chapters\08-sniffing\sniffing-overview.md

Sniffing overview

- Capturing data packets on a network using a program or a device.

Networking concepts

Network adapter

- Can enable Wi-Fi (wireless, WLAN) and Ethernet (wired, LAN) connection.
- Can be a **NIC** (Network interface card)
  - Physical card that connects to an expansion slot in a computer
- Modern systems has usually an integrated network adapter (e.g. on motherboard).
- As default it discards messages that's not destined to it
  - See promiscuous mode for the opposite behavior.

Promiscuous mode

- Allows sniffing the packets after connecting to an access point
- 📝 Network interface controller pass all traffic it receives, rather than only destined ones.
- Works on both wired and wireless connections
- See also • libpcap | Sniffing tools • Turning on promiscuous mode | Wireshark

Monitor mode

- Allows sniffing the packets in the air without connecting (associating) with any access point.
- ❗ Wireless connection only

Sniffing types

Passive sniffing

- Does not require any packets to be sent
- Monitors and captures incoming packets
- Used in networks which use hubs i.e. shared ethernets
  - A **hub** forwards every frame to all ports but the sources filters

Active sniffing

- Require a packet to have a source and destination addresses in order to be sent to its destination
- Used in networks which use switches i.e. switched ethernets
  - A **switch** maps MAC addresses into ports, based on source addresses
  - A switch operates at data link layer (2) to forward data to MAC addresses
    - Some switches can run on network layer (3) with additional routing functionality.
      - Also known as layer-3 switches, or multilayer switches.
- E.g.
  - Port mirroring where each packet is also sent to a port that attacker listens to
  - Lawful interception where electronic surveillance on a target is authorized by a judicial or administrative order.

Port mirroring

- Used on a network switch
- Sends copy of network packets seen on one switch port (or an entire VLAN) to another port
- Often used in Intrusion Detection Systems.
- Also known as **span port**
  - In Cisco system, it's commonly referred as Switched Port Analyzer (SPAN)
- See also STP attack for an exploitation

Sniffer

- Packet sniffing programs
- Designed to capture packets that contain information such as passwords, router configuration, traffic.
- 📝 Works at data link layer (2) of the OSI model where MAC addresses work
  - It may then translate frames to higher level packets.
- Allows attackers to access the network traffic from a single point.
- Turns the network adapter into promiscuous mode or monitor mode

Wiretapping

- Also known as **telephone tapping** or **wire tapping**
- Monitoring of telephone and Internet-based conversations by a third party.
- Legal wiretapping by a government agency is also called **lawful interception (LI)**
- **Active wiretapping**: Alters communication by e.g. interjecting something.
- **Passive wiretapping**: Only monitors or records the traffic.
- 🤗 NSA wiretaps Internet going through using out-of-band signaling with their tool called PRISM

- **Out-of-band vs In-band signaling**
  - **In-Band signaling**: Method where signalling is sent over the voice/data circuit.
  - **Out-of-band signaling**: Data transmission through different channels (or frequencies) than normal ones.

Sniffing countermeasures

- Restrict the physical access to the network media
- 📝 Encryption is, by far, the best option.
  - E.g. • SSH instead of Telnet • Secure Copy (SCP) instead of FTP • SSL for email connection • HTTPS instead of HTTP • SFTP instead of FTP • WPA2 or WPA3 for wireless traffic
  - See also encrypting communication
- 📝 Use Access Control Lists (ACLs) on router/firewall to only allow authorized devices/IP ranges.
- Permanently add the MAC address of the gateway to the ARP cache.
- Use static IP addresses and static ARP tables
- Use switch instead of hub as switch delivers data only to the intended recipient.
- Use • PGP and S/MIPE • VPN • IPSec • SSL/TLS • Secure Shell (SSH) • One-time passwords (OTP).
- Retrieve MAC directly from NIC instead of OS to prevent MAC address spoofing.
- Use tools to determine if any NICs are running in the promiscuous mode.

chapters\08-sniffing\sniffing-tools.md

Sniffing tools

- Also known as • **sniffer** • **packet analyzer** • **protocol analyzer** • **network analyzer**
- 💡 Not only used for hacking but also for troubleshooting by e.g. system administrators

Cain and Abel

- Also known as **Cain & Abel** or **Cain**
- 📝 Recovery of various kind of passwords by sniffing the network
- 📝 Can also do
  - ARP poisoning
  - sniffing
  - recording VoIP conversations
  - password cracking with e.g. dictionary attacks, brute-force etc.
- See also • Cain and Abel | Wireless threats and attacks • Cain and Abel | Web server threats and attacks • ARP poisoning attack steps | ARP poisoning

libpcap

- 📝 Layer 2 Packet capture library for Linux/macOS
  - See Turning on promiscuous mode for Windows alternatives
- 📝 Used by most sniffers including • Wireshark • Snort • tcpdump • TCPflow • Cain and Abel • Kismet • Nmap
- Maintained and developed by tcpdump

