8.1 SNIFFING OVERVIEW

- Sniffing Overview
- Types of Sniffing
- Protocols Vulnerable to Sniffing

WHAT IS SNIFFING?

- Sniffing is the act of capturing (recording) traffic flowing through a network
- It is the network equivalent of wiretapping
- Sniffing allows you to identify hosts, services, device types, protocols, subnets, IP addresses, etc. on the network
- A good sniffer can capture nearly any protocol, even ones it does not recognize
  - For example, Wireshark supports thousands of protocols

WHAT IS SNIFFING? (CONT'D)

- Encrypted packets can also be sniffed
  - You won't be able to read their contents unless you can decrypt them
  - However, you can still read:
    - Source and destination addresses and ports
    - SSID, authentication handshakes and initialization vectors for wireless networks
    - VPN handshake information
- Two conditions must be met for sniffing to be effective:
  - Sniffer interface must be in promiscuous mode
  - Traffic to be captured must be forwarded to, or pass by, the sniffer's interface
    - You need to be on a shared segment such as a hub or Wi-Fi channel
    - You can spoof the switch into copying frames out your switchport

HOW SNIFFING WORKS

- The sniffing app puts the device network interface in promiscuous mode
- The app starts capturing all traffic that reaches the interface, regardless of who it's destined for
- You can stop the capture at any time and:
  - Filter the results based on protocol, port, IP address, or payload key word
  - Perform some analysis on the traffic
  - Recreate entire TCP conversations
  - Recreate certain file types
  - Save the captured traffic in a pcap file for later analysis

Windows needs the WinPcap driver to be able to put a NIC in promiscuous mode

NETWORK SNIFFING THREATS

- Many organizations do not put any restrictions on unused switchports
  - Someone can plug in any device using an Ethernet cable
- Sniffing allows the attacker to:
  - identify potential targets
    - hostnames, device types, IP addresses, MAC addresses, ports, protocols, services
  - capture credentials
  - read private messages
  - eavesdrop on voice and video calls
  - recreate files
  - and more

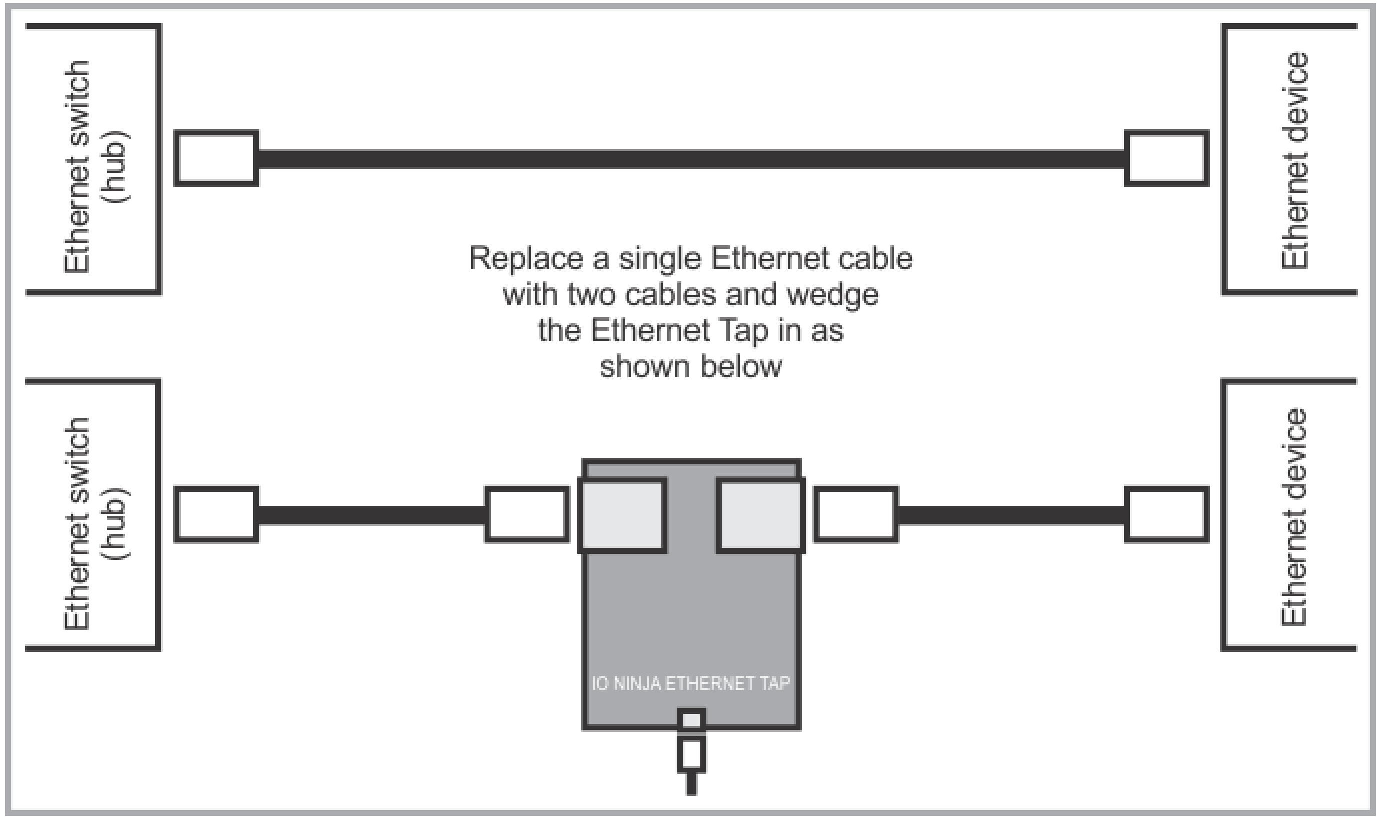
ACTIVE AND PASSIVE SNIFFING

- Passive sniffing involves collecting packets as they pass by your network interface
  - You don't transmit anything
  - You just promiscuously receive
- Active sniffing involves sending out multiple network probes to achieve an objective. Examples:
  - MAC flooding
  - DNS poisoning
  - ARP poisoning
  - DHCP attacks
  - Switch port stealing
  - Spoofing attacks

ADDITIONAL ACTIVE SNIFFING METHODS

- Port spanning
  - Switch configuration that makes the switch send a copy of all frames from other ports to a specific port
  - AKA span port or port mirroring
  - 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
  - Purpose-built hardware device that sits in a network segment between two appliances (router, switch or firewall)
  - Allows you to capture all traffic passing through it

NETWORK TAP EXAMPLE



Replace a single Ethernet cable with two cables and wedge the Ethernet Tap in as shown below

SNIFFING SCENARIO

- Moo connects to the hotel's wireless network to send emails to some of his clients.
- The next day, Moo notices that additional emails have been sent out from his account without consent.
- So what happened?
- If Moo used HTTP instead of HTTPS to sign into his webmail, an attacker could have sniffed it and logged in as him.
- If Moo used Outlook or some email app, if he sent in clear text his SMTP login could have been compromised.
- It is also possible that the additional emails had a spoofed source address.
  - We would have to investigate the email headers to see if Moo was the actual sender or not.

LAWFUL INTERCEPTION

- Legal interception of data communication between end-points
- Some jurisdictions, like the US, require a court order
- For surveillance on traditional phone, VoIP, data, multi-service networks
- **PRISM** - System used by NSA to collect internet communications from various U.S. internet companies

WIRETAPPING

- Process of third-party monitoring of phone/Internet conversations
- Attacker connects a listening device to a circuit between two hosts/phones
- Often covert
- Attack can monitor, access, intercept, and record information
- Types of Wiretapping:
  - Active Wiretapping – Monitors/reads and injects something into communication/traffic
  - Passive Wiretapping – Only monitors/reads/records data

EAVESDROPPING

- Secretly listening to private conversations or communications
- Capture speech or telephone conversations
- Plant a sniffer on a network
- Secretly place a camera or microphone in a room
- Capture VoIP packets off the network and replay them
- Use a phone to record someone entering a password or PIN from across a room
- Use a Wi-Fi Pineapple or other man-in-the-middle device to capture wireless traffic
- Use an IMSI-catcher man-in-the-middle device to intercept cell phone calls

PROTOCOL VULNERABILITIES

- Many protocols are transmitted in clear text (unencrypted)
- Vulnerabilities include:
  - Disclosure of usernames, passwords, host names, IP addresses, sensitive data
  - Keystrokes that provide user names/passwords
  - Reconstructing/capturing files including documents, images, voice, video

TCP/IP CORE PROTOCOLS VULNERABLE TO SNIFFING

- ARP
- IGMP
- ICMP
- TCP shows sequence numbers (usable in session hijacking)
- TCP and UDP show open ports
- IP (both versions) shows source and destination addresses

All six of the core TCP/IP protocols are clear text and vulnerable to sniffing.

