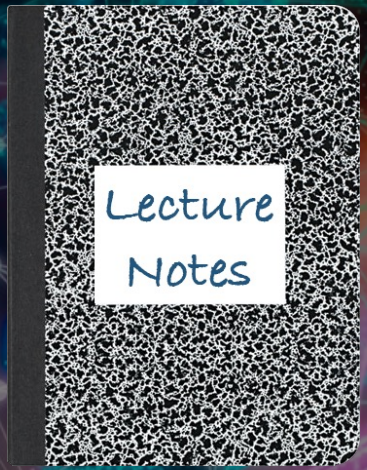
CS 419: Computer Security

Week 10: Network Security

Part 1: Intro & Link Layer



Paul Krzyzanowski

© 2026 Paul Krzyzanowski. No part of this content may be reproduced or reposted in whole or in part in any manner without the permission of the copyright owner.

Overview

How networks work

Addresses, switches, routers, layers, and the steps behind a web request

Why the protocols trust

No authentication was built in, so checks were added around them

Attacks on the local network

CAM tables, ARP, DHCP, VLANs, and rogue access points

Routing between networks

BGP hijacking and its slowly deployed defenses

The transport layer

Guessing, flooding, and resetting TCP connections

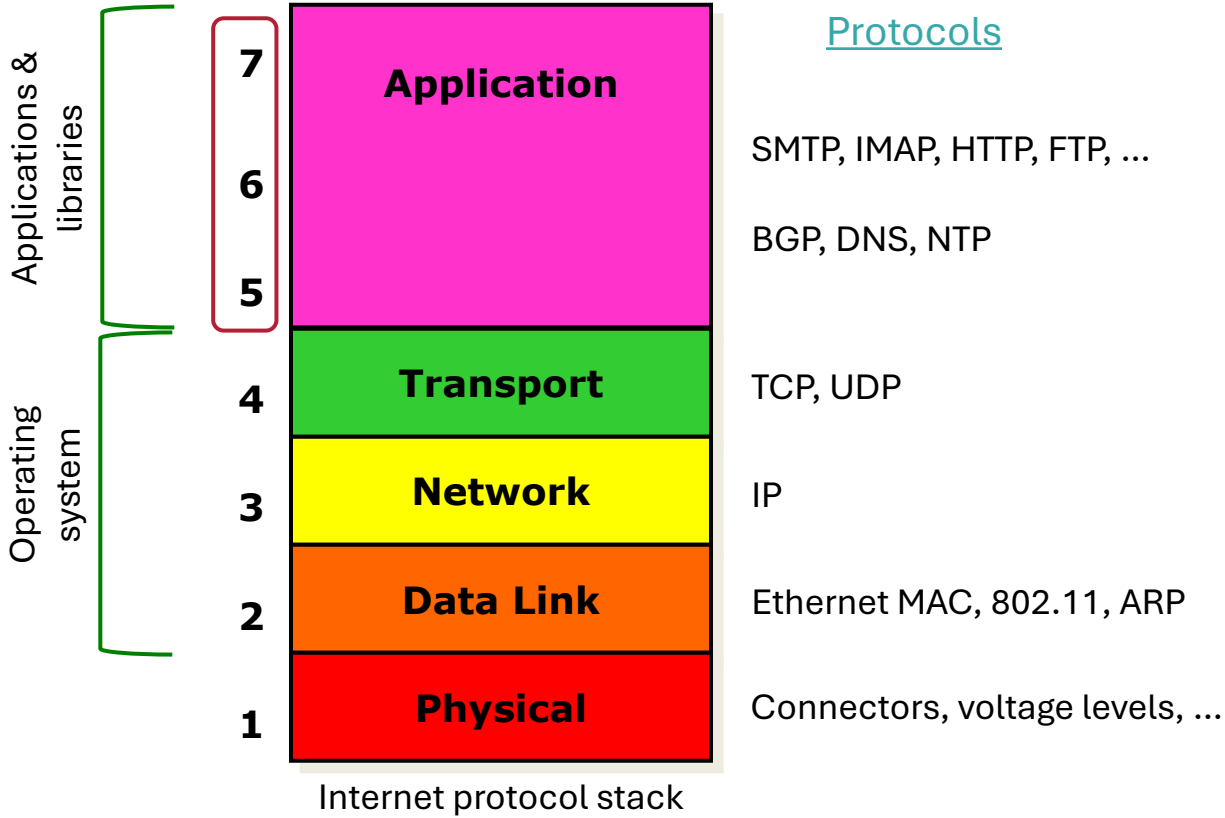
DNS

Pharming, cache poisoning, DNSSEC, rebinding, and hijacked domains

How a Packet Reaches Its Destination

The networking background the attacks depend on

The Internet Protocol Stack



Two Kinds Of Addresses

MAC address: identifies an Ethernet or Wi-Fi interface on its local network

- 48 bits, written as six pairs of hex digits: `b4:96:91:cf:f4:a4`
- Usually set by the manufacturer, but software can change it

IP address: identifies a host across networks

- IPv4: 32 bits, written as four decimal numbers: `128.6.4.2`
- IPv6: 128 bits
- Allocated in blocks called **prefixes**: `128.6.0.0/16` is every address starting with the same 16 bits

Some addresses are never routed on the Internet:

- **Private addresses** (`192.168.x.x`, `10.x.x.x`)
- **Loopback address** (`127.0.0.1`)

Local Networks, Switches, And Routers

- **Local network**
  - Set of devices that send **frames** directly to one another through a switch or Wi-Fi access point
- A **switch** has physical ports – each connected to a device or switch
  - It learns which MAC address is on each port by watching the source addresses of frames