TCPflow

- Open-source TCP/IP packet demultiplexer.
- Stores data in a way that makes it convenient for debugging and analysis
- Like tcpdump however, separate files for each direction are created, making things easier to read.
- Uses `libpcap`

tcpdump

- 📝 Command-line tool to show all TCP traffic from all interfaces live.
- Built-in for all Unix systems, has a Windows clone called WinDump
- Developed and maintains `libpcap`
- See `man page | tcpdump.org`

Wireshark

- 📝 Also known as **Ethereal** (old name)
- 📝 Captures and visualize traffic.
- 📝 **tshark**: Terminal-based Wireshark like tcpdump
- Can be started from Window managers or command line

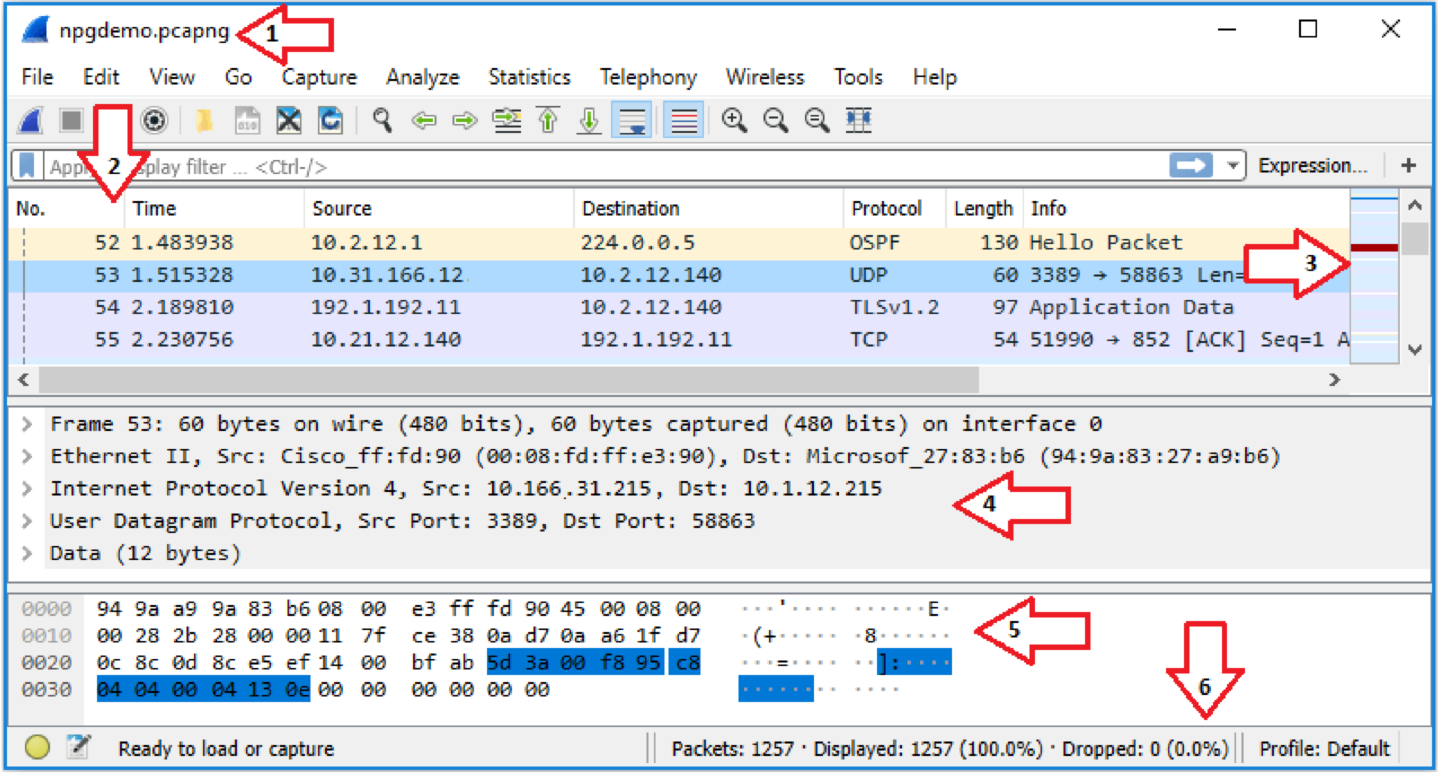
Turning on promiscuous mode

- Allows capturing all traffic, see Promiscuous mode | Sniffing
- On Linux/macOS it's done through `libpcap`
- 📝 On Windows a driver is required:
  - `npcap`: Driver from Nmap developers
  - `WinPcap`: Discontinued driver
  - AirPcap: Obsolete, propriety USB dongle used when there was no open-source Windows driver

Wireshark non-root installation

- Running wireshark as root is not safest
  - You're receiving traffic from an unknown location
  - If anything goes wrong, people can gain root access
- Install wireshark by e.g. `apt-get install wireshark-gtk` (varies by OS)
- `usermod -a -G wireshark <user-name>` to install it as non-root user
  - Adds wireshark to user account.