VULNERABLE LAYER 7 PROTOCOLS (TCP)

Clear Version	TCP Port	Encrypted Replacement	TCP Port
FTP	21	SFTP (part of SSH) FTPS	22 990
Telnet	23	SSH	22
SMTP	25	SMTP/SSL or TLS	587, 465 (previous)
DNS (zone transfer)	53	--	--
HTTP	80	HTTPS SHTTP (obsolete)	443
POP3	110	POP/SSL or TLS	995
NNTP	119	NNTP/SSL or TLS	563, 443
SMBv1	139	SMBv3	445
IMAP4	143	IMAP/SSL or TLS	993
LDAP	389	LDAPS	683
SQL	1433	SQL/SSL or TLS	1433

VULNERABLE LAYER 7 PROTOCOLS (UDP)

Clear Version	UDP Port	Encrypted Replacement	UDP Port
TFTP	69	--	--
SNMP v1-2c	161, 162	SNMP v3	161, 162
NTP	123	(Best practices recommend adding authentication, and encryption)	--
DNS	53	(DNSSEC recommended to add integrity to records)	--
IKE	500	--	--
SIP	5060, 2000 Cisco Call Manager	SIP-TLS	5061
RTSP (SIP competitor for CCTV)	554	--	--
RTP	5004, 9000, 6970-6999 IETF, 16384-32767)	SRTP	5004+
RTCP		SRTCP	5005

8.2 SNIFFING TOOLS

- Wireshark
- TCPDump
- Wi-Fi Sniffers
- Other Sniffers

WHAT IS A SNIFFER?

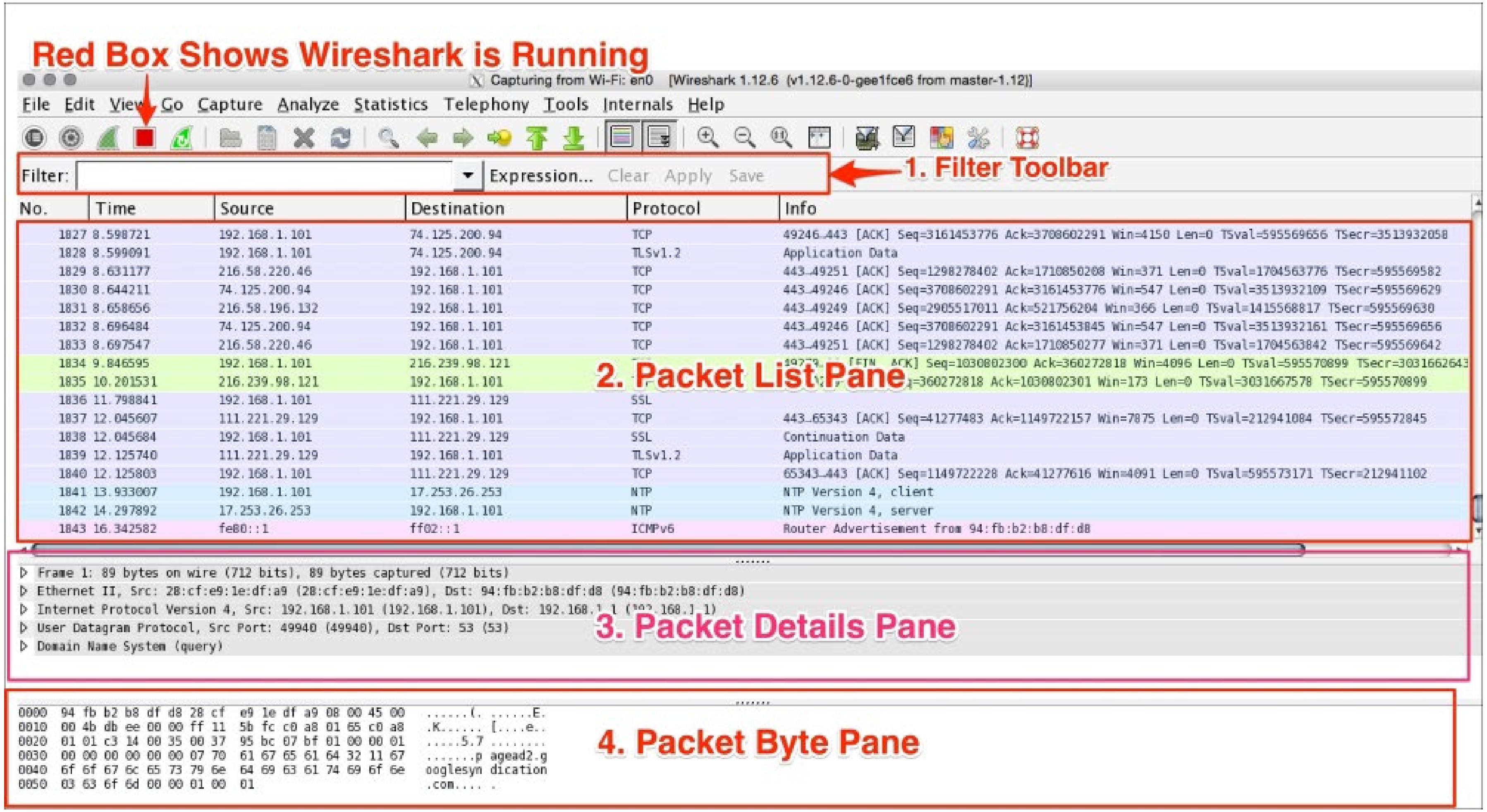
- AKA Protocol Analyzer or Packet Analyzer
- Records all network traffic that reaches its interface
- Can be software- or hardware-based
- Depending on the product, can capture different Layer 2 protocols on various media types
- Typically requires a driver to place the interface in promiscuous mode
  - Allows the sniffer to intake frames even if they are not destined for the sniffing machine

WIRESHARK

- The most popular software-based sniffer
  - Open source
  - Previously known as Ethereal
  - Runs on *nix or Windows
- Captures live traffic from any interface, on different types of media
  - Any protocol including raw packets that are unidentified
  - Follow and recreate entire TCP/HTTP streams
  - Recreate captured files from raw packet hex data
- Has extensive filtering and search capabilities, and packet analysis features
- Can save, export and import packet captures (pcap files)
- With the correct driver, can capture radio and management headers from Wi-Fi

Note: Wireshark is not an IDS or packet crafter

WIRESHARK EXAMPLE



Red Box Shows Wireshark is Running

Capturing from Wi-Fi: en0   [Wireshark 1.12.6 (v1.12.6-0-gee1fce6 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression...
Clear
Apply
Save