  - A frame for a known destination goes only to that port
- A **broadcast** frame reaches every device on the local network
- A **router** connects the local network to other networks
  - Hosts send packets for anywhere else to the router: the **default gateway**

Joining A Network: DHCP And ARP

DHCP = Dynamic Host Configuration Protocol

A new laptop in a new network broadcasts a **DHCP** request

- Any DHCP server responds with: IP address, a default gateway, and a DNS server
- The laptop usually takes the first response
- **Nothing proves the server was authorized to make the offer**

ARP = Address Resolution Protocol

A machine needs the MAC address of a machine it wants to send a packet to.

It broadcasts an **ARP** request

- "Who has 128.6.13.4?" The device with that address replies with its MAC address
- The reply is cached. Reaching the router works the same way

TCP And UDP

TCP: Transmission Control Protocol

- A reliable, ordered stream of bytes between two programs
- A three-message handshake sets up a **connection**
- Every byte has a **sequence number**: the receiver orders data, detects gaps, and acknowledges
- Lost segments are retransmitted

UDP: User Datagram Protocol

- Individual messages, without a connection, sequence numbers, or retransmission
- Used by DNS, voice and video, games, and time synchronization

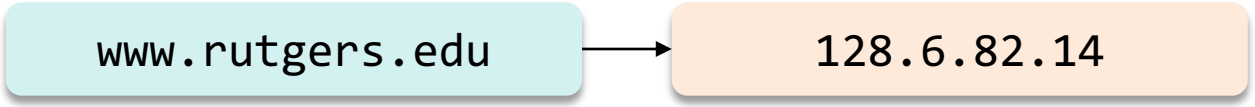
- A **port number** on each host identifies the program the data is for

Names And DNS

People use names: `www.rutgers.edu`

- IP Packets are addressed to IP addresses: `128.6.82.14`

The **Domain Name System** (DNS) translates names into addresses



A host sends its query to a **resolver**

- If the resolver has not cached the answer, it asks a chain of servers, starting at the root, until it reaches one that is authoritative for the domain

A Web Request, Step By Step

A laptop loads `https://www.rutgers.edu`:

