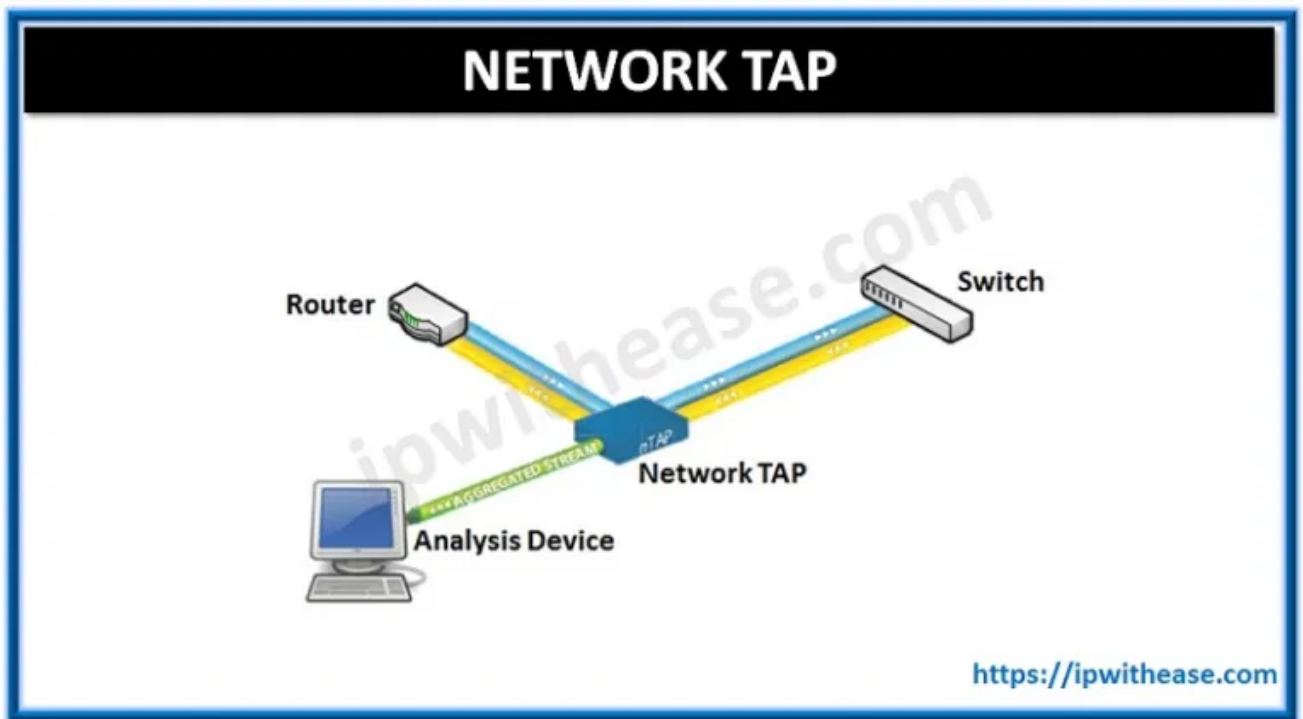


Sniffing

Sniffing is the process of monitoring and capturing all the packets passing through a given network using sniffing tools. It is a form of "tapping phone wires" and get to know about the conversation. It is also called wiretapping applied to the computer networks.

Active and Passive Sniffing

- **Passive sniffing** - watching network traffic without interaction; only works for same collision domain
- **Active sniffing** - uses methods to make a switch send traffic to you even though it isn't destined for your machine
- **Span port** - switch configuration that makes the switch send a copy of all frames from other ports to a specific port
 - Not all switches have the ability to do this
 - Modern switches sometimes don't allow span ports to send data - you can only listen
- **Network tap** - special port on a switch that allows the connected device to see all traffic



-
- **Port mirroring** - another word for span port

Basics

- Sniffing is capturing packets as they pass on the wire to review for interesting information
- **MAC** (Media Access Control) - physical or burned-in address - assigned to NIC for communications at the Data Link layer
 - 48 bits long
 - Displayed as 12 hex characters separated by colons
 - First half of address is the **organizationally unique identifier** - identifies manufacturer
 - Second half ensures no two cards on a subnet will have the same address
- NICs normally only process signals meant for it
- **Promiscuous mode** - NIC must be in this setting to look at all frames passing on the wire
- **CSMA/CD** - Carrier Sense Multiple Access/Collision Detection - used over Ethernet to decide who can talk
- **Collision Domains**
 - Traffic from your NIC (regardless of mode) can only be seen within the same collision domain
 - Hubs by default have one collision domain
 - Switches have a collision domain for each port

Protocols Susceptible

Some of the protocols that are vulnerable to sniffing attacks.