1. Filter Toolbar

2. Packet List Pane

```
No.    Time        Source          Destination      Protocol  Info
  1827 8.598721    192.168.1.101   74.125.200.94    TCP       49246…443 [ACK] Seq=3161453776 Ack=3708602291 Win=4150 Len=0 TSval=595569656 TSecr=3513932058
  1828 8.599091    192.168.1.101   74.125.200.94    TLSv1.2   Application Data
  1829 8.631177    216.58.220.46   192.168.1.101    TCP       443…49251 [ACK] Seq=1298278402 Ack=1710850208 Win=371 Len=0 TSval=1704563776 TSecr=595569582
  1830 8.644211    74.125.200.94   192.168.1.101    TCP       443…49246 [ACK] Seq=3708602291 Ack=3161453776 Win=547 Len=0 TSval=3513932109 TSecr=595569629
  1831 8.658656    216.58.196.132  192.168.1.101    TCP       443…49249 [ACK] Seq=2905517011 Ack=521756204 Win=366 Len=0 TSval=1415568817 TSecr=595569630
  1832 8.696484    74.125.200.94   192.168.1.101    TCP       443…49246 [ACK] Seq=3708602291 Ack=3161453845 Win=547 Len=0 TSval=3513932161 TSecr=595569656
  1833 8.697547    216.58.220.46   192.168.1.101    TCP       443…49251 [ACK] Seq=1298278402 Ack=1710850277 Win=371 Len=0 TSval=1704563842 TSecr=595569642
  1834 9.846595    192.168.1.101   216.239.98.121             [FIN, ACK] Seq=1030802300 Ack=360272818 Win=4096 Len=0 TSval=595570899 TSecr=3031662643
  1835 10.201531   216.239.98.121  192.168.1.101              q=360272818 Ack=1030802301 Win=173 Len=0 TSval=3031667578 TSecr=595570899
  1836 11.798841   192.168.1.101   111.221.29.129   SSL
  1837 12.045607   111.221.29.129  192.168.1.101    TCP       443…65343 [ACK] Seq=41277483 Ack=1149722157 Win=7875 Len=0 TSval=212941084 TSecr=595572845
  1838 12.045684   192.168.1.101   111.221.29.129   SSL       Continuation Data
  1839 12.125740   111.221.29.129  192.168.1.101    TLSv1.2   Application Data
  1840 12.125803   192.168.1.101   111.221.29.129   TCP       65343…443 [ACK] Seq=1149722228 Ack=41277616 Win=4091 Len=0 TSval=595573171 TSecr=212941102
  1841 13.933007   192.168.1.101   17.253.26.253    NTP       NTP Version 4, client
  1842 14.297892   17.253.26.253   192.168.1.101    NTP       NTP Version 4, server
  1843 16.342582   fe80::1         ff02::1          ICMPv6    Router Advertisement from 94:fb:b2:b8:df:d8
```

3. Packet Details Pane

```
▷ Frame 1: 89 bytes on wire (712 bits), 89 bytes captured (712 bits)
▷ Ethernet II, Src: 28:cf:e9:1e:df:a9 (28:cf:e9:1e:df:a9), Dst: 94:fb:b2:b8:df:d8 (94:fb:b2:b8:df:d8)
▷ Internet Protocol Version 4, Src: 192.168.1.101 (192.168.1.101), Dst: 192.168.1.1 (192.168.1.1)
▷ User Datagram Protocol, Src Port: 49940 (49940), Dst Port: 53 (53)
▷ Domain Name System (query)
```

4. Packet Byte Pane

```
0000  94 fb b2 b8 df d8 28 cf  e9 1e df a9 08 00 45 00   ......(. ......E.
0010  00 4b db ee 00 00 ff 11  5b fc c0 a8 01 65 c0 a8   .K...... [....e..
0020  01 01 c3 14 00 35 00 37  95 bc 07 bf 01 00 00 01   .....5.7 ........
0030  00 00 00 00 00 00 07 70  61 67 65 61 64 32 11 67   .......p agead2.g
0040  6f 6f 67 6c 65 73 79 6e  64 69 63 61 74 69 6f 6e   ooglesyn dication
0050  03 63 6f 6d 00 00 01 00  01                        .com.... .
```

COMMON 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
- `tcp.port ==21 || tcp.port ==20`
  - Look for TCP 21 or 20, which are used by FTP
- `ip.addr == 10.0.0.165`
  - Look for specific IP address

COMMON WIRESHARK FILTERS (CONT'D)

- `ip.addr == 172.17.15.12 && tcp.port == 23`
  - Display telnet packets for a particular 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`
  - Display HTTP GET requests

COMMON WIRESHARK FILTERS (CONT'D)

- `tcp.port==21`
  - Display FTP packets (unencrypted file transfers)
- `tcp contains string`
  - Displays TCP segments that contain the word "string"
- `tcp.flags==0x16`
  - Filters TCP requests with ACK flag set

TCPDUMP AND WINDUMP

- Tcpdump is a command-line tool for sniffing traffic
  - Similar to Wireshark, but Linux command-line only
  - It captures and displays traffic
  - Good for:
    - Passive fingerprinting
    - Sniffing passwords
    - Intercepting any clear text transmissions
- Syntax
  - `tcpdump flag(s) interface`
  - `tcpdump -i eth1`
    - Puts the specified interface in listening mode
- WinDump is a Windows version similar to tcpdump

PCAP ANALYSIS

- You can send capture files (pcap) from Wireshark, tcpdump, WinDump, EtherPeek, etc. to an analysis tool
- Example tools include:
  - Tcptrace
  - PRTG Network Monitor
  - Wireshark (open dumps from command line tools like tcpdump)
  - NetworkMiner

WI-FI-SPECIFIC SNIFFERS

- Airodump-ng
- Carnivore
- snoop
- SkyGrabber

Note:

- Wireshark will capture Wi-Fi packets from any interface, including wireless
  - Wireshark presents Wi-Fi packets as if they are Ethernet
  - 802.11 LLC header was designed to be directly interoperable with Ethernet
- If you want to see Wi-Fi 802.11 management frames/radio headers:
  - Wireshark: Select "802.11" as the "Link-layer header type" in the "Capture Options" dialog
  - dumpcap, TShark, or Wireshark (capture started from the command line):
    - add the argument `-y IEEE802_11` to the command

HARDWARE PROTOCOL ANALYZER

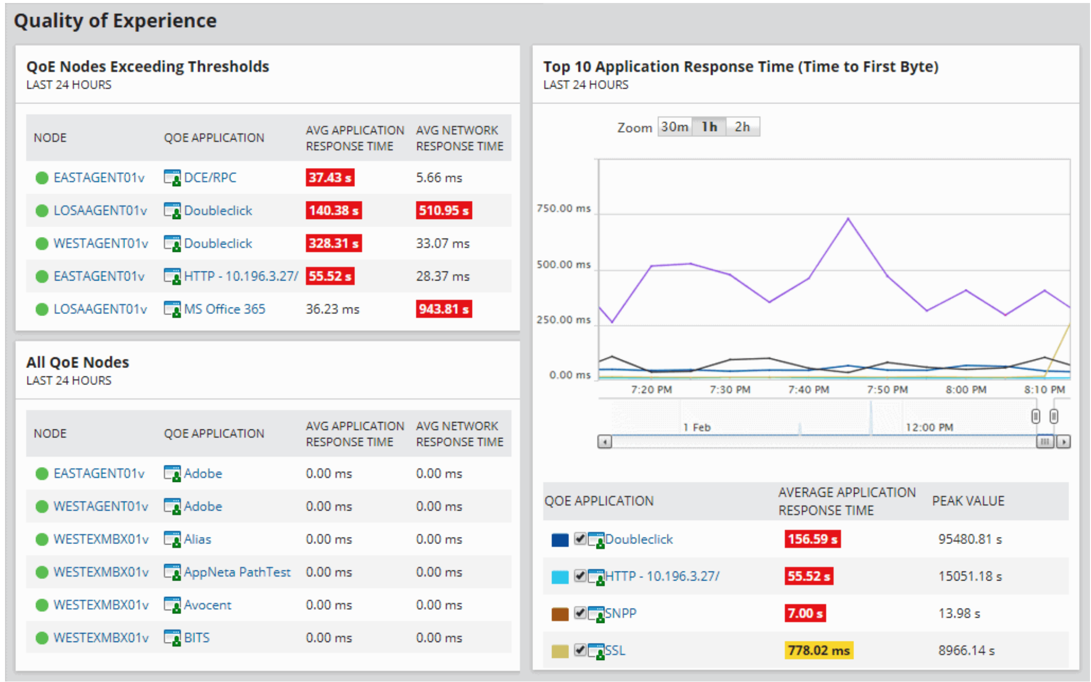
- Equipment that captures signals to monitor network usage
- Does not alter traffic in cable segment
- Identifies malicious network traffic generated via hacking network software
- Grabs data packets
- Decodes and analyzes packet content based on predetermined rules
- Able to view individual bytes of data in each packet passing through cable

HARDWARE PROTOCOL ANALYZER EXAMPLES

- Keysight N2X N5540A
- Keysight E2960B
- RADCOM PrismLite Protocol Analyzer
- RADCOM Prism UltraLite Protocol Analyzer
- FLUKE Networks OptiView XG Network Analyzer
- FLUKE Networks OneTouch AT Network Assistant

ADDITIONAL SNIFFING TOOLS

- SolarWinds Deep Packet Inspection and Analysis Tool
- ManageEngine NetFlow Analyzer
- Paessler Packet Capture Tool
- Omnipeek Network Protocol Analyzer
- tshark
- NetworkMiner
- Fiddler
- Capsa