1. The laptop received its IP address, gateway, and DNS server from **DHCP**
2. The DNS server is outside the local network
   - The laptop finds the router's MAC address with **ARP**
3. The browser asks the resolver for the address of `www.rutgers.edu`
   - **DNS** query(`www.rutgers.edu`) via UDP → 128.6.82.14
4. The laptop starts a TCP connection by sending a SYN
   - The Ethernet frame is addressed to the router's MAC address; the IP packet inside is addressed to the web server

A Web Request, Step By Step (cont'd)

5. Each router forwards the packet toward the destination.
   - Between organizations, routes come from **BGP** announcements

home firewall → router | Comcast | Twelve99 | Rutgers

6. The SYN reaches the server, which replies
   - Laptop and server complete the TCP handshake
7. A **TLS** handshake proves the server's identity with a certificate
   - Encrypts the connection
8. The browser sends an **HTTP** request
   - The server sends the page

Where Each Step Can Be Attacked

Step	Protocol	Attack
1. Receive network configuration	DHCP	A rogue DHCP server supplies the attacker's gateway or DNS server
2. Find the router's MAC address	ARP	A forged ARP reply redirects traffic to the attacker
3. Translate the name to an address	DNS	A forged reply sends the browser to the attacker's server
4. Send the frame to the router	Ethernet	An overflowed CAM table copies the frame to other ports
5. Route packets across networks	BGP	A false route announcement pulls traffic into the attacker's network
6. Set up a connection	TCP	A flood of connection requests exhausts the server
7. Authenticate the server and encrypt	TLS	The certificate is verified, so this step resists forgery
8. Send a request	HTTP	A flood of requests exhausts the application

Why The Protocols Lack Authentication

End-to-end principle (1984): application-specific functions such as reliability, security, and error recovery belong in the hosts, not in the core of the network

- IP does not guarantee delivery or ordering
- TCP at the endpoints provides reliability and ordering
- TLS provides encryption and authentication

- ARP, DNS, and BGP were designed for a trustworthy environment
  - No cryptographic authentication.
  - They cannot be replaced: every OS, router, and name server implements them

- Security was added around them: TLS, DNSSEC, RPKI, and checks in switches, routers, and resolvers

The Internet Introduces Risks

"The internet was designed to be open, transparent, and interoperable. Security and identity management were secondary objectives in system design. This lower emphasis on security in the internet's initial design not only gives attackers a built-in advantage. It can also make intrusions difficult to attribute, especially in real time. This structural property of the current architecture of cyberspace means that we cannot rely on the threat of retaliation alone to deter potential attackers. Some adversaries might gamble that they could attack us and escape detection."

– William J. Lynn III, Deputy Defense Secretary, 2010

http://archive.defense.gov/speeches/speech.aspx?speechid=1593

Attacks on the Local Network

The attacker is on the same network as the victim

Local Access And What It Buys

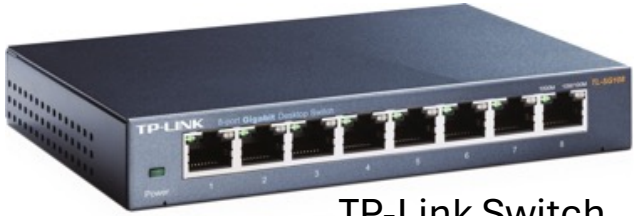
- Most attacks in this section need the attacker on the victim's local network
  - An open Wi-Fi network in a coffee shop or airport
  - A compromised machine inside the organization
  - A device plugged into an unused Ethernet jack
  - The **evil twin** is the exception: it creates a network the victim joins
- The protocols on a local network assume every device is telling the truth
- Most of these attacks aim at one position: **between the victim and the router**

Overflowing a Switch's CAM Table

Make the switch copy other devices' frames to the attacker's port

Ethernet Switches





TP-Link Switch

- 8 1-GbE ports

Cisco Nexus 9516 Switch

- 1/10/40 GbE
- 21-rack-unit chassis
- Up to 576 1/10 Gb ports

The CAM Table

- A switch is **self-learning**
  - Its table of MAC address → port starts empty
  - Whenever a frame arrives, the switch records the frame's source MAC address against the port it came in on
  - Entries that have not been used for a while are deleted
- A frame for a known destination goes to one port. A frame for an unknown destination is copied to every port
- Switches must be fast, so the table is held in content addressable memory (CAM): memory searched by content, which returns the port for a MAC address in one step
  - CAM is expensive, so the table is small: a few thousand entries. It is called the CAM table

