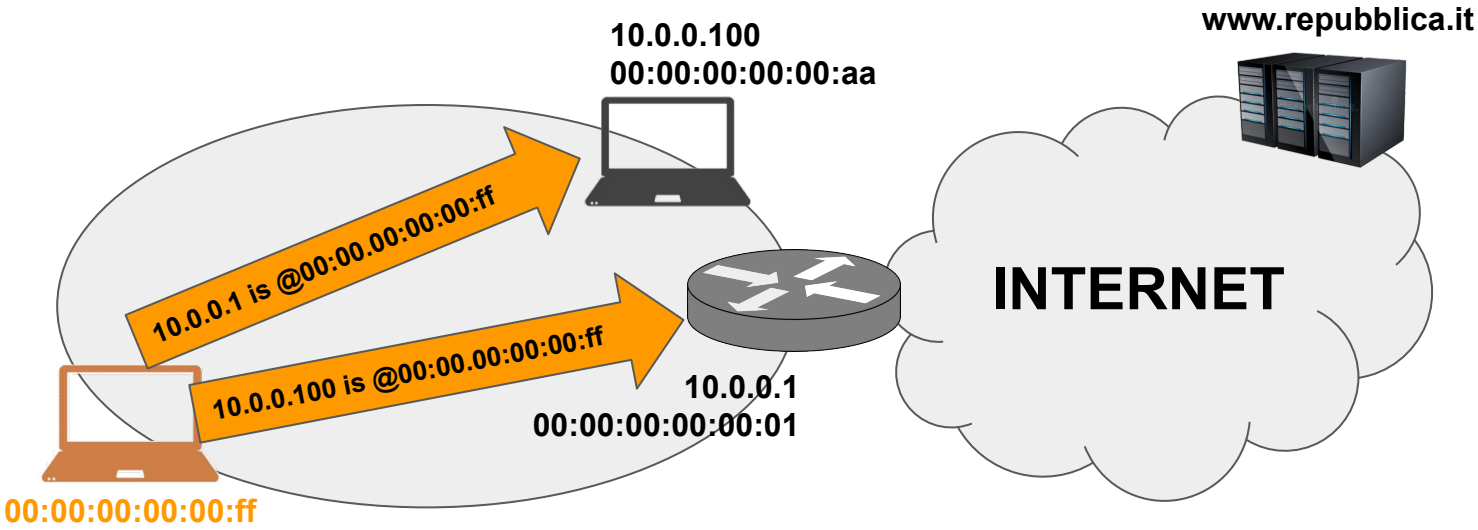
- If an attacker can direct traffic to pass through himself and that traffic is not protected by an integrity verification mechanism, the attacker can easily eavesdrop or modify the traffic
- These ***Man in the Middle (MITM)*** attacks against higher layer protocols are performed relatively easily on an Ethernet segment.
- IP being the most common higher layer protocol on Ethernets, ***the previously mentioned ARP and DHCP poisoning attacks can be deployed to redirect traffic to go through the attacker's host for modification or just eavesdropping***
- On the Ethernet layer this is harder
  - it can be done using STP. If a host is connected to two switches, it can act as the ***Root Bridge in the STP environment*** and create a tree topology, where part of the traffic goes through this host.
  - it can be done with a double port stealing attack (not easy, victims send packets...)



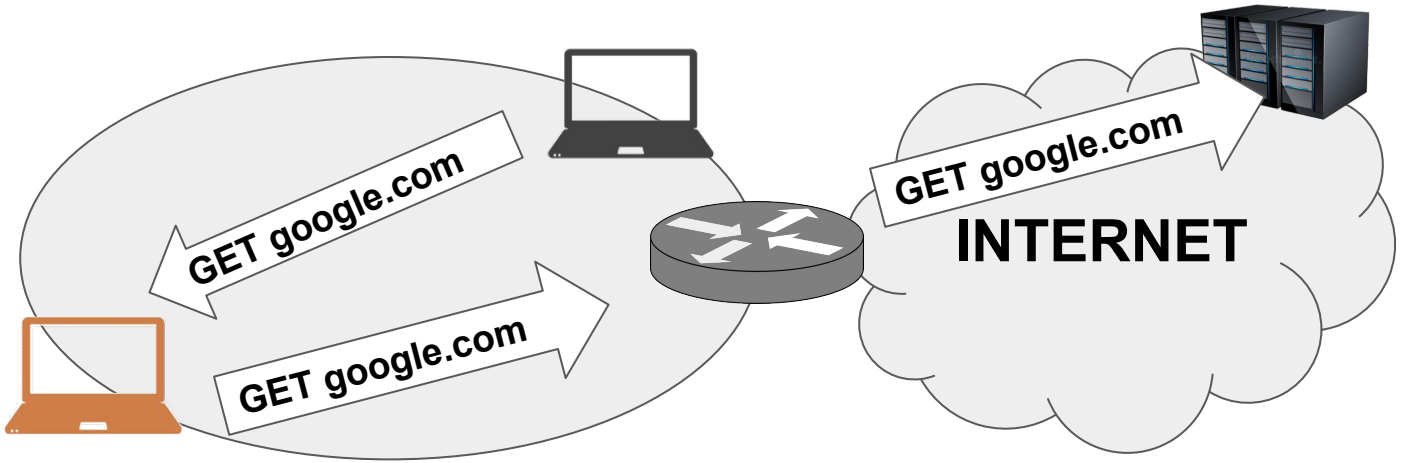
Traffic Integrity (3): Man in The Middle with ARP poisoning

- Reference scenario
  - Attaker, victim, default GW in the same LAN
- Attacker send 2 spoofed ARP responses in loop
  - IP_addr_GW @ MAC_attacker (to the host)
  - IP_addr_host @ MAC_attacker (to the GW)
- As soon as these bidings are injected in the local ARP caches, the attacker becomes the MiTM between host and GW
- and since GW is the host's default GW to the internet, all the traffic sent/received to/from the internet is intercepted by the attacker

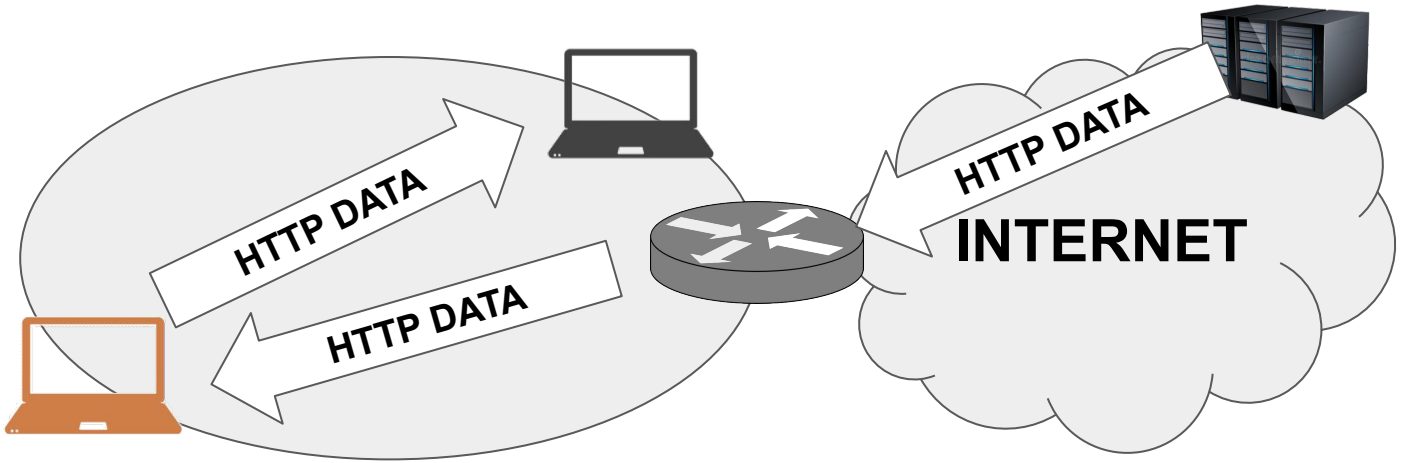
Traffic Integrity (3): Man in The Middle with ARP poisoning