Quality of Experience

QoE Nodes Exceeding Thresholds
LAST 24 HOURS

NODE	QOE APPLICATION	AVG APPLICATION RESPONSE TIME	AVG NETWORK RESPONSE TIME
● EASTAGENT01v	DCE/RPC	37.43 s	5.66 ms
● LOSAAGENT01v	Doubleclick	140.38 s	510.95 s
● WESTAGENT01v	Doubleclick	328.31 s	33.07 ms
● EASTAGENT01v	HTTP - 10.196.3.27/	55.52 s	28.37 ms
● LOSAAGENT01v	MS Office 365	36.23 ms	943.81 s

All QoE Nodes
LAST 24 HOURS

NODE	QOE APPLICATION	AVG APPLICATION RESPONSE TIME	AVG NETWORK RESPONSE TIME
● EASTAGENT01v	Adobe	0.00 ms	0.00 ms
● WESTAGENT01v	Adobe	0.00 ms	0.00 ms
● WESTEXMBX01v	Alias	0.00 ms	0.00 ms
● WESTEXMBX01v	AppNeta PathTest	0.00 ms	0.00 ms
● WESTEXMBX01v	Avocent	0.00 ms	0.00 ms
● WESTEXMBX01v	BITS	0.00 ms	0.00 ms

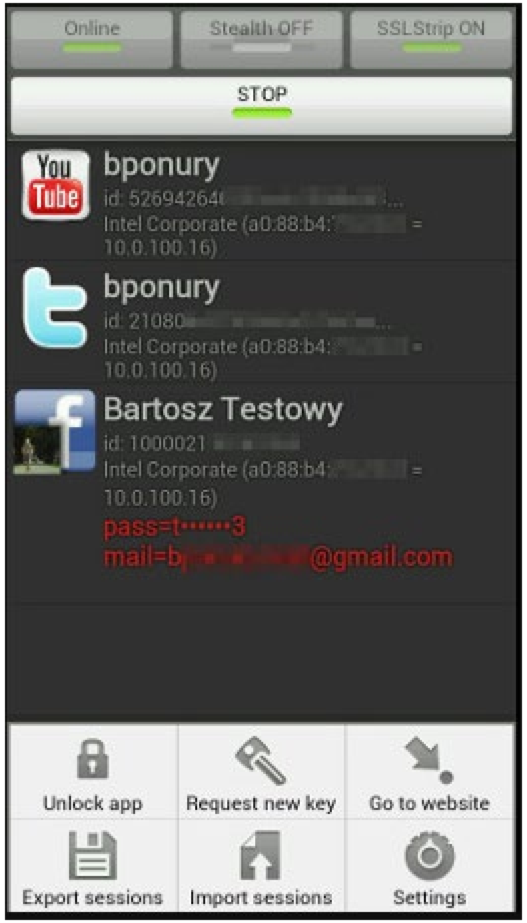
Top 10 Application Response Time (Time to First Byte)
LAST 24 HOURS

QOE APPLICATION	AVERAGE APPLICATION RESPONSE TIME	PEAK VALUE
■ ☑ Doubleclick	156.59 s	95480.81 s
■ ☑ HTTP - 10.196.3.27/	55.52 s	15051.18 s
■ ☑ SNPP	7.00 s	13.98 s
■ ☑ SSL	778.02 ms	8966.14 s

SNIFFING TOOLS FOR MOBILE DEVICES

- Wi.cap.Network Sniffer Pro
- FaceNiff
- Sniffer
- zAnti
- cSploit
- Packet Capture
- Debug Proxy
- WiFinspect
- tPacketCapture
- Android tcpdump

Note: Many mobile sniffer apps require root access (you will have to root or jailbreak your device)



bponury
id: 52694264[redacted]...
Intel Corporate (a0:88:b4:[redacted] =
10.0.100.16)

bponury
id: 21080[redacted]...
Intel Corporate (a0:88:b4:[redacted] =
10.0.100.16)

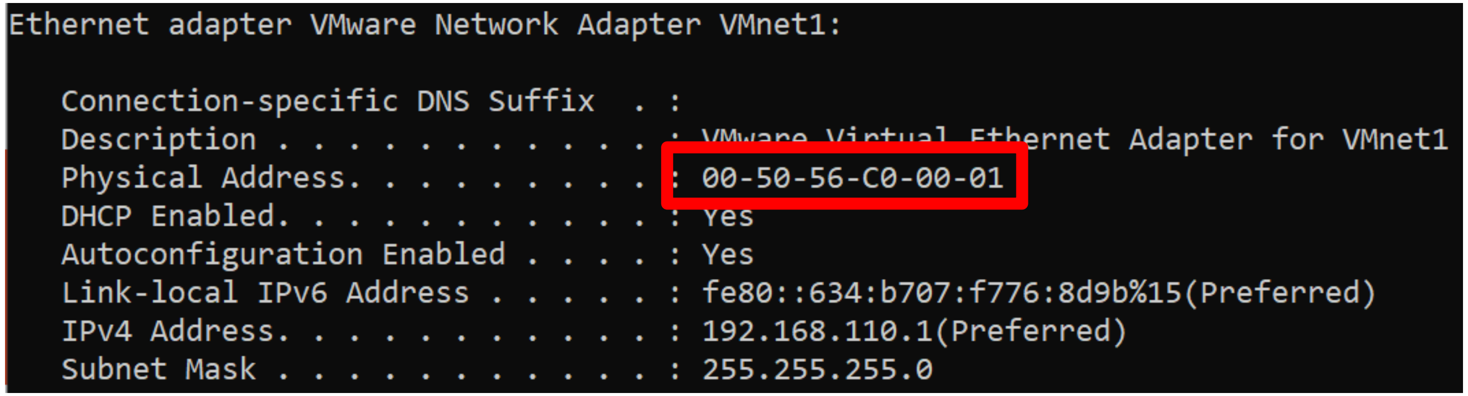
Bartosz Testowy
id: 1000021 [redacted]
Intel Corporate (a0:88:b4:[redacted] =
10.0.100.16)
pass=t•••••••3
mail=b[redacted]@gmail.com

8.3 MAC AND ARP ATTACKS

- MAC Addresses
- MAC Spoofing
- MAC Flooding
- ARP
- ARP Spoofing
- ARP Poisoning

MAC ADDRESS (MAC)

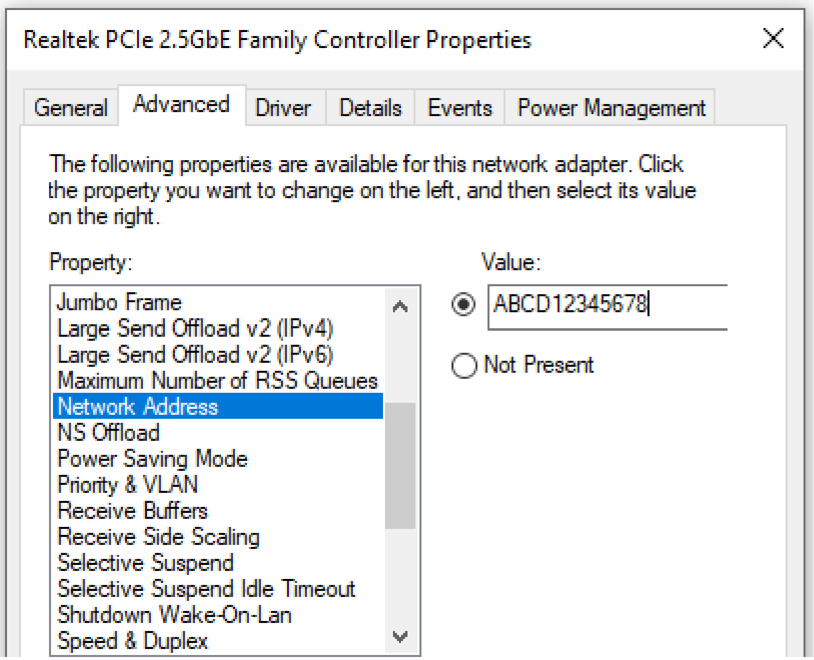
- Physical address of a network interface card (NIC)
- AKA burned-in address
  - Set by the factory – cannot be changed in the NIC firmware
  - Some NIC drivers allow the OS to temporarily override it
- Used to identify a node at Layer 2 on Ethernet and Wi-Fi segments
  - An IP packet must also include the source and destination MAC addresses



```
Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VMware Virtual Ethernet Adapter for VMnet1
   Physical Address. . . . . . . . . : 00-50-56-C0-00-01
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::634:b707:f776:8d9b%15(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.110.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
```