CAM Table Overflow Attack

Goal: eavesdrop on traffic between other devices on the switch

- Requires: a port on the switch
- Flood the switch with frames carrying invented source MAC addresses
  - Each new address takes a table entry. The table fills with junk
- Frames for destinations no longer in the table are copied to every port
  - **The attacker's port now receives other devices' frames**
  - Destinations still in the table are switched normally
- Captured: passwords and session cookies on unencrypted protocols. Encrypted traffic is copied too, but cannot be read

dsniff: collection of tools for network auditing and penetration testing
`https://monkey.org/~dugsong/dsniff/`

Defending Against CAM Table Overflow

- All defenses require a **managed switch**
- **Port security**
  - Limit how many MAC addresses may appear on a port, or list which ones are allowed
  - Disable the port or raise an alert when the limit is exceeded
- **Disable unused ports** so a rogue device gains nothing from a spare jack
- **802.1X**: traffic from a port is unauthorized until the device authenticates
- Unmanaged switches, the home kind, have none of these controls

ARP Cache Poisoning

Put the attacker between a victim and its router

Finding A MAC Address For An IP Address

- We need to send a datagram to an IP address
- It is encapsulated in an Ethernet frame that contains a MAC address



How do we know what MAC address to use for a given IP address?

`ilab1.cs.rutgers.edu` :
IP address: `128.6.13.2`
MAC address: `b4:96:91:c4:36:76`

`ilab4.cs.rutgers.edu`:
IP address: `128.6.13.5`
MAC address: `b4:96:91:e3:24:cc`

The ARP Table

- The operating system keeps an **ARP table** (IP address → MAC address) and fills in each frame's destination from it
  - Entries come from ARP replies and expire after a while

```
$ arp -a
vlan13-lcsr-gw.rutgers.edu (128.6.13.1) at 50:9a:4c:e3:0d:91 [ether] on
enp194s0f0
ilab1.cs.rutgers.edu (128.6.13.2) at b4:96:91:c4:36:76 [ether] on enp194s0f0
ilab4.cs.rutgers.edu (128.6.13.5) at b4:96:91:e3:24:cc [ether] on enp194s0f0
ilabu.cs.rutgers.edu (128.6.13.6) at b4:96:91:78:c8:ac [ether] on enp194s0f0
ilab2.cs.rutgers.edu (128.6.13.3) at b4:96:91:c4:38:b0 [ether] on enp194s0f0
rlab1.cs.rutgers.edu (128.6.13.7) at b4:96:91:dc:f4:ad [ether] on enp194s0f0
```

What if the IP address we want is not in the cache?