- **IMAP, POP3, NNTP and HTTP** all send over clear text data
- **SMTP** is sent in plain text and is viewable over the wire. SMTP v3 limits the information you can get, but you can still see it.
- **FTP** sends user ID and password in clear text
- **TFTP** passes everything in clear text
- **TCP** shows sequence numbers (usable in session hijacking)
- **TCP** and **UDP** show open ports
- **IP** shows source and destination addresses

ARP

- Stands for Address Resolution Protocol
- Resolves IP address to a MAC address

- Packets are ARP_REQUEST and ARP_REPLY
- Each computer maintains its own ARP cache, which can be poisoned
- **Commands**
 - `arp -a` displays current ARP cache
 - `arp -d *` clears ARP cache
- Works on a broadcast basis - both requests and replies are broadcast to everyone
- **Gratuitous ARP** - special packet to update ARP cache even without a request
 - This is used to poison cache on other machines

IPv6

- Uses 128-bit address
- Has eight groups of four hexadecimal digits
- Sections with all 0s can be shorted to nothing (just has start and end colons)
- Double colon can only be used once
- Loopback address is ::1

IPv6 Address Type	Description
Unicast	Addressed and intended for one host interface
Multicast	Addressed for multiple host interfaces
Anycast	Large number of hosts can receive; nearest host opens

IPv6 Scopes	Description
Link local	Applies only to hosts on the same subnet (Address block fe80::/10)
Site local	Applies to hosts within the same organization (Address block FEC0::/10)
Global	Includes everything

- Scope applies for multicast and anycast
- Traditional network scanning is **computationally less feasible**

Wiretapping

Wiretapping, also known as telephone tapping, is the process of monitoring telephone and Internet conversations by a third party, often by covert means.

- **Lawful interception** - Legally intercepting communications between two parties

- **Active** - Interjecting something into the communication
- **Passive** - Only monitors and records the data
- **PRISM** - System used by NSA to wiretap external data coming into US

MAC Flooding

- Switches either flood or forward data
 - If a switch doesn't know what MAC address is on a port, it will flood the data until it finds out
 - **CAM Table** - the table on a switch that stores which MAC address is on which port
 - If table is empty or full, everything is sent to all ports
 - MAC Flooding will often destroy the switch before you get anything useful, doesn't last long and it will get you noticed. Also, most modern switches protect against this.
 - **CAM Table Overflow Attack** - Occurs when an attacker connects to a single or multiple switch ports and then runs a tool that mimics the existence of thousands of random MAC addresses on those switch ports. The switch enters these into the CAM table, and eventually the CAM table fills to capacity. *(This works by sending so many MAC addresses to the CAM table that it can't keep up).* **This attack can be performed by using macof.**
- ![macof](<https://i0.wp.com/kalilinuxtutorials.com/wp-content/uploads/2015/09/macof2.png>)
- **Tools for MAC flooding**
 - Etherflood
 - Macof
 - Dsniff