MAC SPOOFING

- Deliberately change the MAC address of your NIC
  - Many OSes can use the NIC driver to temporarily override the MAC address
- Used to:
  - Impersonate another machine
  - Bypass MAC-based access control restrictions
  - Spoof (fool) a switch



Realtek PCIe 2.5GbE Family Controller Properties

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

- Jumbo Frame
- Large Send Offload v2 (IPv4)
- Large Send Offload v2 (IPv6)
- Maximum Number of RSS Queues
- Network Address
- NS Offload
- Power Saving Mode
- Priority & VLAN
- Receive Buffers
- Receive Side Scaling
- Selective Suspend
- Selective Suspend Idle Timeout
- Shutdown Wake-On-Lan
- Speed & Duplex

Value:

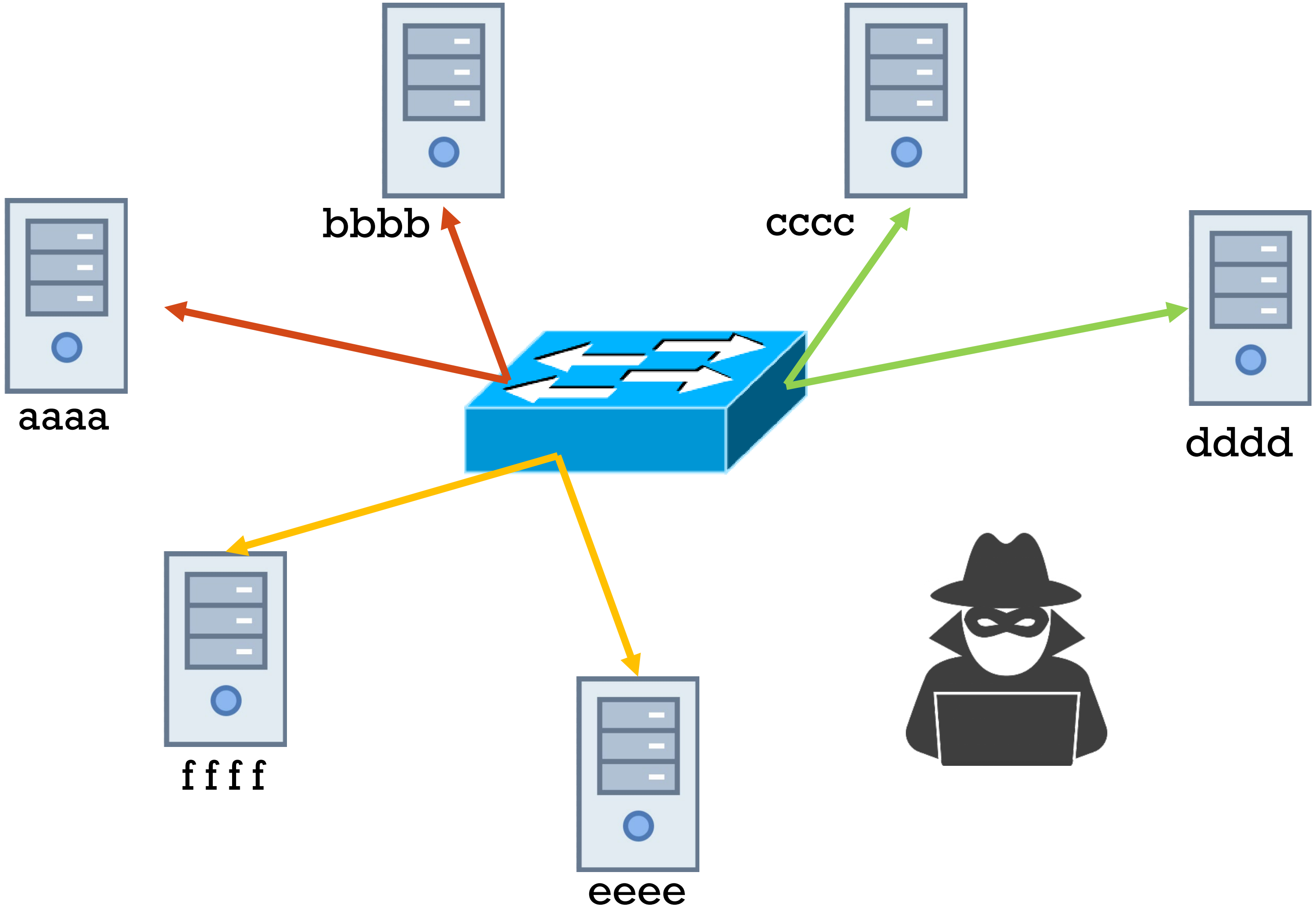
● ABCD12345678

○ Not Present

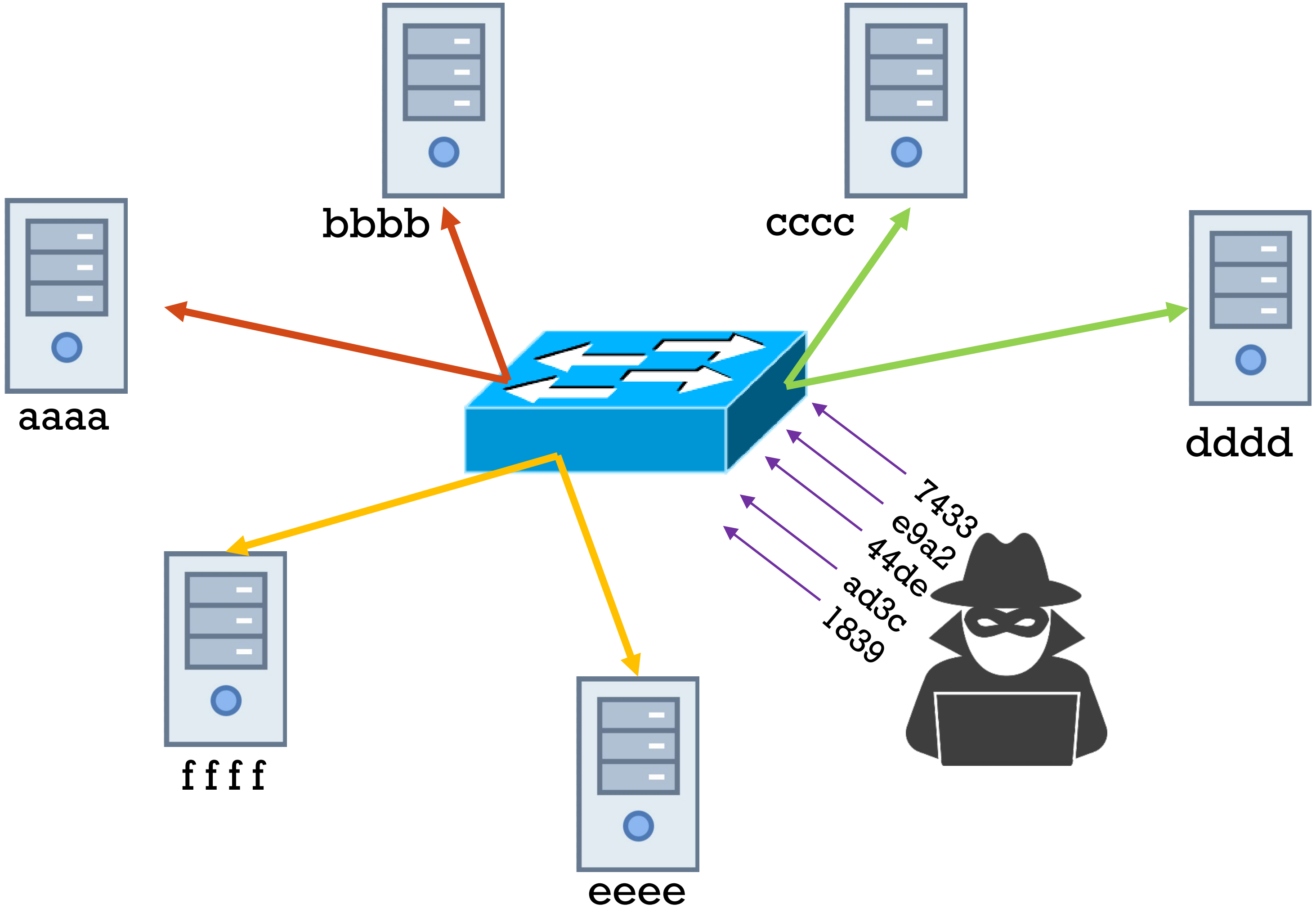
MAC FLOODING

- A common attack on a network switch
- The goal is to force a switch to behave like a hub
  - Forward all frames out all ports
  - The attacker can sniff any traffic
- Intentionally overwhelming a switch with phony MAC addresses
  - Specially crafted Ethernet frames are rapidly sent into a switch port
  - Typically the frames have random spoofed source MAC addresses
- The switch will enter the spoofed MAC addresses into its MAC table
- The MAC table fills and cannot take in any new MAC addresses
- Vulnerable switches will then change into hub mode
  - They repeat any incoming frame out all ports
- This allows the attacker to sniff traffic from all nodes on the switch
- Most modern switches are not vulnerable