ARP Cache Poisoning

Goal: redirect a victim's traffic through the attacker's machine

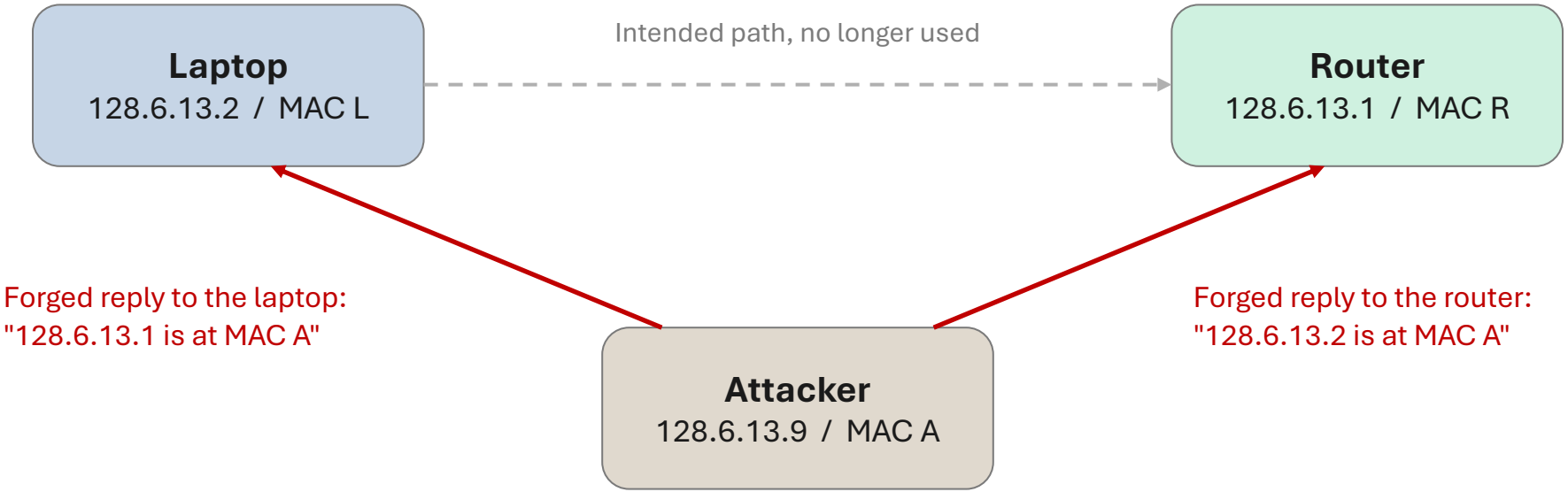
- Requires: the attacker on the victim's local network
- ARP does not authenticate replies: a host updates its cache from any reply, often whether or not it asked
  - A **gratuitous ARP** (an unrequested announcement) is normal: devices send one when they boot
- The attacker sends forged replies: "the router's IP address belongs to my MAC address"
  - The victim replaces its cached mapping. Its traffic for the router now goes to the attacker
- Also called **ARP spoofing**

See *Ettercap* – a multipurpose sniffer/interceptor/logger
https://github.com/Ettercap/ettercap

ARP Poisoning: One Laptop, One Router

Laptop's ARP cache
128.6.13.1 → MAC R **→ MAC A**

Router's ARP cache
128.6.13.2 → MAC L **→ MAC A**



Every packet between the laptop and the router now goes through the attacker, who forwards it on. The laptop doesn't see any disruption.

What The Attacker Gains, And What TLS Limits

- The attacker now **reads** anything unencrypted, **alters** anything unauthenticated, and can **drop or delay** anything
  - This is the adversary-in-the-middle position from the key exchange lecture
- A correctly validated TLS session protects its contents
  - The attacker can neither read nor alter the data
  - The browser rejects a certificate that does not match the site's name
- The attacker still sees which addresses the victim talks to, reads any unencrypted protocol, and can block the traffic
- **Content protection and availability are different properties**

Defenses Against ARP Cache Poisoning

- **Dynamic ARP Inspection** (DAI) on a managed switch
  - The switch checks every ARP reply against a table of known IP-to-MAC bindings and drops replies that do not match
  - The table is usually built by watching DHCP traffic (described shortly)
- **Static ARP entries** for critical hosts
  - Replies cannot overwrite them
  - Does not scale to networks where devices come and go
- **Encryption** limits what a successful attacker can read or change

IPv6 Neighbor Discovery

Goal: the same as ARP poisoning, redirecting a victim's traffic

- IPv6 replaced ARP with the **Neighbor Discovery Protocol** (NDP)
  - It trusts every message on the local network just as ARP does, including the message that tells hosts which router to use
- An attacker can claim a neighbor's address to intercept its traffic, or claim to be the router
- Works even on networks that never deployed IPv6, since hosts still listen to those messages
- Defense: NDP inspection on managed switches, equivalent to the ARP and DHCP checks