📝 Wireshark UI



No.	Time	Source	Destination	Protocol	Length	Info
52	1.483938	10.2.12.1	224.0.0.5	OSPF	130	Hello Packet
53	1.515328	10.31.166.12	10.2.12.140	UDP	60	3389 → 58863 Len=
54	2.189810	192.1.192.11	10.2.12.140	TLSv1.2	97	Application Data
55	2.230756	10.21.12.140	192.1.192.11	TCP	54	51990 → 852 [ACK] Seq=1 A

```
> Frame 53: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: Cisco_ff:fd:90 (00:08:fd:ff:e3:90), Dst: Microsof_27:83:b6 (94:9a:83:27:a9:b6)
> Internet Protocol Version 4, Src: 10.166.31.215, Dst: 10.1.12.215
> User Datagram Protocol, Src Port: 3389, Dst Port: 58863
> Data (12 bytes)
```

```
0000  94 9a a9 9a 83 b6 08 00  e3 ff fd 90 45 00 08 00   ···'····  ······E·
0010  00 28 2b 28 00 00 11 7f  ce 38 0a d7 0a a6 1f d7   ·(+·····  ·8······
0020  0c 8c 0d 8c e5 ef 14 00  bf ab 5d 3a 00 f8 95 c8   ···=····  ··]:····
0030  04 04 00 04 13 0e 00 00  00 00 00 00               ········  ····
```

Ready to load or capture | Packets: 1257 · Displayed: 1257 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

1. **Title Bar**
   - Shows the name of the interface you're capturing until you save your capture
   - Then it shows the name of the capture dump
2. **Packet List Pane**
   - You can add/remove/reorder columns
   - Selecting a packet will show more details in the Packet Details Pane and Packet Bytes Pane
3. **Intelligent Scrollbar**
   - Mini-map of packets
   - Depends on the height of the list and your physical display's specifications.
4. **Packet Details Pane**
   - Displays protocol fields
   - **Generated Fields**
     - Enclosed in brackets ([ ])
     - Contains info such as TCP analysis, response time, checksum validation, and IP geolocation.
   - **Links**
     - Wireshark will generate a link if it detects relationships between packets.
     - Formatted blue with an underline.
     - Double-clicking on the link will jump you to the related packet.
5. **Packet Bytes Pane**
   - Hexdump style with each line displaying the data offset
     - 16 hexadecimal bytes, and 16 ASCII bytes
6. **The Statusbar**
   - Informational messages

Wireshark filtering

- ❗ Capture filters (like `tcp port 80`) are not to be confused with display filters (like `tcp.port == 80`)

Display filters

- Control which packets are displayed
- Uses search and match operators such as `contains` and `matches`
  - E.g. `http contains hello`: TCP packets containing string "hello"
- Uses search comparisons
  - Such as
    - Equal: `eq` | `==`
    - Not equal: `ne` | `!=`
    - Greater than: `gt` | `>`
    - Less than: `lt` | `<`
    - Greater than or equal to: `ge` | `>=`
    - Less than or equal to: `le` | `<=`
  - 📝 E.g.
    - `tcp.port eq 21 or ssh`: show only FTP (port 21) or SSH traffic.
    - `ip.addr == 192.168.1.1`: examine all traffic from and to `192.168.1.1`s

Capture filters

- Also known as **PCAP filters**
- Same syntax as tcpdump or any other application using `libpcap`
- Much more limited than display filters
- Reduce the size of a raw packet capture, set before capturing
- E.g.
  - Only from traffic to / from specific host: `host 172.18.5.4`
  - Only from a range of IP addresses: `src net 192.168.0.0/24`

Kismet

- Kismet is an open-source wireless network and device detector, passive network sniffer, wardriving tool, and WIDS (Wireless Intrusion Detection system) framework.
- Can export in a compatible format for
  - cracking with aircrack-ng for deep packet analysis with a tool like Wireshark / tshark.
- Kismet can discover wireless networks that are not sending beacon frames.
  - Even if the security admin turns beaconing off (so no one can supposedly search for the SSIDs)

Kismet vs Wireshark

- Both looks at the contents of the packets and decodes them but presents them differently
  - Wireshark is packet oriented: digs into specifics of each packet
  - Kismet is device oriented: more device details, association with client.
- Both are passive-monitoring tools i.e. works without sending any loggable packets.
- Kismet is Wi-Fi only while Wireshark can also sniff on wired networks.