Traffic Integrity (3): Man in The Middle with ARP poisoning



Traffic Integrity (3): Man in The Middle with ARP poisoning



MiTM with python/scapy

```
#! /usr/bin/env python
import sys
from scapy.all import *
import time

ip_victim="10.0.0.100"
ip_router="10.0.0.1"
hw_attacker="00:00:00:00:00:FF"
hw_victim="00:00:00:00:00:AA"
hw_router="00:00:00:00:00:01"

arp_to_victim = Ether(src=hw_attacker, dst=hw_victim)/ARP(op=2, psrc=ip_router,
pdst=ip_victim, hwsrc=hw_attacker, hwdst=hw_victim)

arp_to_router = Ether(src=hw_attacker, dst=hw_router)/ARP(op=2, psrc=ip_victim,
pdst=ip_router, hwsrc=hw_attacker, hwdst=hw_router)

if not arp_to_victim or not arp_to_router:
    exit()

while (True):
    sendp(arp_to_victim)
    sendp(arp_to_router)
    time.sleep(1)
```

Traffic Integrity (4): Session Hijacking

- Ethernet is a stateless protocol, but ***many higher level protocols create a sessio***n.
- Once a session is set up, it is often assumed to be trusted and no further traffic verification is made.
- If an attacker can eavesdrop on, or otherwise gain enough information about a session (IP addresses, TCP ports and sequence numbers, and application data, like an HTTP authentication cookie), ***the attacker can re-create the session and act like one endpoint***
- ***If one endpoint of the session can not be diverted, it might partake in the communications and disrupt the sessions.***
- Gratuitous ARP can be used to direct the local endpoint's traffic to a bogus MAC address and incoming traffic from the gateway router to the attacker's host. One endpoint can be silenced with a DoS attack. With the correct timing, a session may be brought up to date with the correct application messages or by trusting TCP to discard packets that appear to be duplicates based on the sequence number.

Traffic Integrity (5): Replay

- A message eavesdropped earlier can be sent again.
- As the message is not modified, it can be authenticated or encrypted by the original sender without affecting the attack – the attacker just needs to guess at the content of the message to consider whether it is worth resending.
- Within the Ethernet domain useful messages to resend would be small, stateless control messages that fit within one frame.
- Typical messages for targeting a resend attack could be routing notifications or SNMP “set” or “trap” messages.

Denial of Service

- The attacker's motivation for DoS ***is not to gain access to data but to prevent its use***
- The attacks can cause total loss of service or degradation of service
- **@Layer 1:** cutting links physically or damaging the circuitry with electricity (obvious)
- **@Layer 2:** attacks can cause much more damage
  - ***Resource Exhaustion Attacks***
    - target the control and management planes of a switch by sending frames that require additional processing and handling (log, VLAN configuration)
    - Unknown unicast flooding → BROADCAST
      - same as MAC flooding, but the intention is to congest the network and success depends on being able to cause sufficient traffic
  - ***Protocol Based DoS***
    - The STP that makes a tree out of a mesh network is designed to be self-configuring
    - An attacker that controls a node on the network can send STP messages and pretend to be a switch.
    - The whole switching network can be brought to halt by flooding it with STP control messages

Ethernet vulnerability countermeasures

Intro

- Ethernet's lack of security has been solved by defining any Ethernet segment as unsecure and requiring it to be placed inside a protected domain
  - E.g.: behind a firewall in a secure building with trusted staff.
- Higher layer cryptographic solutions are used to solve the remaining issues
- When looking for security in the Ethernet layer itself, it is clear that the switches form the core of the solution
  - A major problem is that a switch has no way of knowing if each of its ports is connected to: one computer (a host); a host with several virtual hosts (and virtual MAC addresses); a hub; a silent switch (that does not talk STP and other topology revealing protocols); a regular switch; or a switch that has other switches behind it.
  - This dynamic ambiguity makes the issue challenging
- 4 categories: ***Router Based Security, Access Control, Secure Protocols, Security Monitoring***

Router Based Security

this is somehow obvious and not always applicable...

- ***Replacing one central Ethernet switch with an IP router affects security***
- ***An IP router partitions the rest of the Ethernet network into several segments***
  - Each new segment is a separate broadcast domain
    - ARP, STP, VLAN, and MAC address table based attacks are no longer possible between the segments.
  - Inside the segments the same attacks remain feasible, unless each switch is replaced with a multiport router.
  - The traffic between segments thus becomes impossible to eavesdrop on from other segments or to be redirected for a MITM attack. Ethernet's MAC headers are dropped at the router and traffic is guided by the IP addresses and router's IP table.
- ***The router blocks Ethernet's control plane protocols*** (ARP and STP) and DHCP
  - The router also prohibits easy mobility (unless additional protocols are enforced e.g. Mobile IP)
  - A host may move in the Ethernet network and keep its IP and MAC addresses, the MAC address tables in switches are updated automatically.
- ***A router also splits the broadcast domain. Autodiscovery protocols are blocked*** (if no support in the router)
- Compared to an Ethernet switch an IP router provides a considerable amount of protection against other users connected to the same router (but then of course you have other vulnerabilities...)

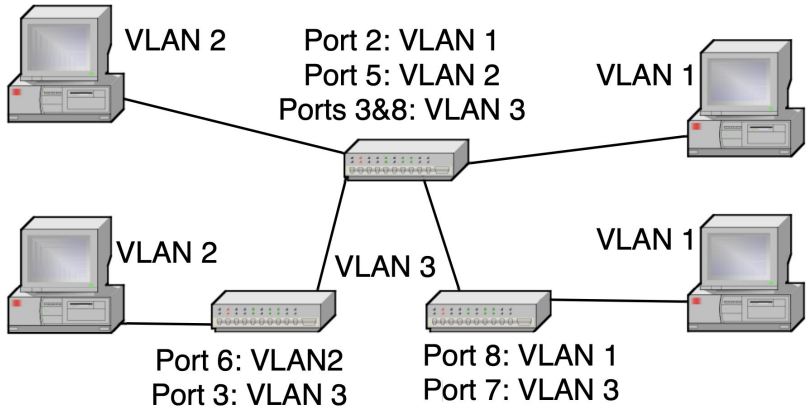
Access control

- An attacker needs access before being able to perform any attacks
- Untrusted entities can be kept out by limiting access to the network or requiring authentication.
- Limiting the access capabilities of trusted entities reduces the threat potential even further.
- 5 strategies: ***1) Physical Protection of the Network***; ***2) Segmentation and VLANs; 3) Individual VLANs; 4) Authentication Based Access Control; 5) Access Control Lists;***
- ***Physical Protection of the Network***
  - Network equipment can be located in locked cabinets and racks and wiring installed inside walls to prevent unauthorized access.
  - However, access is needed for the network to be useful and physical protection is of limited value
- ***Segmentation and VLANs***
  - Limiting the size of an Ethernet segment limits the area vulnerable to attacks.
  - A segmentation method external to Ethernet would be a higher layer device, such as a router or firewall.
  - Inside Ethernet the IEEE 802.1Q virtual LAN mechanism provides a way to limit broadcasts and other traffic to specific segments.