DHCP Spoofing

Hand the victim a configuration the attacker chose

DHCP: Dynamic Host Configuration Protocol

- A computer joins a network and needs to be configured
  - It broadcasts a DHCP Discover message
- A DHCP server offers:
  - An IP address
  - The subnet mask
  - The default router (gateway)
  - DNS servers
  - A lease time
- The client selects an offer, usually the first to arrive, and confirms it with that server
- **Nothing identifies an authorized server**

DHCP Spoofing

Goal: control where the victim's traffic goes and whose name lookups it trusts

- Requires: the attacker on the victim's local network
- The attacker runs a **rogue DHCP server** and answers requests
  - If the victim selects the attacker's offer (usually: the attacker answered first), the victim configures itself as the attacker says
- A **false default gateway**: all the victim's outbound traffic goes through the attacker (the ARP poisoning position, with one exchange)
- A **false DNS server**: the attacker controls every name lookup

DHCP Starvation

Goal: keep devices off the network, or force them to the rogue server

- A separate attack on the same protocol
- The attacker requests leases for thousands of invented MAC addresses
  - The legitimate server runs out of addresses
- New devices cannot join the network
  - Or they accept the rogue server's offer, because it is the only one
- Often combined with DHCP spoofing

Defending Against DHCP Attacks

- **DHCP snooping** on a managed switch
  - Ports leading to the real DHCP server are marked **trusted**
  - DHCP server replies arriving on any other port are dropped
- The switch records every lease in a **binding table**: IP address, MAC address, port
  - Dynamic ARP Inspection uses this table to validate ARP replies
- Blocking rogue replies does not stop starvation: the attacker sends requests, not replies
  - Port security and a cap on DHCP requests per port limit starvation

VLAN Hopping

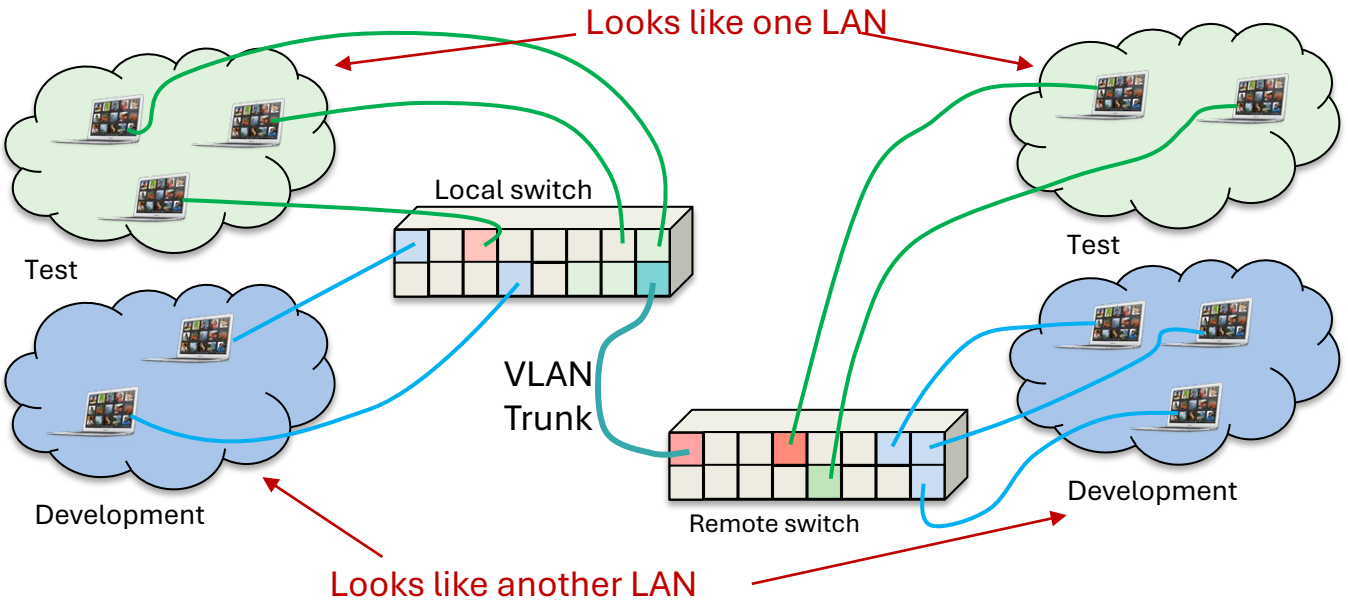
Reach a network segment that was supposed to be isolated