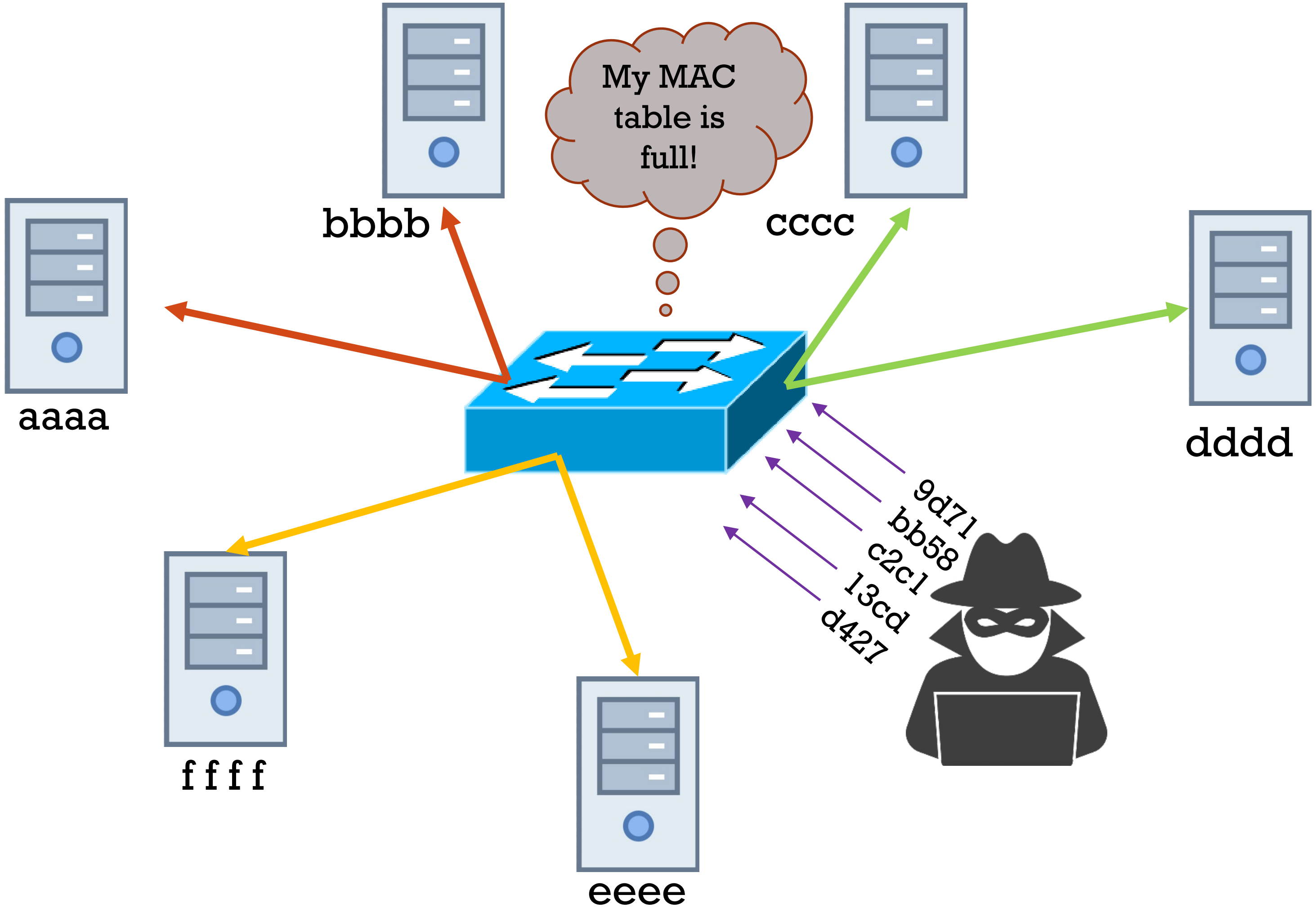
MAC FLOODING EXAMPLE



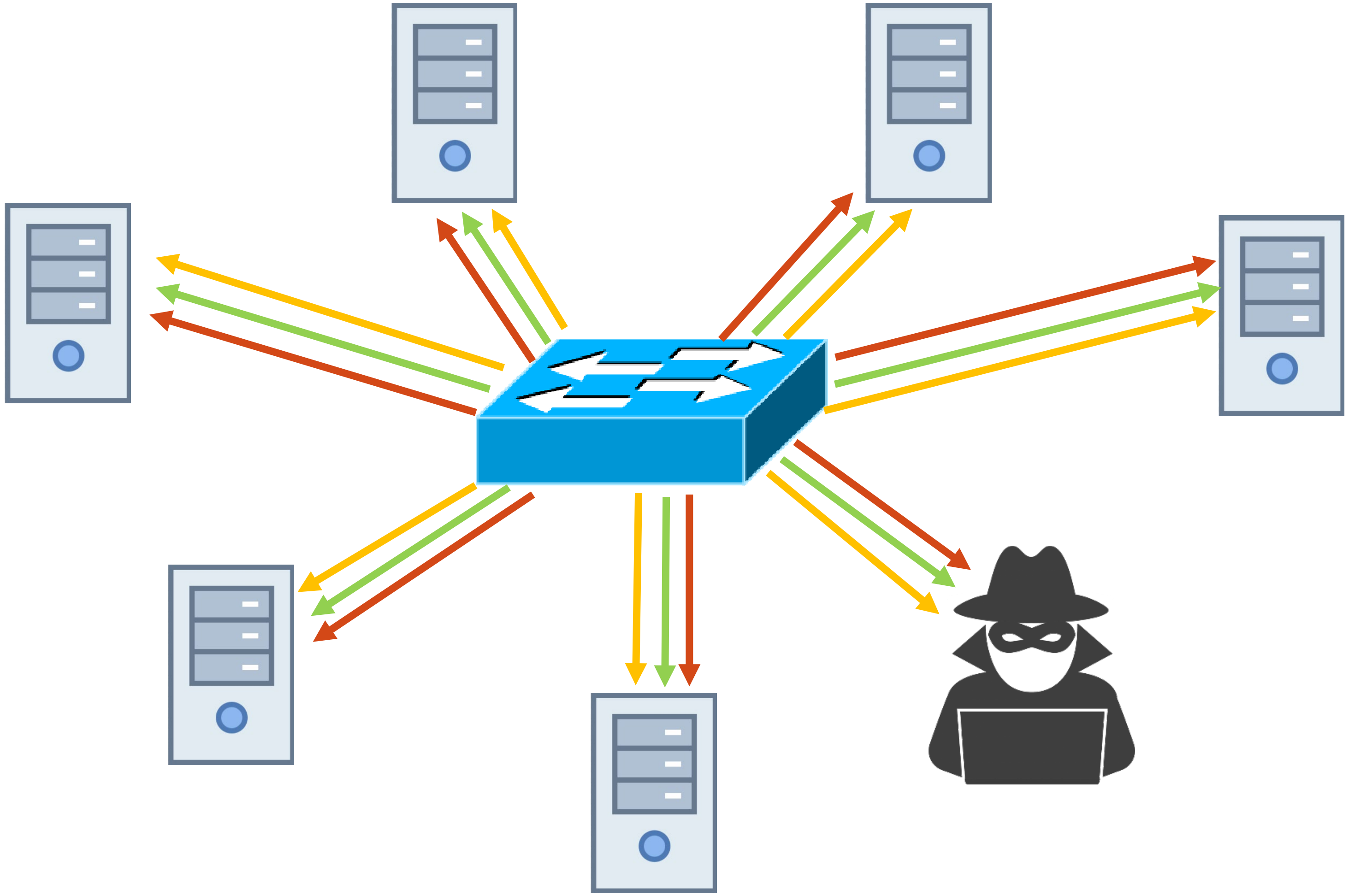
MAC FLOODING EXAMPLE



MAC FLOODING EXAMPLE



MAC FLOODING EXAMPLE



ADDRESS RESOLUTION PROTOCOL (ARP)