Access control: VLAN segmentation example

Switches are configured to assign VLANs 1 and 2 to specific ports and use VLAN 3 as a trunk.

- Hosts on VLAN 1 are not able to reach hosts on VLAN 2 on layer 2.
- Vendors recommend using VLANs for security.
- However, VLAN based security depends on proper switch configuration and vendor documentations also note that the default settings of switches are not secure, thus enabling, e.g., VLAN hopping



Access control: Authentication Based Access Control

- ***IEEE 802.1X port authentication*** supports several types of authentication credentials, such as a user-name and password pair, or a certificate and corresponding private key.
  - 802.1X requires supporting client software in the end host, software in the switch, and a centralized authentication database server.
  - The host communicates with the switch and the switch verifies the authentication from the database
- 802.1X uses Extensible Authentication Protocol (EAP) that has broad support for different types of authentication methods and structures (cryptographic exchanges related to certificates are more complex than just supplying a user-name and password)
- 802.1X authenticates a host at the beginning of a session, attaching the MAC address to a specific port in the switch
- 802.1X capable switches provide protection from MAC spoofing and flooding attacks
  - however, ARP poisoning and other attacks remain possible.
  - An attacker may place a hub or switch between the authenticated host and the authenticating switch. After authentication the authenticated host can be disconnected without losing the electrical connection to the authenticating switch and another host, configured with same MAC address, be used in the network.
- Authentication can also be used between switches to form a trusted inner network
  - This can be used to prevent attacks where a host acts as a switch.

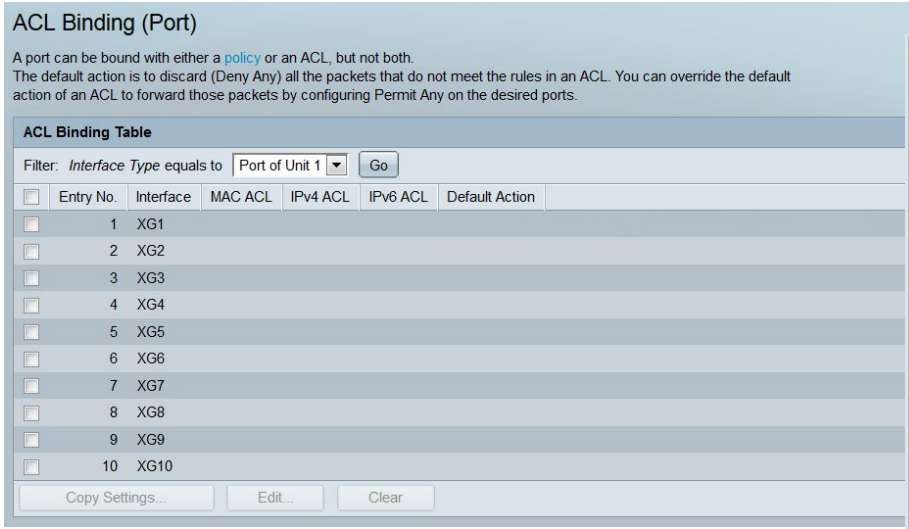
Access control: Authentication Based Access Control

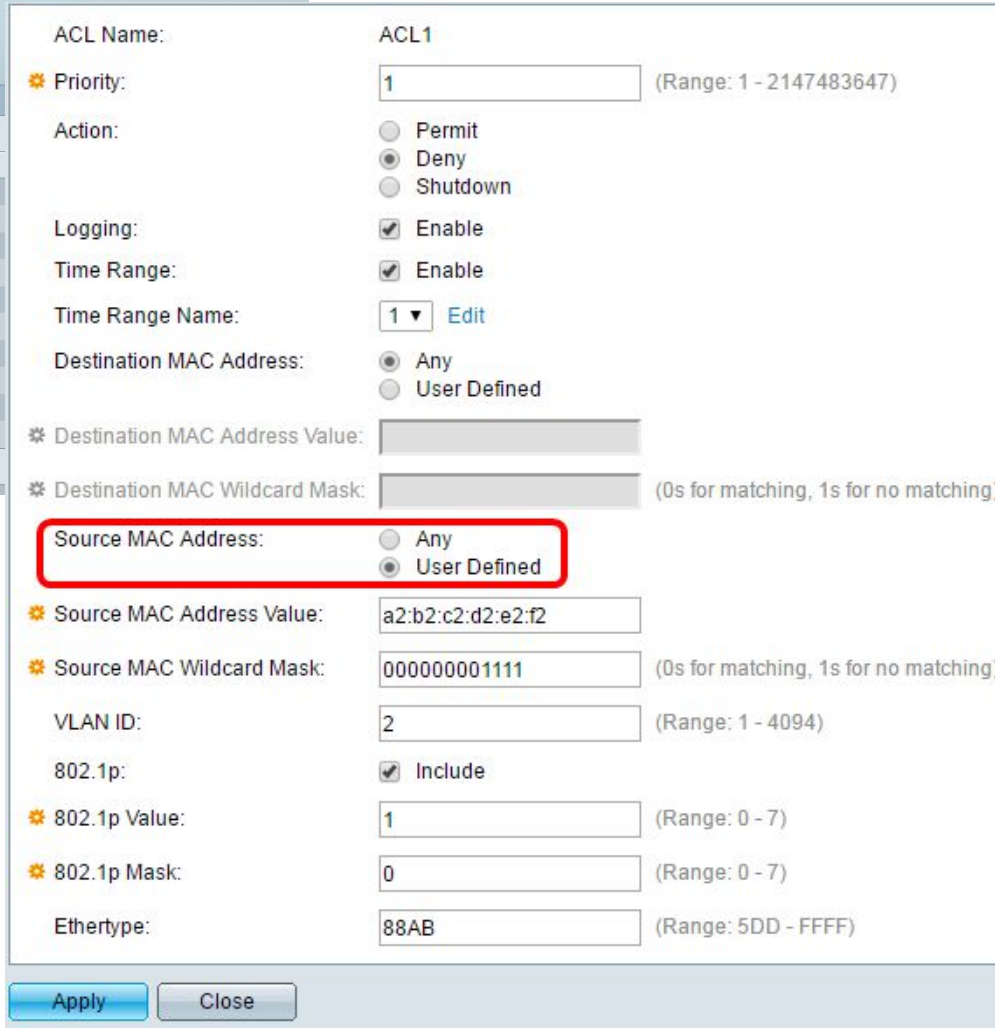
- ***IEEE 802.1X port authentication*** supports several types of authentication credentials, such a user-name and password pair, or a certificate and corresponding private key.
  - 802.1X requires supporting client software in the end host, software in the authentication database server.
  - The host communicates with the switch and the switch verifi
- 802.1X uses Extensible Authentication Protocol (EAP) th ation methods and structures (cryptographic exchange g a user-name and password)
- 802.1X authenticates a host at specific port in the switch
- 802.1X capab ng attacks
  - authenticated host and the authenticating switch.
    an be disconnected without losing the electrical
    itch and another host, configured with same MAC address, be
- Aut e used between switches to form a trusted inner network
  - used to prevent attacks where a host acts as a switch.

more about 802.1x in a dedicated class

Access control: Access Control Lists (ACLs)

- ***ACLs are not part of the Ethernet specification***
- The Ethernet frame does not have many features: for a simple Ethernet frame ACL the usable attributes are the sender's or receiver's MAC address or the Ethertype field
  - advanced switches permit to filter upper layer protocols field (CISCO L3 switches)
- ***Port security lets the administrator limit access to a port in a switch, based on the number of MAC addresses***
  - This blocks MAC flooding and can make it more difficult to expand the network by adding switches without authorization. Typically the functionality has detailed control features.
  - Besides just blocking new MAC addresses when their number exceeds a limit, port security may also be set to block a port from existing MAC addresses.