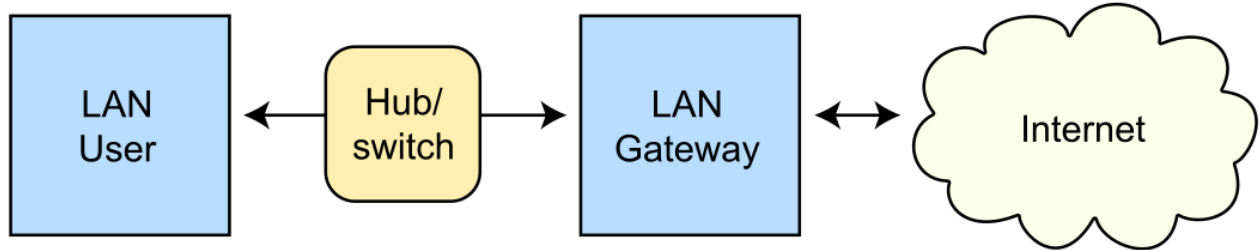
Switch port stealing

Tries to update information regarding a specific port in a race condition

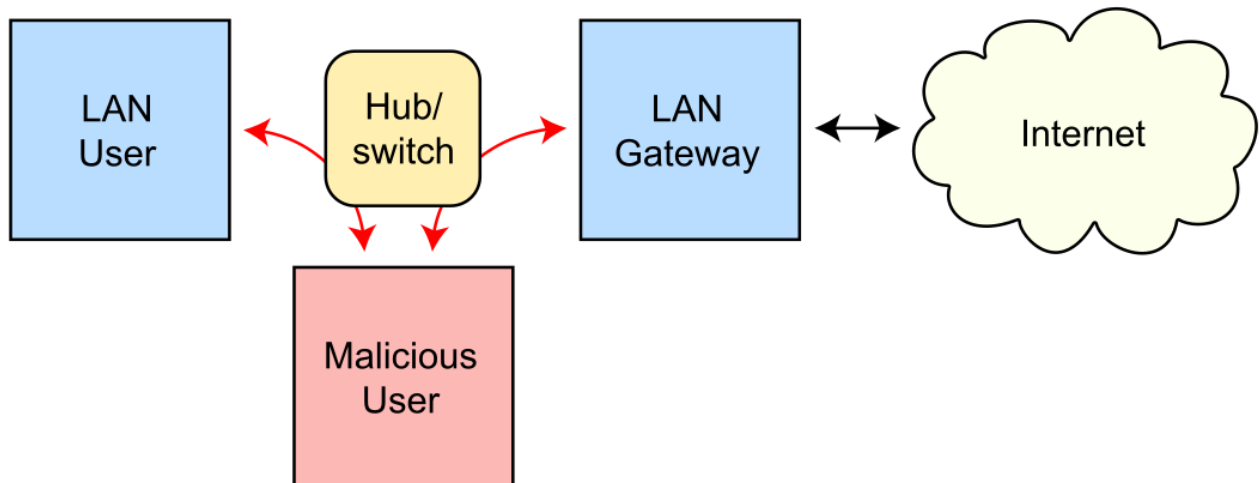
1. ARP Flood
 - Source MAC address same as victim
 - Destination MAC is attacker's
 - CAM updates port info (stolen)
2. Attacker now intercepts victim traffic
3. Attacker stops flooding
4. Victim reclaims port
5. Attacker retransmits captured data

ARP Poisoning

Routing under normal operation



Routing subject to ARP cache poisoning



ARP spoofing is a type of attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network. This results in the linking of an attacker's MAC address with the IP address of a legitimate computer or server on the network.

- Also called ARP spoofing or gratuitous ARP
- This can trigger alerts because of the constant need to keep updating the ARP cache of machines
- Changes the cache of machines so that packets are sent to you instead of the intended target
- Countermeasures
 - Dynamic ARP Inspection using DHCP snooping
 - Can use Static ARP ACL to map
 - Header to Payload validation
 - XArp software can also watch for this
 - Default gateway MAC can also be added permanently into each machine's cache
- Tools for ARP Poisoning

- Cain and Abel
- WinArpAttacker
- Ufasoft
- dsniff

DHCP Starvation

Is an attack that targets DHCP servers whereby forged DHCP requests are crafted by an attacker with the intent of exhausting all available IP addresses that can be allocated by the DHCP server.

- Attempt to exhaust all available addresses from the server
- Attacker sends so many requests that the address space allocated is exhausted
- DHCPv4 packets - DHCPDISCOVER , DHCPOFFER , DHCPREQUEST , DHCPACK
- DHCPv6 packets - Solicit, Advertise, Request (Confirm/Renew), Reply
- **DHCP Steps**
 - i. Client sends DHCPDISCOVER
 - ii. Server responds with DHCPOFFER
 - iii. Client sends request for IP with DHCPREQUEST
 - iv. Server sends address and config via DHCPACK
- **Tools**
 - Yersinia
 - DHCPstarv
- Mitigation is to configure DHCP snooping
- **Rogue DHCP Server** - setup to offer addresses instead of real server. Can be combined with starvation to real server.