Mobile tools

- Wi.cap. Network Sniffer Pro for Android
- FaceNiff for Android (rooted only)
- PacketCapture for android

chapters\08-sniffing\sniffing-attacks-overview.md

Sniffing attacks overview

- Spoofing attacks
- ARP posioning

MAC flooding

MAC

- MAC address is a unique identifier of a network node.
- E.g. `52:54:00:e5:83:bb`
  - First three sets (`52:54:00`): Manufacturers signature
  - Last three sets is set in different ways depending on manufacturers
- Embedded in the device (firmware or some read-only part of the device)
- In a network, each device has its own MAC address
  - Associates the device with a physical port
- 🤗 If your MAC address is logged, police can use it to contact the manufacturer to ask who purchased the device.
  - Difficult to trace it if it was paid by cash.
- 💡🤗 You may have free WiFi forever if you can change your MAC address.
  - Usually checked in public places e.g. in an airport when they give you free WiFi.

Content Addressable Memory (CAM) table

- Used by switches
- Stores all available MAC addresses and their virtual LAN parameters for each port.
- Possible to sniff by flooding it.

MAC flooding attack

- Flooding the switch with thousands of MAC address mappings such that it cannot keep up.
  - When the table can't keep up it starts sending every message out to every port.
  - I.e. switch is forced to behave as a hub.
- Allowed by the fixed size of the CAM table.
- Steps:
  i. Send large number of fake MAC addresses to the switch until CAM table becomes full

        ii. Switch enters fail-open mode
            - where it broadcasts the incoming traffic to all ports on the network
        iii. Attacker (with promiscuous mode) starts sniffing the traffic passing through the network.
- Can be followed up using ARP spoofing to retain access to data after switches recover.
- See also MAC spoofing

DHCP attacks

DHCP introduction

- DHCP: Dynamic Host Configuration Protocol
- Client/server protocol
- Used by routers as they start a DHCP server
- Server provides following to DHCP-enabled clients:
    - IP addresses
    - Configuration information
    - Time period of the lease offer
- A possible way to drop connection of others in network is to brute-force DHCP server with "returning lease" messages.
    - It'll force everybody to lose connection and request IP addresses again

DHCP snooping

- Layer 2 security feature
- Built into operating system of a capable network switches
- Filters, rate-limits suspicious DHCP traffic
- Builds and maintains the **DHCP snooping binding database**
    - Also known as **DHCP snooping binding table**
    - Stores MAC + assigned IP + VLAN and switch ports
    - Uses to validate subsequent requests from untrusted hosts.
- 📝 **Dynamic ARP Inspection (DAI)**
    - Defense against too many incoming ARP broadcasts.
    - Each port on VLAN is untrusted by default
    - Each IP to MAC conversion is validated using DHCP snooping binding database.

DHCP starvation

- Exhaust all available addresses from the server
- Exploits that DHCP has a limited number of ip addresses to lease.
- A type of Denial of Service attack
- Flow

        i. Starve it, and no new clients will be able to connect
            a. Attacker broadcasts large number of DHCP REQUEST messages with spoofed source MAC addresses.
            b. Available IP addresses in the DHCP server scope becomes depleted.
            c. DHCP server becomes unable to allocate configurations to new clients and issue any IP addresses
        ii. Set-up rogue (fake server) to respond to the discovery requests
            a. Attacker sets up a rogue DHCP server to respond to DHCP discovery requests.
            b. If a client accepts the rogue server as its DHCP server, then the attacker can listen to all traffic going from or to the client.

- Tools
  - `yersinia`
    - Start UI using `yersinia -G` then click on "Start attack"
  - DHCPstarv

DHCP starvation countermeasures

- Authentication
- Configure DHCP snooping
- Trusted sources
  - ❗Vulnerable to mimicing them

Port security

- Allows traffic from a specific MAC address to enter to a port
- Only allowing one MAC through a port
- Only one IP at a time can be requested
- ❗ Vulnerable to spoofing MAC addresses

DNS poisoning

DNS introduction

- Domain Name Server
- 📝 Protocol that resolves domain names into IP addresses using default port 53.
- Stores domain name and IP address pairs in a **DNS table**.

DNS poisoning attack

- 📝 Also known as **DNS cache poisoning** and **DNS spoofing**
- 📝 Manipulating the DNS table by replacing a legitimate IP address with a malicious one
  - E.g. redirecting `cloudarchitecture.io` to attackers IP address.