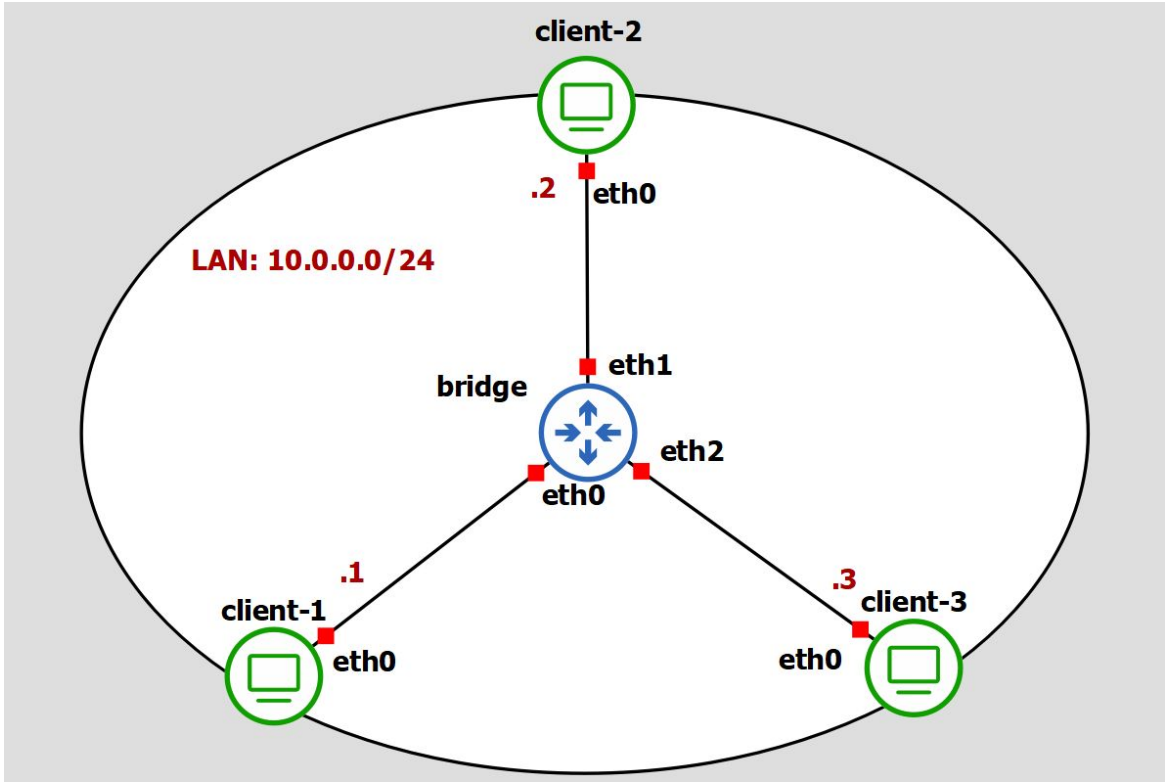


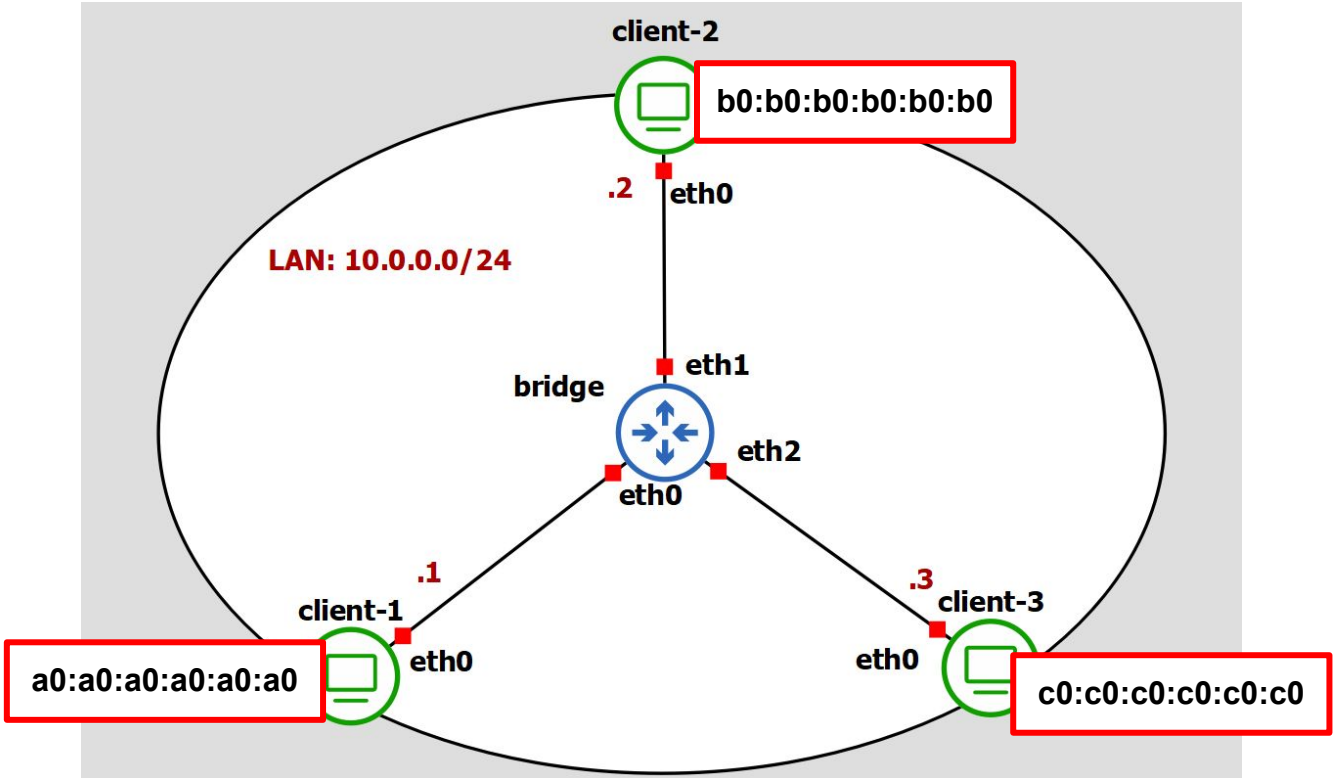


CISCO Port ACL binding to a MAC ACL

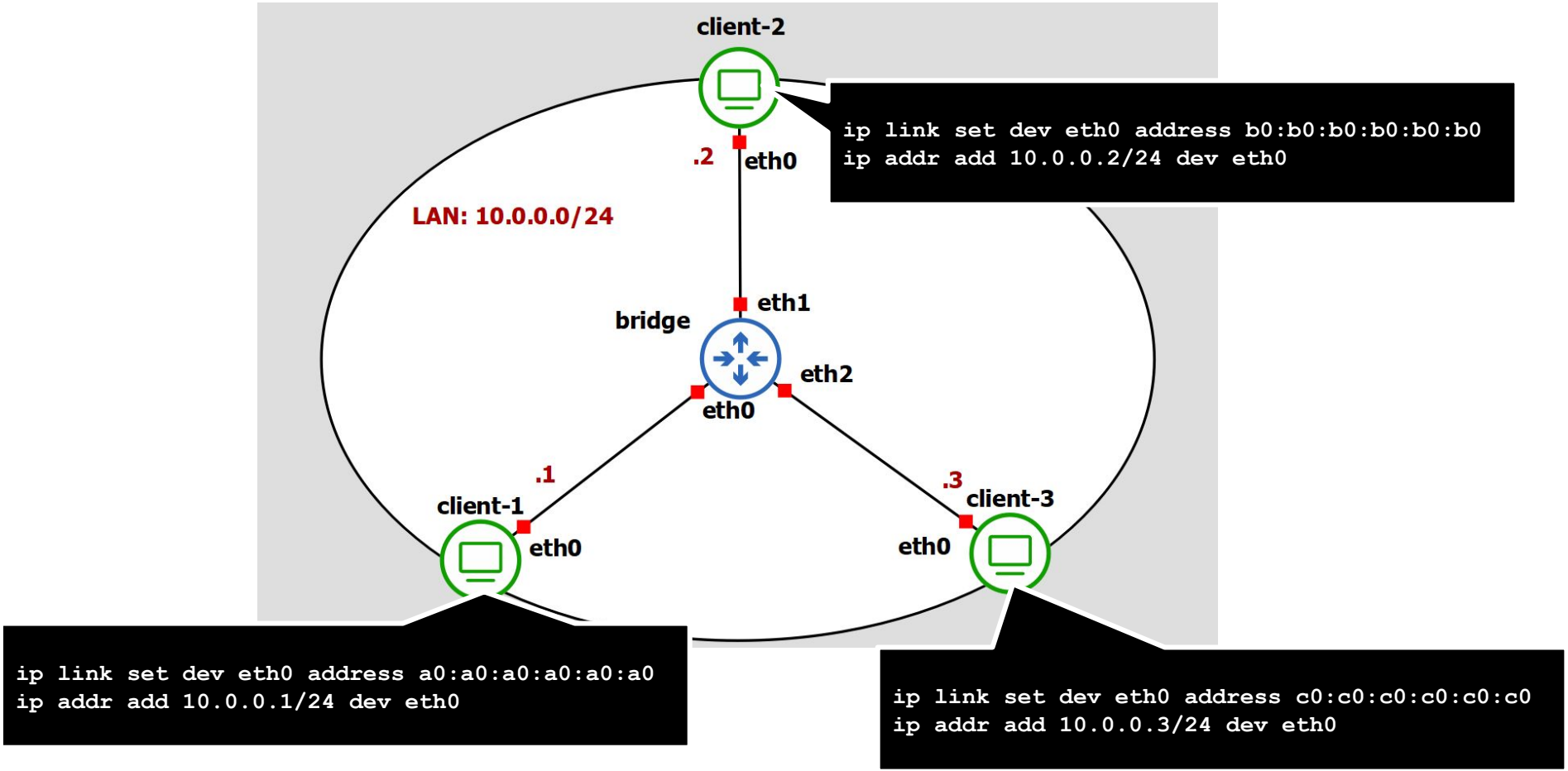
Lab 2: ACLs with Linux Bridge and ebtables

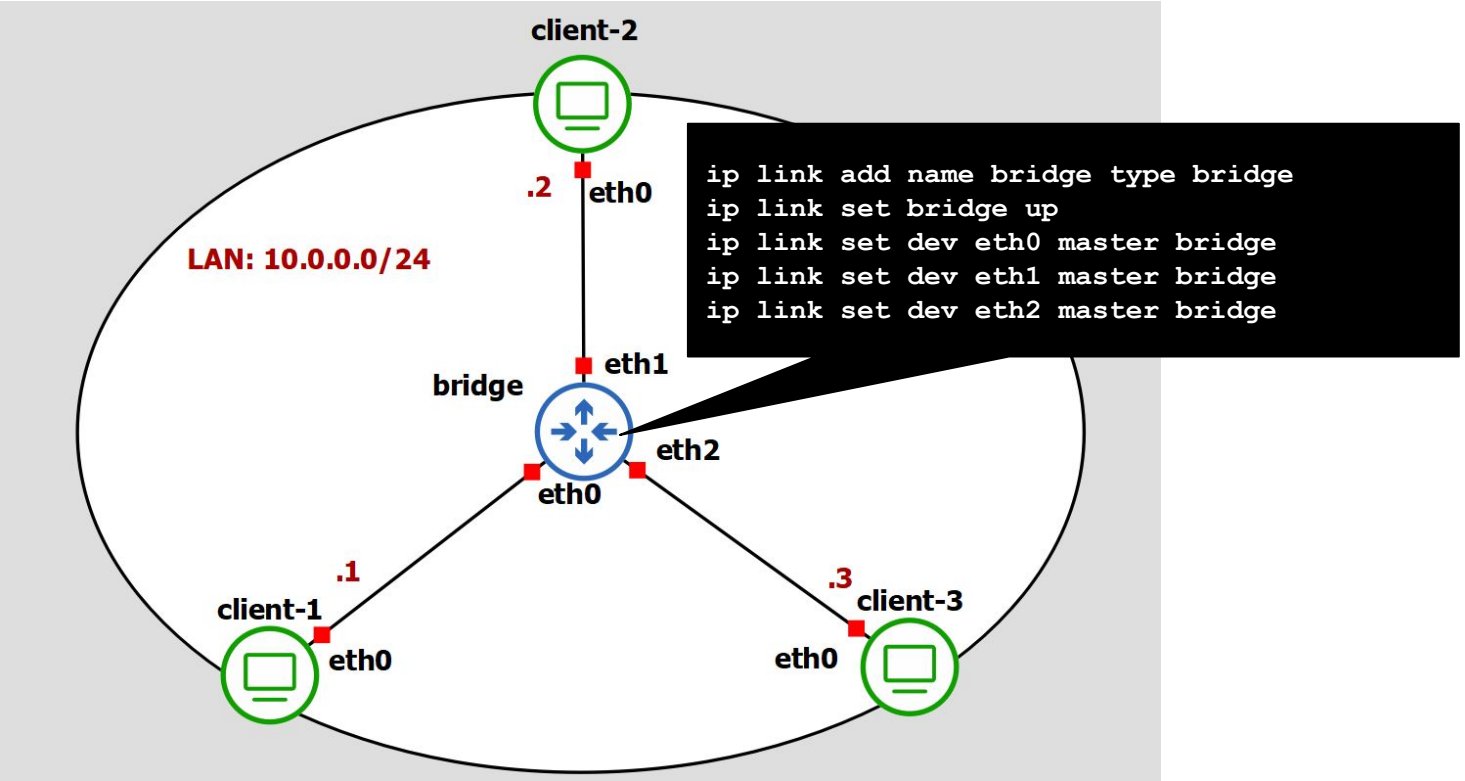
this is a very simple lab which requires a few configuration lines. Anyway, it is a good opportunity to play with our emulation environment together


client-2
.2
eth0
LAN: 10.0.0.0/24
eth1
bridge
eth2
eth0
.1
client-1
eth0
.3
client-3
eth0