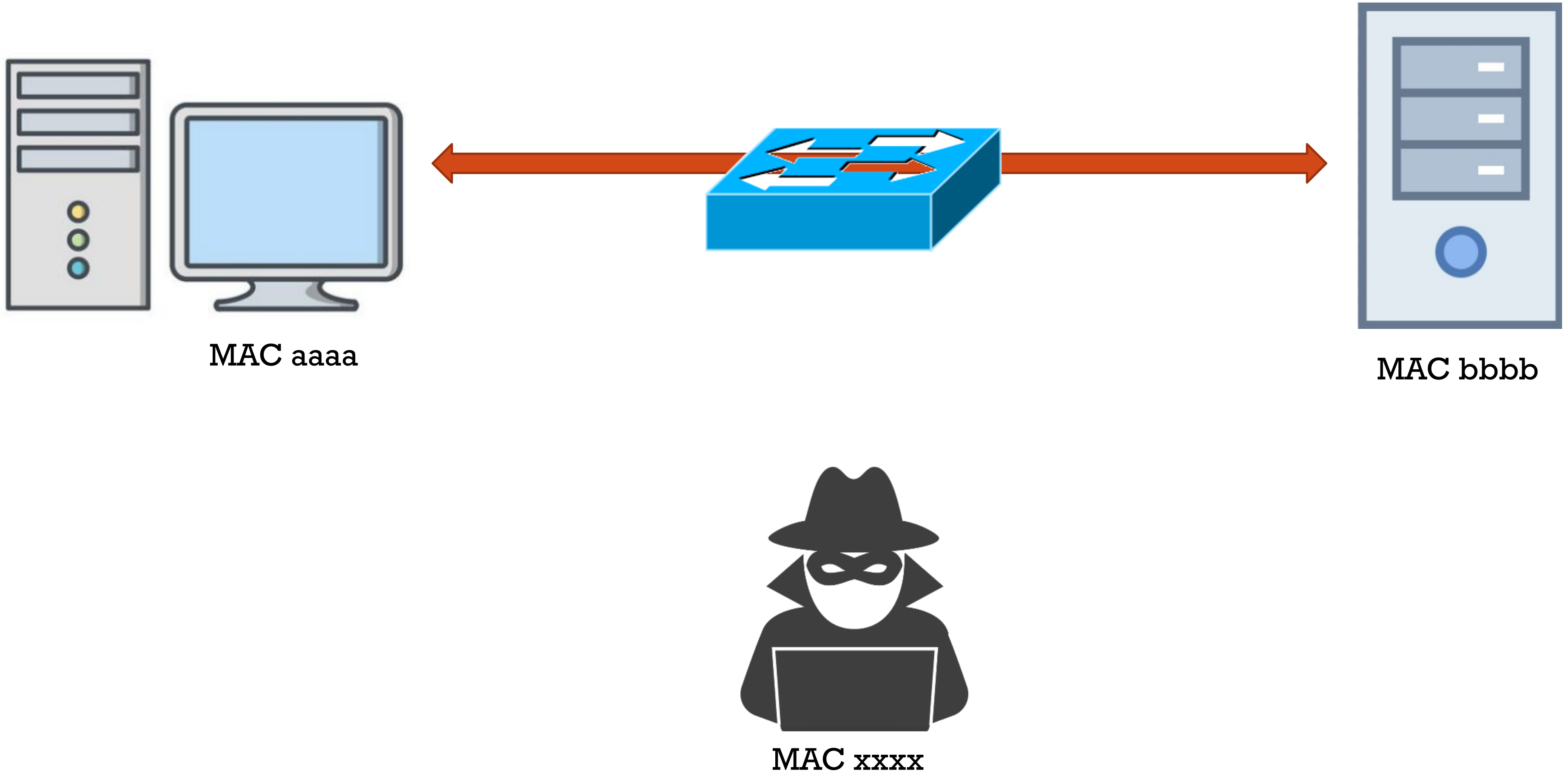
- A core TCP/IP protocol
- Maps MAC addresses to IP addresses
  - In Ethernet and Wi-Fi, you cannot transmit a packet until the Layer 2 header contains the source and destination MAC addresses
- ARP process:
  - Sender transmits an ARP request
    - Layer 2 broadcast (FFFFFFFFFFFF)
    - Asks which MAC "owns" the specified IP address
  - All nodes on the same segment receive and process the request
  - The "owner" sends an ARP reply
    - Layer 2 unicast
    - Affirms it owns the IP address
  - The sender updates its ARP cache, mapping MAC to IP
    - Mappings must be refreshed periodically

ARP SPOOFING

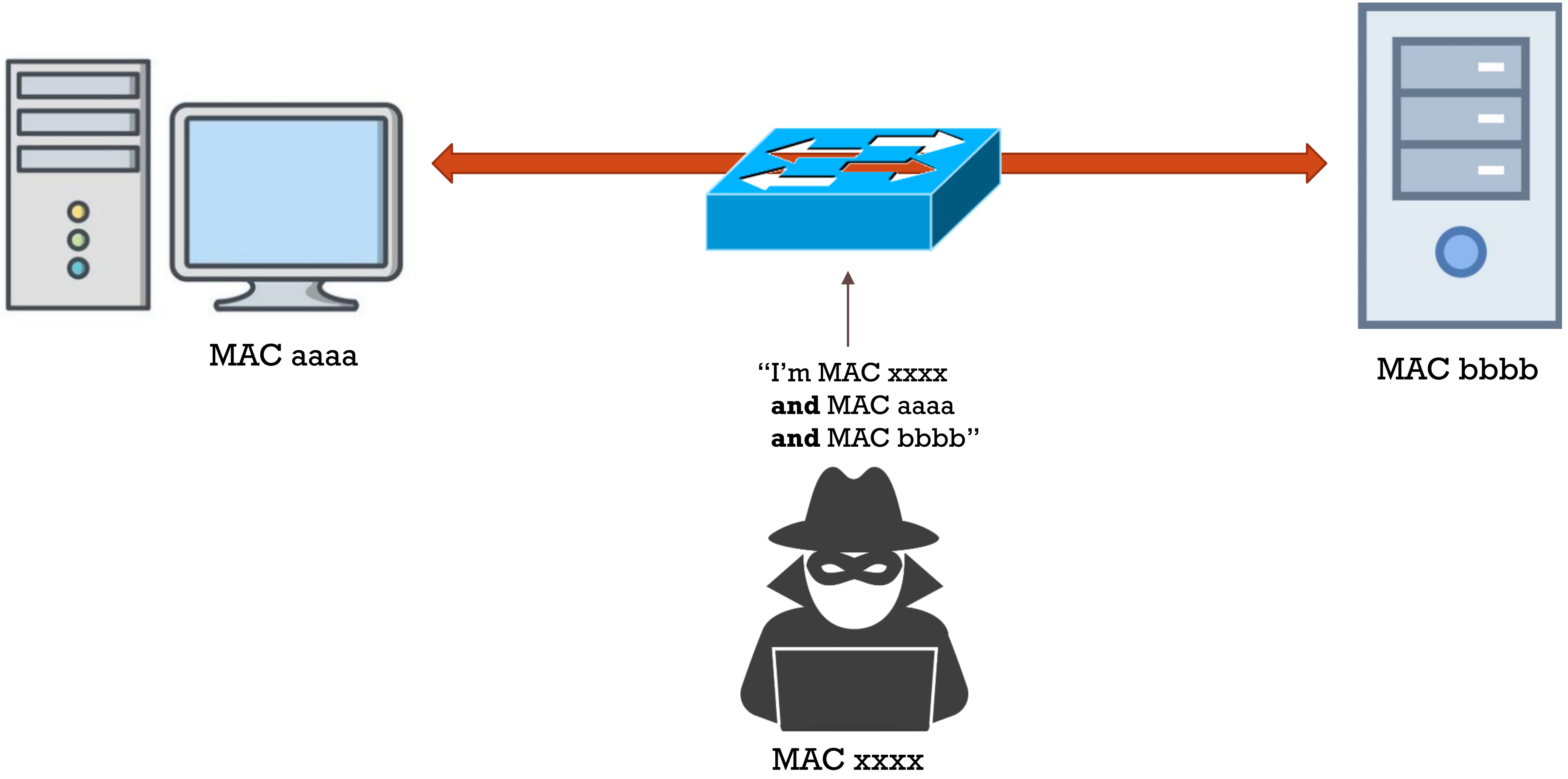
- Used for sniffing someone else's traffic
- Transmit spoofed ARP frames into the switch
  - Pretend to have the same MAC as the node(s) you want to eavesdrop on
  - The IP address is irrelevant, because the switch only deals in MAC addresses
- The switch will add the spoofed MAC to its table, associating it with your port
  - The switch will actually have the same MAC associated with two switchports
- Any traffic destined for the other node will also be forwarded out your port