- 🤗 Used for internet censorship in many countries.
- Flow
    1. Attacker makes DNS request to target
    2. DNS server asks the root name server for the entry
    3. Attacker floods the DNS server with a fake response for the targeted domain until legitimate response from root server is ignored
    4. The poisoned entry remains in cache for hours and even days
- Can be used after ARP poisoning through **DNS spoof** plugin of Ettercap.
- Can be followed up with e.g. • man-in-the-middle attacks • website defacement attacks

DNS poisoning countermeasures

- **Active monitoring**
    - Monitor DNS data for new patterns such as new host
    - E.g. by using intrusion detection system (IDS)
- **Keep DNS servers up-to-date**
    - Updated versions have port randomization and cryptographically secure transaction IDs against attackers.
- **Randomize source and destination IP, query IDs, during name requests**
    - Makes harder for attackers to send spoofed responses as it'd be harder to guess the address and query ID.
- **Use HTTPS and/or TLS for securing the traffic**
    - Also known as **DNS over HTTPS (DoH)** and **DNS over TLS (DoT)**
    - SSL and TLS use certificates to verify the identity of the other party.
    - So although they do not protect against cache poisoning itself, the certificates help to protect against the results

DNSSEC (Domain Name System Security Extension)

- Developed by The Internet Engineering Task Force (IETF)
    - Open standards organization, which develops and promotes voluntary Internet standards
- Help verifying the true originator of DNS messaging
- 📝 Provides secure DNS data authentication by using digital signatures and encryption.
    - Adds cryptographic signatures to existing DNS records, stored in DNS name servers.
- Widely considered one of the greatest cache poisoning prevention tool as a defense
- Allows verifying that a requested DNS record comes from its authoritative name server and wasn't altered, opposed to a fake record injected in a man-in-the-middle attack.
- **Chain of trust**: E.g. `cloudarchitecture.io`'s signature is verified by `.io` signature that is verified by root certificate (signed by IANA)
    - **IANA**: Centrally coordinates Internet for DNS Root, IP addressing, and other Internet protocol resources.

VLAN hopping

VLAN

- 📝 Allows multiple separate LANs/networks on same switch through logical grouping
- Provides network separation
  - Hosts one one VLAN does not see hosts on other one
- **Port-based VLAN**
  i. Designate set of ports on the switch
    - account department VLAN, shipping department VLAN..
  ii. Connect devices to right ports each group is a VLAN
- **Tag-based VLAN** aka IEEE 802.1q VLANs
  - Basically a tags frames with which VLAN it belongs to
    - Frame = Primitive packet on layer 2
  - Tagged frame = IEEE 802.1q frame
  - Can tag/assign based on e.g. 802.1x
- **Trunk** (=802.1q link)
  - Allows sharing VLANs (VLAN IDs) between switches

VLAN hopping attack

- Attacking host on a VLAN to gain access to traffic on other VLANs
- E.g. using Frogger
- **Switch spoofing**
  - Attacking host imitates a trunking switch
- **Double tagging**
  - Attacker prepends two VLAN tags to frames
  - Second tag is the target host
  - First switch removes first innocent VLAN tag and sends packet to second switch.
  - Allows bypassing security mechanisms and reaching the target hosts.
  - Replies are not forwarded to the attacker host

OSPF attacks

- Forms a trusted relationship with the adjacent router
- Usually these attacks go undetected
- **Remote attacks**: caused by misconfigurations

OSPF: Open Shortest Path First

- Most popular routing protocol for IP networks

- Dynamically discovers neighbors like RIPv2 and BPG (Border Gateway Protocol)
- Used by e.g. internet service providers (ISP) and cloud providers for hybrid communication

Compromised router attacks

- Placing a rogue router in target network e.g. remote branch/headquarters
- Allows attacker to inject routes to redirect traffic for MITM attacks or DoS attacks.
- Attacker learns about that entire routing domain such network types, links etc

OSPF attacks countermeasures

- 📝 Configure OSPF to authenticate every OSPF message
  - Routers must pass the authentication process before becoming OSPF neighbors.
- Monitor OSPF neighbors for eavesdropping through e.g. a SIEM

chapters\08-sniffing\spoofing-attacks.md

Spoofing attacks

- Entails changing a computer's identity
- Allow the bypassing of access control lists on servers or routers
- Allows hiding a device on network by impersonating another network device / system.

IP address spoofing

- Used most commonly in DDoS attacks.
- Helps with overcoming authentication based on IP addresses
  - Usually in corporate networks where trust between devices exists
  - E.g. accessing intranet without any password.
- ❗ The response is sent to the spoofed IP address instead of the spoofer.