VLANs

- A switch and cables make a local area network (LAN)
- Organizations divide one physical network into **virtual LANs** (VLANs)
  - Accounting, engineering, and guest devices on separate logical networks over the same switches
  - The administrator assigns each switch port to a VLAN
  - Each VLAN is a separate broadcast domain
- Traffic between VLANs must go through a router or firewall
  - The access rules that make the separation useful are configured there

VLAN Trunking

- VLANs span multiple switches and locations
  - A trunk forwards frames for several VLANs between two switches; each frame has a tag naming its VLAN

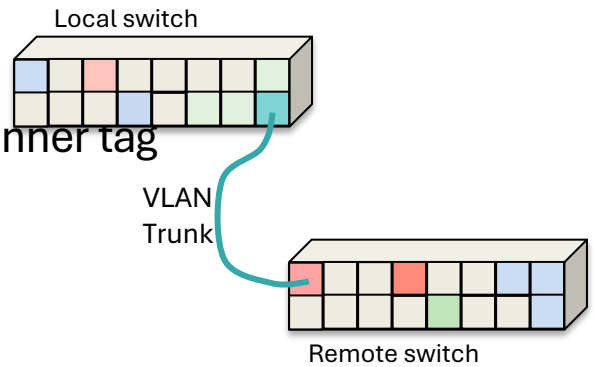


VLAN Hopping Attack

Goal: send frames into a VLAN the attacker is not a member of

Attacker needs: a port on a switch with permissive defaults

- **Switch spoofing**: negotiate a trunk with a switch that auto-negotiates
- **Double tagging**:
  - Attacker sends a frame with two tags
  - The first switch strips the outer tag (the native VLAN)
  - The next switch delivers to the VLAN in the inner tag



Avoiding VLAN Hopping (Requires a Managed Switch)

Disable	Disable unused ports & assign them to an unused VLAN • Stops an attacker from plugging a device into an unused port
Disable	Disable auto-trunking • Stops an attacker from masquerading as a switch
Configure	Explicitly configure trunking on switch ports that are used for trunks • Allows legitimate connected switches to work

Rogue Access Points: The Evil Twin

Goal: become the gateway without joining the victim's network

Attacker requires: radio range of the victims

- A Wi-Fi network is identified by a name anyone can broadcast. On an open network, there is nothing else to check
- The attacker sets up an access point with the same name. A victim connects to it
  - The attacker runs the DHCP and DNS servers the victim is told to use
  - A fake login page collects whatever the victim types into it
- TLS still protects validated sessions. A password-protected network stops an attacker who lacks the password
  - Enterprise Wi-Fi authenticates the network to the client, if the client validates the server's certificate

What Each Attack Requires And Produces

Attack	Required access	What is exploited	Result
CAM table overflow	A port on the switch	The switch learns any source address	Copies of other devices' frames
ARP cache poisoning	Same local network	Hosts accept unauthenticated ARP replies	Victim's traffic routed through the attacker
NDP spoofing	Same local network	Hosts accept unauthenticated neighbor and router messages	Victim's traffic routed through the attacker
DHCP spoofing	Same local network	Clients cannot verify a DHCP server	Attacker chooses the victim's gateway and DNS server
DHCP starvation	Same local network	The address pool is finite	New devices cannot join
VLAN hopping	A port on a permissive switch	Trunk negotiation or native VLAN handling	Frames cross a segment boundary
Evil twin	Radio range, and a victim who connects	A network name proves nothing	Victim's traffic routed through the attacker

The End