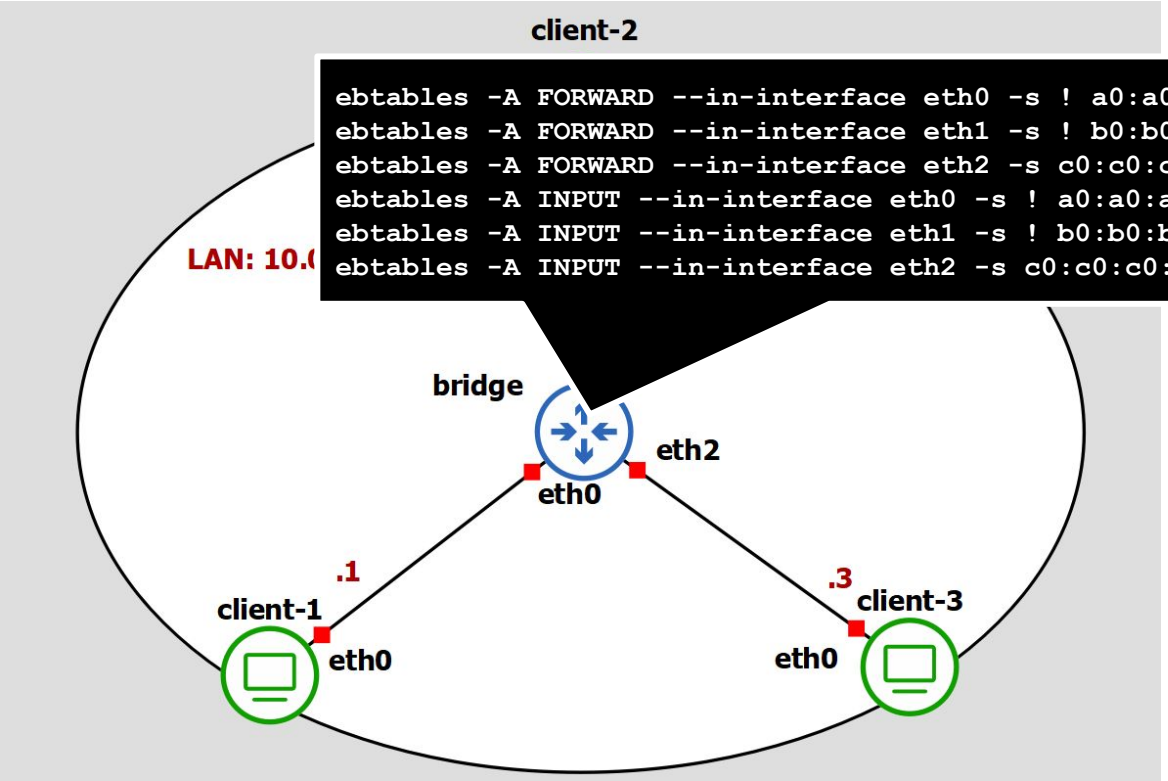


GOAL: accept incoming packets only from known MAC addresses according to the following mac:port binding table: ***eth0*** → a0:a0:a0:a0:a0:a0 (client-1); ***eth2***→ b0:b0:b0:b0:b0:b0 (client-2); ***eth3*** → whatever address != client3 (c0:c0:c0:c0:c0:c0). Verify that client 3's incoming packets are dropped


client-2
.2
eth0
LAN: 10.0.0.0/24
bridge
eth1
eth2
eth0
.1
client-1
eth0
.3
client-3
eth0
ip link set dev eth0 address b0:b0:b0:b0:b0:b0
ip addr add 10.0.0.2/24 dev eth0
ip link set dev eth0 address a0:a0:a0:a0:a0:a0
ip addr add 10.0.0.1/24 dev eth0
ip link set dev eth0 address c0:c0:c0:c0:c0:c0
ip addr add 10.0.0.3/24 dev eth0



basic configuration. eth{0,1,2} are enabled. all clients can communicate with each others

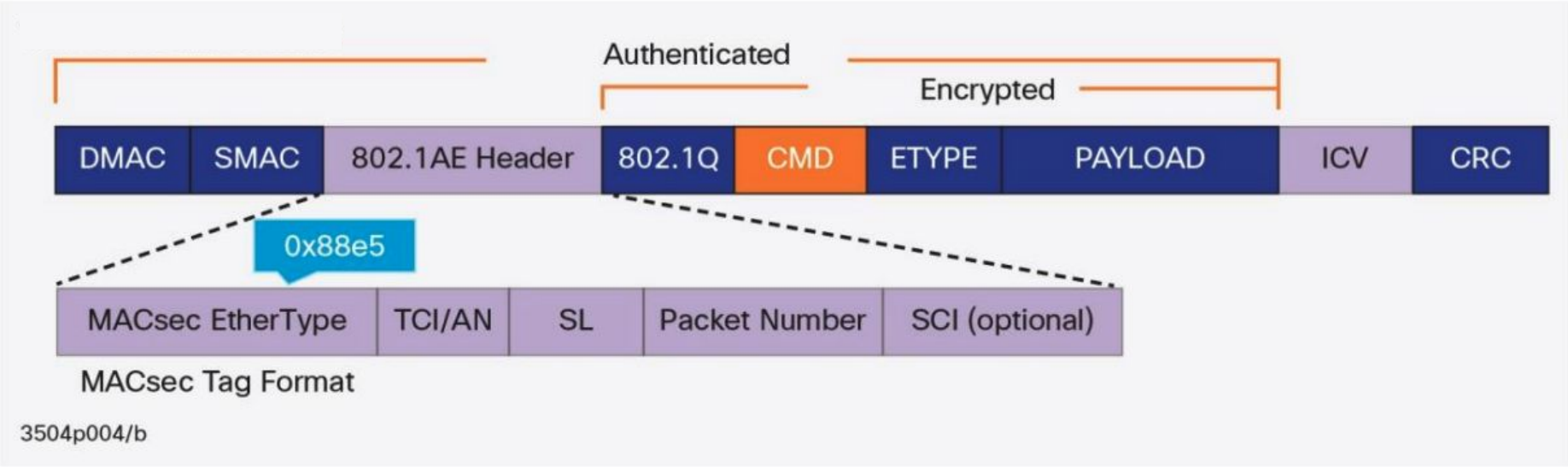


ACL are implemented with the NETFILTER framework. The rule must be listed under the specific category of rule: iptables corresponds to IPv4, ip6tables corresponds to IPv6, ebtables corresponds to either IPv4 or IPv6 depending on the rule, or just layer2 if no IP is specified within the rule.