IP address spoofing countermeasures

- Packet filtering by a gateway
  - Ingress: block packets from outside of the network having an IP address within the network.
  - Egress: block outgoing packets from inside with a source address that is not inside
- 💡 Design network protocols and services so that they do not rely on the source IP address for authentication.
- Sequence number
  - Used by upper layer TCP
  - Negotiated to ensure that arriving packets are part of an established connection.
  - 📝 Must be guessed in order to hijack the connection

MAC spoofing

- Response is received to spoofing party as opposed to IP address spoofing
- See also MAC | MAC flooding | Sniffing attacks

MAC spoofing use-cases

- New hardware for existing Internet Service Providers (ISP) where ISP charges per device.
- Fulfilling software requirements where one software can only be installed on a single device.

- Identity masking for pushing responsibility for other users.
- **MAC address randomization**: Implemented in Android, Linux, iOS, and Windows to prevent third parties from using the MAC address to track devices

MAC spoofing attack

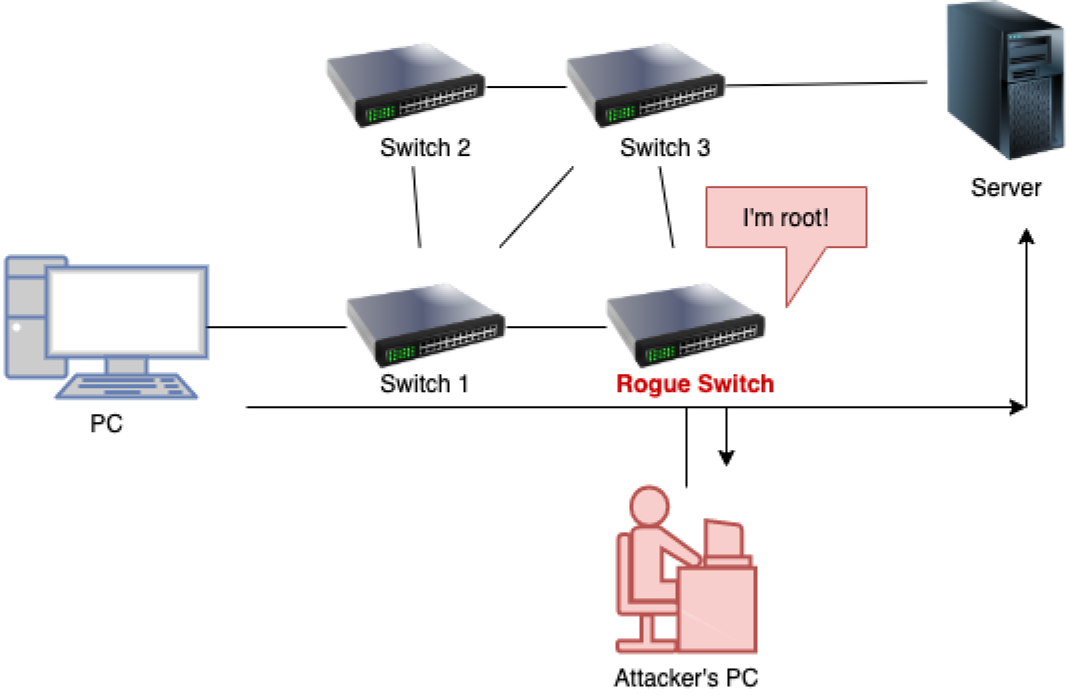
- Flow
  1. Attacker sniffs the network for MAC addresses of legitimate users
  2. Spoofs one of those addresses
  3. The attacker receives the traffic intended for that user
- Effective against MAC filtering
- E.g. using `ifconfig`
  1. `ifconfig` to get name of network interface e.g. `eth0`
  2. `ifconfig eth0 down` to deactivate it to be able to change it (will lose connection)
  3. `ifconfig eth0 hw ether 88:88:88:88:88:88` to change the MAC address
  4. `ifconfig eth0 up` to change the MAC address
- E.g. using macchanger
  - `-r` to get a random MAC address e.g. `macchanger -r eth0`
  - `-m` set specify MAC address manually to pretend to be someone else

STP spoofing

- STP: Spanning tree protocol
  - Layer 2 link management protocol
  - Provides path redundancy while preventing loops in the network
- Allows intercepting traffic when attacker emulates a device with a (lower) root switch identifier

STP spoofing attack

- Also known as ***STP manipulation attack***, ***STP attack*** or ***STP root role attack***.
- Flow
  1. Attacker introduces a rogue switch
  2. Switch advertises superior BPDUs to force a STP recalculation
     - BPDU = Bridge Protocol Data Units (BPDUs)
       - Frames that contain information about STP that's between exchanged switches
  3. Rouge router becomes elected as root switch
     - All the traffic will cross this switch giving the attacker possibility to sniff all traffic in the company