Spoofing

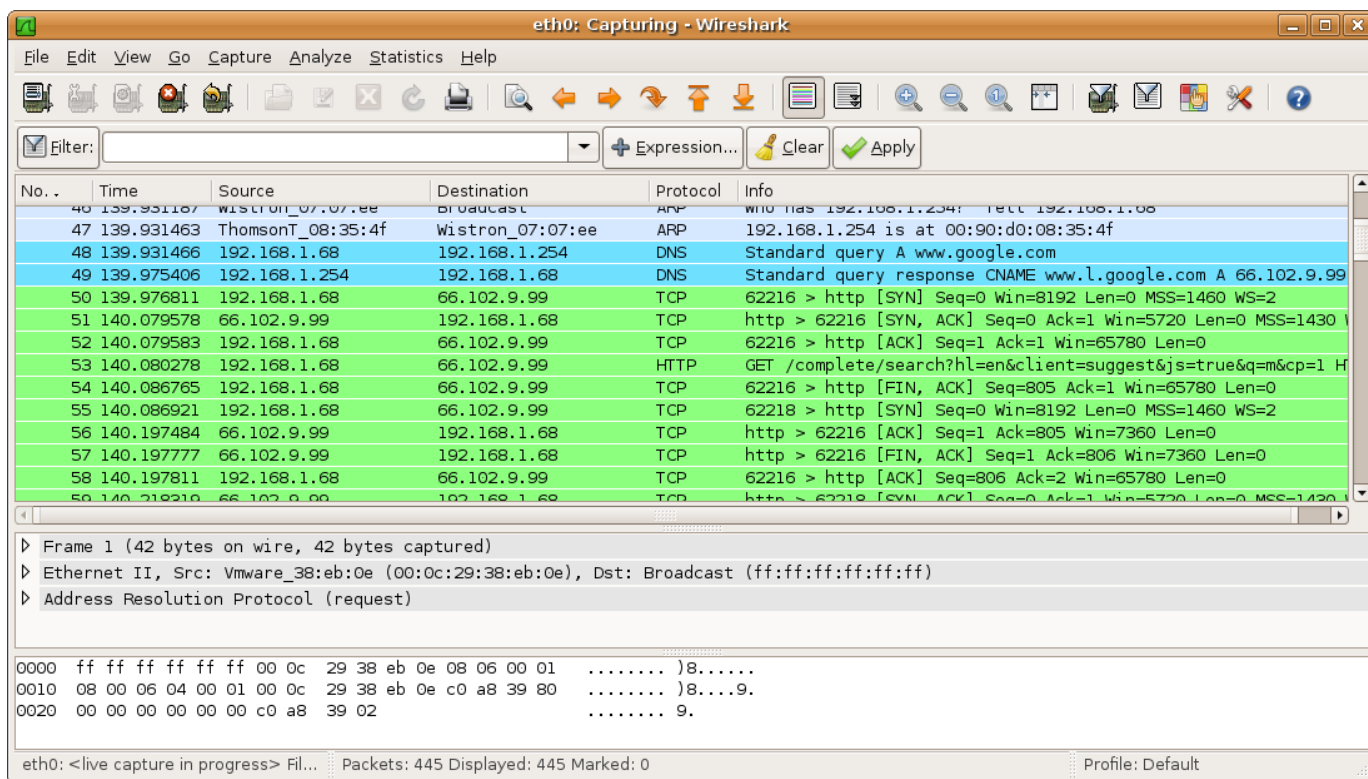
- **MAC Spoofing** - Changes your MAC address. Benefit is CAM table uses most recent address.
 - Port security can slow this down, but doesn't always stop it.
 - MAC Spoofing makes the switch send all packets to your address instead of the intended one until the CAM table is updated with the real address again.
- **IRDP Spoofing** - Attacker sends ICMP Router Discovery Protocol messages advertising a malicious gateway.
- **DNS Poisoning** - Changes where machines get their DNS info from, allowing attacker to redirect to malicious websites.

Sniffing Tools

Wireshark

Wireshark is the world's foremost and widely-used network protocol analyzer. It lets you see what's happening on your network at a microscopic level.

- With Wireshark you can inspect and detect ARP poisonings, Rogue DHCP servers, Broadcast Storm etc.




- - Previously known as Ethereal
 - Can be used to follow streams of data
 - Can also filter the packets so you can find a specific type or specific source address
- **Wireshark filters:**
 - **!(arp or icmp or dns)**
 - Filters out the "noise" from ARP, DNS and ICMP requests
 - ! - Clears out the protocols for better inspection
 - **tcp.port == 23**
 - Look for specific ports using tcp.port
 - **ip.addr == 10.0.0.165**
 - Look for specific IP address
 - **ip.addr == 172.17.15.12 && tcp.port == 23**
 - Displays telnet packets containing that IP

- `ip.src == 10.0.0.224 && ip.dst == 10.0.0.156`
 - See all packets exchanged from IP source to destination IP
- `http.request`
 - Displays HTTP GET requests
- `tcp contains string`
 - Displays TCP segments that contain the word "string"
- `tcp.flags==0x16`
 - Filters TCP requests with ACK flag set

tcpdump

Tcpdump is a data-network packet analyzer computer program that runs under a command line interface. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.

-  [tcpdump](#)
- Syntax
 - `<tcpdump flag(s) interface>`
 - `tcpdump -i eth1`
 - Puts the interface in listening mode
- WinDump is a Windows version similar to tcpdump.

tcptrace

- Analyzes files produced by packet capture programs such as Wireshark, tcpdump and Etherpeek

Other Tools

- Ettercap - also can be used for MITM attacks, ARP poisoning. Has active and passive sniffing.
- Capsa Network Analyzer
- Snort - usually discussed as an Intrusion Detection application
- Sniff-O-Matic
- EtherPeek
- WinDump
- WinSniffer

Defending and Countermeasures techniques against Sniffing:

- Disable ARP Dynamic
- ARP Spoofing detection tools
- Encrypt all the traffic that leaves your system

- Avoid public Wi-Fi spots
- Network scanning and monitoring
- Reverse DNS lookup's on logs == Sniffer
- **Ping** suspect clients with **wrong MAC address**
 - If suspect accepts the packet, is a good indication that he is sniffing the network / using NIC in promiscuous mode.
- Use **Nmap** with nse-script for **Sniffer Detect**:
 - `nmap --script=sniffer-detect <target>`