Secure Protocols (1): MACsec

- Cryptography can solve integrity and confidentiality requirements.
- ***IEEE 802.1AE*** (also known as **MACsec**) is a network security standard that operates at the medium access control layer and defines connectionless data confidentiality, integrity and replay protection
- The standard defines
  - ***MACsec frame format***, which is similar to the Ethernet frame, but includes additional fields:
    - Security Tag, which is an extension of the EtherType
    - Message authentication code (ICV)
  - ***Secure Connectivity Associations***: groups of stations connected via unidirectional secure channels
  - ***Security Associations within each secure channel***. Each association uses its own key (SAK). More than one association is permitted within the channel for the purpose of key change without traffic interruption
  - ***A default cipher suite*** of GCM-AES-128 (Galois/Counter Mode of AES cipher with 128-bit key)
    - GCM-AES-256 using a 256 bit key was added to the standard 5 years later.
- ***Key management is outside the scope of 802.1AE, but is specified by 802.1X-2010***
  - we'll see in the 802.1x lecture
- MACsec provides confidentiality, data integrity and replay protection of data frames
  - *However, authorized hosts may misbehave*

IEEE 802.1AE frame format



nice short reading with further details:
https://www.cisco.com/c/dam/en/us/td/docs/solutions/Enterprise/Security/MACsec/WP-High-Speed-WAN-Encrypt-MACsec.pdf

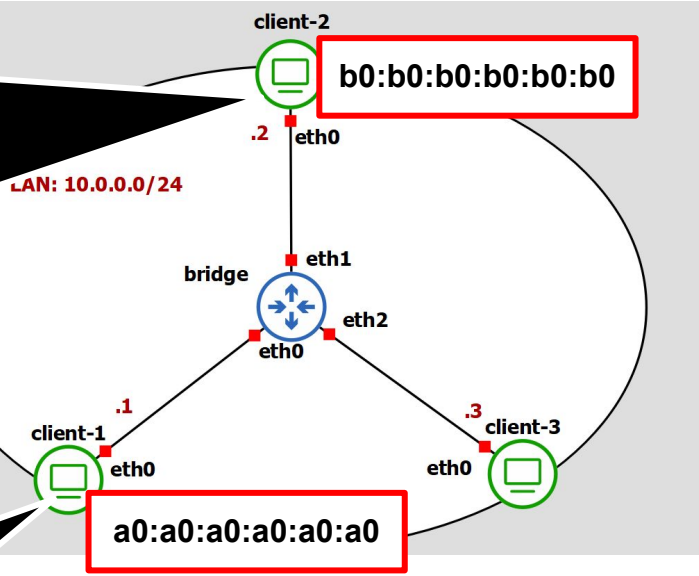
IEEE 802.1AE on Linux

- Starting with kernel 4.6, support for MACsec has been added in Linux
- There are two ways to implement MACsec:
  - manually configure secure channel(SC), security association(SA) and the keys(this is what we are going to see)
  - use dot1x with MACsec extensions that allows dynamic discovery of MACsec peers, SA and SC setup, key generation and distribution
- Let’s briefly see how to configure a static secure channel in Lab2
- Nice reading:
  - https://legacy.netdevconf.info/1.1/proceedings/slides/dubroca-macsec-encryption-wire-lan.pdf

Bidirectional Secure Channel between client1 and client2 in Lab2

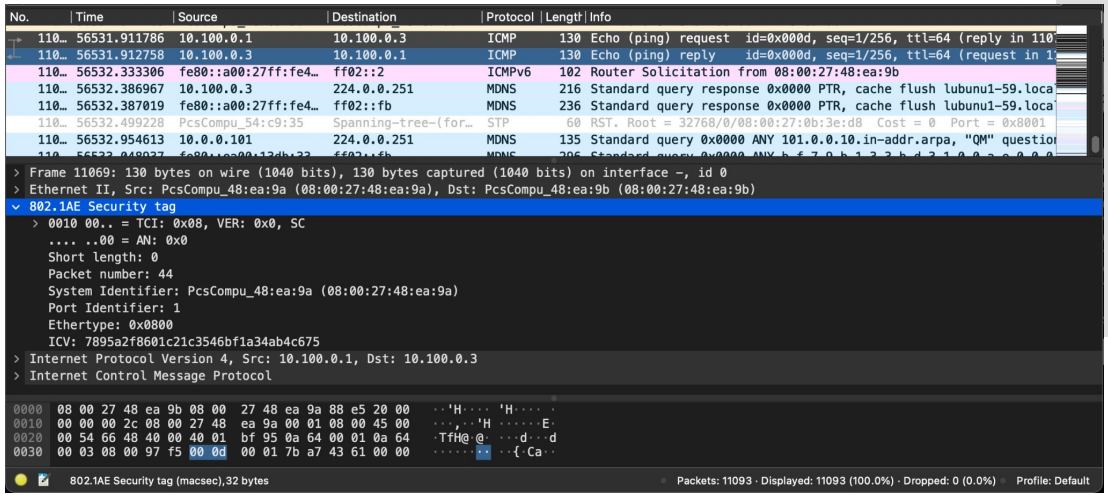
```
ip link add link eth0 macsec0 type macsec
ip macsec add macsec0 tx sa 0 pn 1 on key 02
12345678901234567890123456789012
ip macsec add macsec0 rx address a0:a0:a0:a0:a0:a0 port 1
ip macsec add macsec0 rx address a0:a0:a0:a0:a0:a0 port 1 sa 0 pn 1
on key 01 09876543210987654321098765432109
ip link set macsec0 up
ip addr add 10.100.0.2/24 dev macsec0

# with this conf only integrity is on
# to encrypt: ip link set macsec0 type macsec encrypt on
# for antireply: ip link set macsec0 type macsec replay on
# to test the configuration: ping 10.100.0.3 (check wireshark)
```

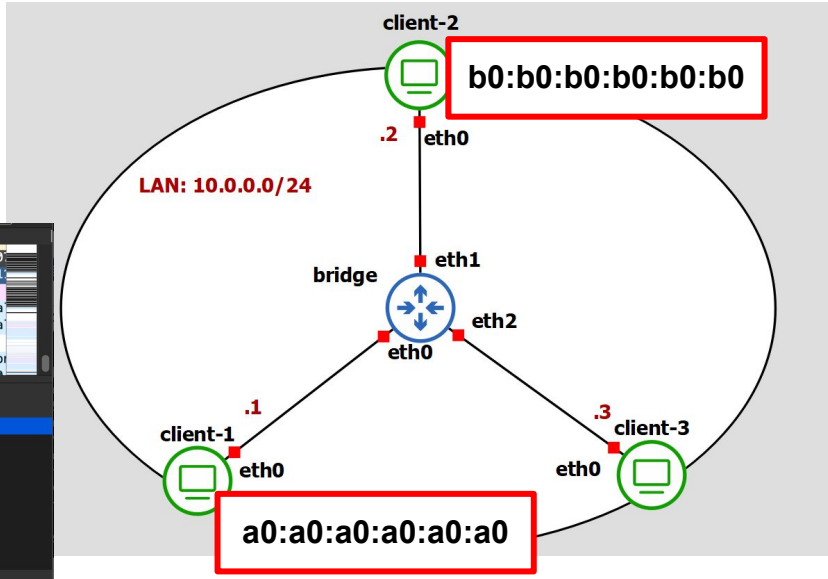
```
ip link add link eth0 macsec0 type macsec
ip macsec add macsec0 tx sa 0 pn 1 on key 01
09876543210987654321098765432109
ip macsec add macsec0 rx address b0:b0:b0:b0:b0:b0 port 1
ip macsec add macsec0 rx address b0:b0:b0:b0:b0:b0 port 1 sa 0
pn 1 on key 02 12345678901234567890123456789012
ip link set macsec0 up
ip addr add 10.100.0.1/24 dev macsec0
```