  - 

- Allows for
  - **DoS attacks**
    - Recalculation of STP have interruption on the system as the root bridge changes
    - Just sending BPDU messages would be enough as becoming root is not needed.
  - **MITM attacks**
    - Also known as dual-homing (dual-homed)
    - Attacker uses two interfaces, one to win the root other to send data to the attacker.
    - 📝 Attacker can configure one of the switch ports as a SPAN port to receive copy of the traffic.
- Mitigations
  - Enable **Root Guard** to not forward traffic to port with superior BPDUs
  - Enable **BPDU Guard** to enforce the STP domain borders

IRDP spoofing

- **IRDP**: ICMP Router Discovery Protocol
  - Protocol for computer hosts to discover routers on their IPv4 local area network.
  - ICMP router discovery messages are called "Router Advertisements" or "Router Solicitations"
- Vulnerable as it does not have any validation
- Attacker needs to be in the same network as the victim.
- Attacker adds bad route entries into a victim's routing table redirecting victim traffic to malicious address.
- Allows
  - Passive sniffing through rerouting victim machine to attacker machine
  - Man-in-the-middle where attacker acts as proxy
  - DoS by flooding wrong entries

- **Countermeasures**
  - Disable IRDP
  - Use digital signatures
  - Block all type 9 and type 10 ICMP packets.

chapters\08-sniffing\arp-poisoning.md

ARP poisoning

ARP

- ARP stands for "Address Resolution Protocol"
- 📝 In charge of resolving IP addresses to MAC addresses
- Can be used for obtaining MAC addresses of devices on the network
- Packets are `ARP_REQUEST` and `ARP_REPLY`
- Commands
  - `arp -a`: displays current ARP cache
  - `arp -d *`: clears ARP cache

ARP table

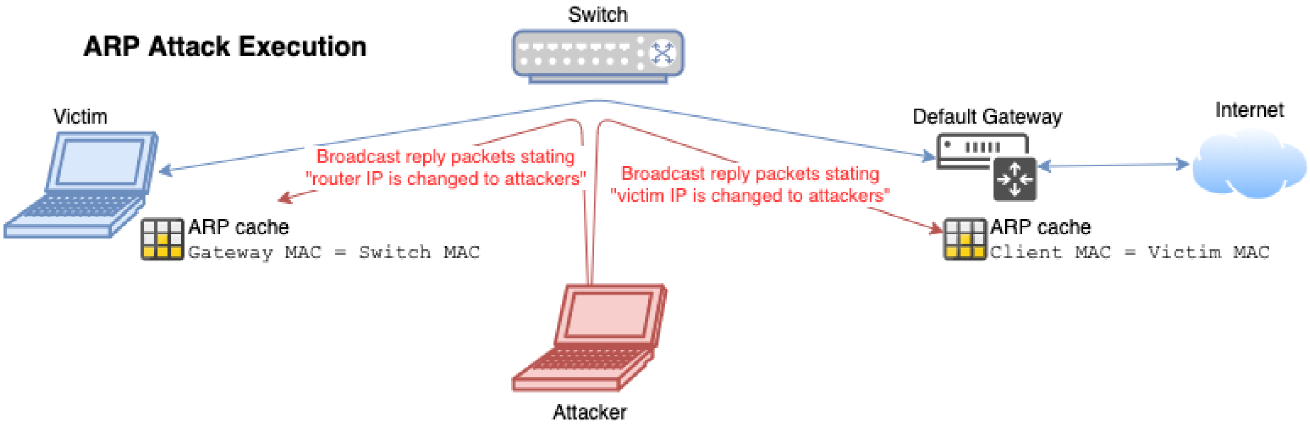
- Used to map MAC addresses to ip addresses
- Every network interface has its own ARP table
- 📝 If no ARP entry exist:
    i. Computer A broadcasts an APR request in network asking for the MAC address from a specific IP.
    ii. Computer B replies its MAC and IP address
    iii. Computer A inserts it to its ARP table for future use

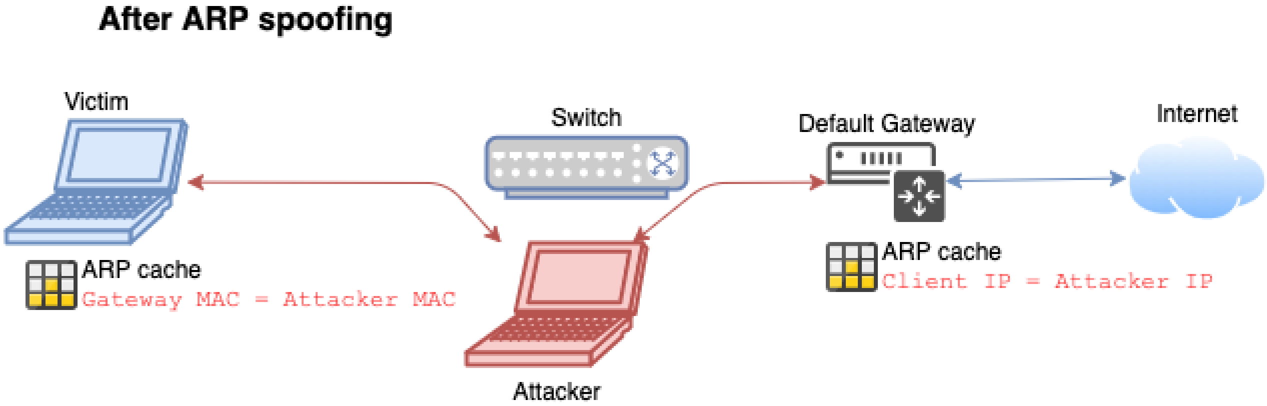
ARP poisoning attack

- Also known as • ***ARP spoofing*** • ***ARP spoofing*** • ***ARP cache poisoning*** • ***ARP poison routing*** • ***ARP cache flooding*** • ***ARP flooding***.
- Man in the middle attack between the victim and switch.
- Floods the target machines ARP cache with forged requests and responses.
- Exploits ARP not verifying the device authenticity
- If ARP cache becomes full, different behaviors can be observed depending on the manufacturer/implementation:
  - May force switch to operates in fail-safe mode
    - Behaves as a hub i.e. sends packets to every to all hosts
    - Same behavior is also seen in MAC flooding

- In Linux it may:
  - Drop the oldest / most stale entry from the table (by garbage collector)
  - Reject new entries

ARP poisoning attack steps

1. **Gather information**
   1. Get victim IP address, e.g. `192.168.122.183`
      - E.g. through host discovery using `nmap` e.g. `nmap -sn 192.168.0.0/24`
   2. Get default gateway IP, e.g. `192.168.122.1`
      - Usually IP of the machine ending with `.1`
      - Usually same for everyone on same network
      - Default gateway is the forwarding host (router) to internet when no other specification matches the destination IP address of a packet.
2. **Enable forwarding mode to sniff the traffic**
   - `echo 1 > /proc/sys/net/ipv4/ip_forward` in Linux.
   - ❗Otherwise no traffic is going through and you're just DOSing.
3. **Attack**
   - 
     ARP Attack Execution
Switch
Victim
Default Gateway
Internet
Broadcast reply packets stating "router IP is changed to attackers"
Broadcast reply packets stating "victim IP is changed to attackers"
ARP cache
Gateway MAC = Switch MAC
ARP cache
Client MAC = Victim MAC
Attacker
   - Deceive the victim device through flooding ARP reply packets to it.
     - Change gateways MAC address is to the attackers
   - 📝 Use an ARP spoofing tool e.g.
     - `arpspoof`
       - `arpspoof -t <victim-machine-ip> <default-gateway-ip>`
       - `arpspoof -t <default-gateway-ip> <victim-machine-ip>`
     - `ettercap`
       - Also sniffs passwords automatically
       - `ettercap -NaC <default-gateway-ip> <victim-machine-ip>`
         - `N` : make it non-interactive
         - `a` : arp posion
         - `c` : parse out passwords and usernames.
     - Cain and Abel (Cain & Abel) on Windows



4. **Sniff**
   - Now you sniff the traffic between two devices.
     - If through HTTPS & SSL you can only see basic data such as User Agent and domain names.
   - Can use e.g. `wireshark` or dsniff

ARP poisoning attack countermeasures

- Configure DHCP snooping
- Add **static** IP-MAC entries to the cache.
  - Then it will not process any ARP Replies received unlike a dynamic ARP cache.
- Use Intrusion Detection Systems (IDS)

ARP poisoning countermeasures

- **ARP spoofing detection and prevention**
  - Relies on some form of certification or cross-checking of ARP responses
  - Can be implemented on individual hosts, hypervisors or switches
  - 📝 E.g. DHCP snooping feature on switch OS can activate **Dynamic ARP Inspection** with an internal database.
  - ❗ Not possible if any host holds a static IP, and static ARP entries must be used.
- **Static ARP entries**
  - Manually mapping IP addresses to MAC addresses (maintaining ARP entries)
  - A lot of administrative overhead
  - Provides only basic security
- **OS security**
  - Linux ignores unsolicited replies, behavior can often be configured in other OSes