Bidirectional Secure Channel between client1 and client2 in Lab2

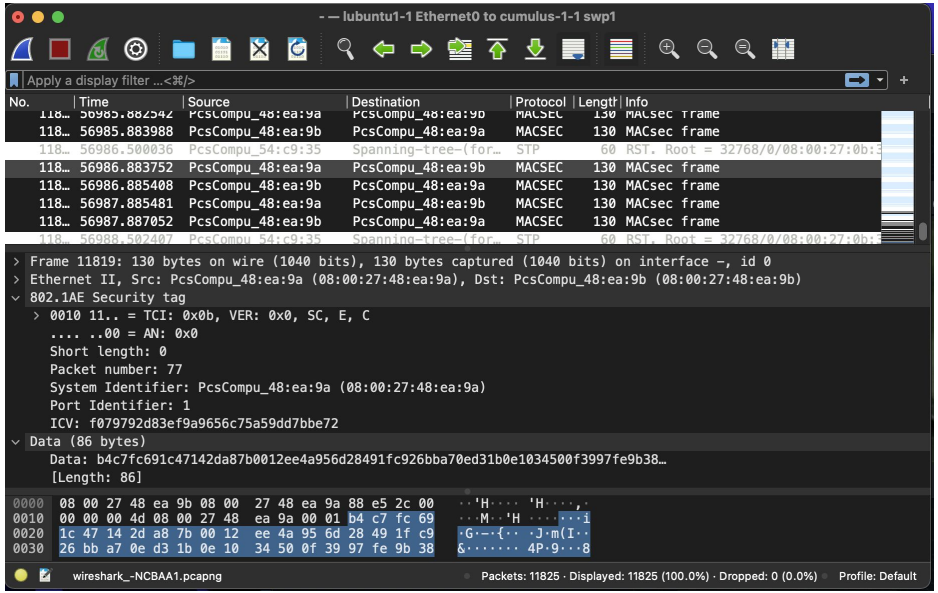
only integrity

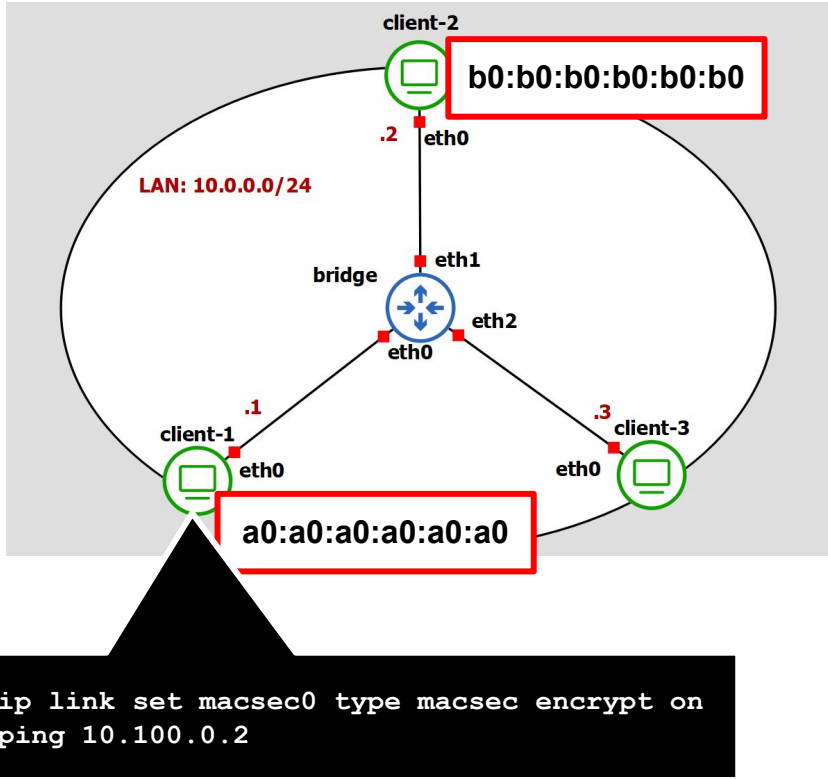




Bidirectional Secure Channel between client1 and client2 in Lab2

encryption on





Secure Protocols (2): Securing Address Resolution

- Address Resolution creates a major vulnerability in the Ethernet architecture
  - Information gained by DHCP snooping can be used to **prevent ARP spoofing attacks**, by tying MAC addresses to their corresponding IP addresses and ports, based on information gained from DHCP messages.
  - DHCP snooping suffers from a **potential lack of scope,** as a single switch can not see the allocations made to hosts whose path to the DHCP server does not pass through this switch
- The research community has mostly focused on cryptography based solutions
  - ***IPv4 secure address resolution: S-ARP***
    - adds an authentication field to ARP messages
    - provides a corresponding key management structure,
      - that uses cryptographic name space binding
      - or extends MACsec's reach from endpoint to endpoint and multicast protection
  - ***IPv6 secure address resolution: SEcure Neighbor Discovery (SEND)***
    - The Secure Neighbor Discovery (SEND) protocol is a security extension of the Neighbor Discovery Protocol (NDP) in IPv6 defined in RFC 3971 and updated by RFC 6494
    - It is the intent of SEND to provide an alternate mechanism for securing NDP with a cryptographic method that is independent of IPsec
    - SEND uses (i) Cryptographically Generated Addresses (CGA) and (ii) other new NDP options for the ICMPv6 packet types used in NDP

Security Monitoring

- ***Ethernet Firewall and Deep Packet Inspection (DPI)***
  - Firewalls are used to limit traffic between network segments (more complex cases of ACLs)
  - Firewalls can also employ DPI and application layer session recreation for inspection purposes
    - DPI means analyzing the contents of the packet at the application level, beyond the headers
  - Current firewall products can operate on all network layers and thus the concept of an “Ethernet firewall” lacks separate meaning (e.g. ebtables/iptables and NETFILTER).
  - The switches’ ACLs can be used to limit traffic on the Ethernet layer and standard firewall products can control the higher layers.
- ***Intrusion Detection and Prevention Systems***
  - IDS and IPS systems use DPI to identify network attacks, usually from a signature library of known attacks.
  - They require access to the network traffic that can be gained by placing an IDS/IPS device directly between two endpoints (typical when used as a firewall or to enhance a firewall) or they can monitor traffic from a switch via the port mirroring feature.
  - Port mirroring copies traffic to and from selected ports to a listening port, where the monitoring device is located

Discussion

<< It appears that the level of security increases with the efforts of administration and that there is no simple technological way to add self-configuring transparent security to the Ethernet layer. Plain, self-configured, out-of-the-box Ethernet is clearly not secure against any threats. MAC flooding, ARP spoofing, and STP attacks are easy to perform.

An Ethernet network configured according to vendor's instructions fares better. ARP spoofing is still possible (unless the switches feature DHCP snooping and ARP inspection). Switches still leak frames at MAC address table time outs. However, the key problem is that any mistake by network administrators will compromise the security.

ACLs, 802.1X or other authentication does not secure the network by itself. It just limits the potential attackers to known users, while adding to the workload of the administration.

Additional software may be needed at the hosts and the management of authentication information adds a considerable workload. MACsec or other frame encryption mechanisms solve eavesdropping issues, including frames broadcast at MAC table time out, and negates MITM attacks. DoS and traffic analysis are still possible, as are attacks through the network to higher layers. The workload is roughly the same as using authentication without MACsec.

Intrusion detection and prevention systems can detect several attacks. Some attacks, such as VLAN double tagging, are easily identifiable. DHCP snooping pairs MAC addresses to IP addresses and thwarts ARP spoofing >>