You use the target's MAC address to fool the switch

ARP SPOOFING EXAMPLE



ARP SPOOFING EXAMPLE



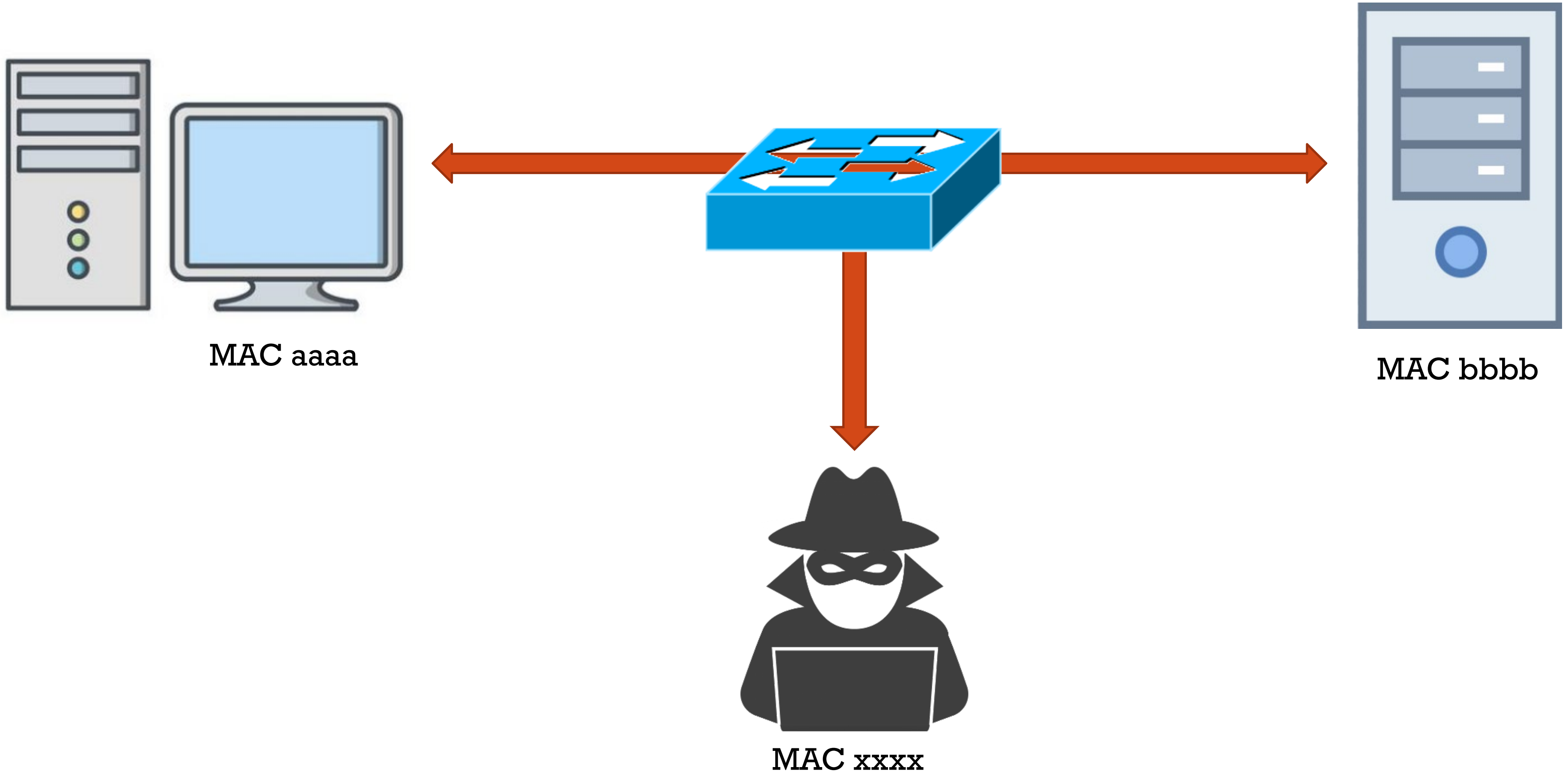
MAC aaaa

“I’m MAC xxxx
and MAC aaaa
and MAC bbbb”

MAC bbbb

MAC xxxx

ARP SPOOFING EXAMPLE

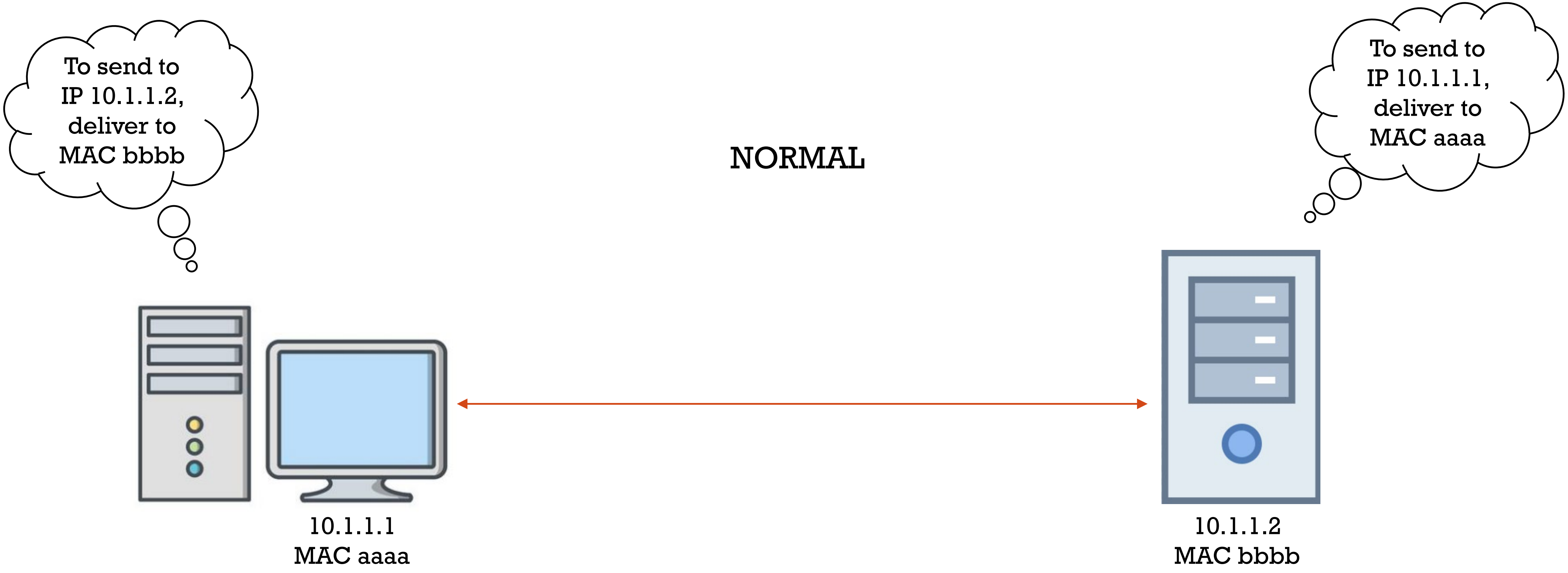


ARP POISONING

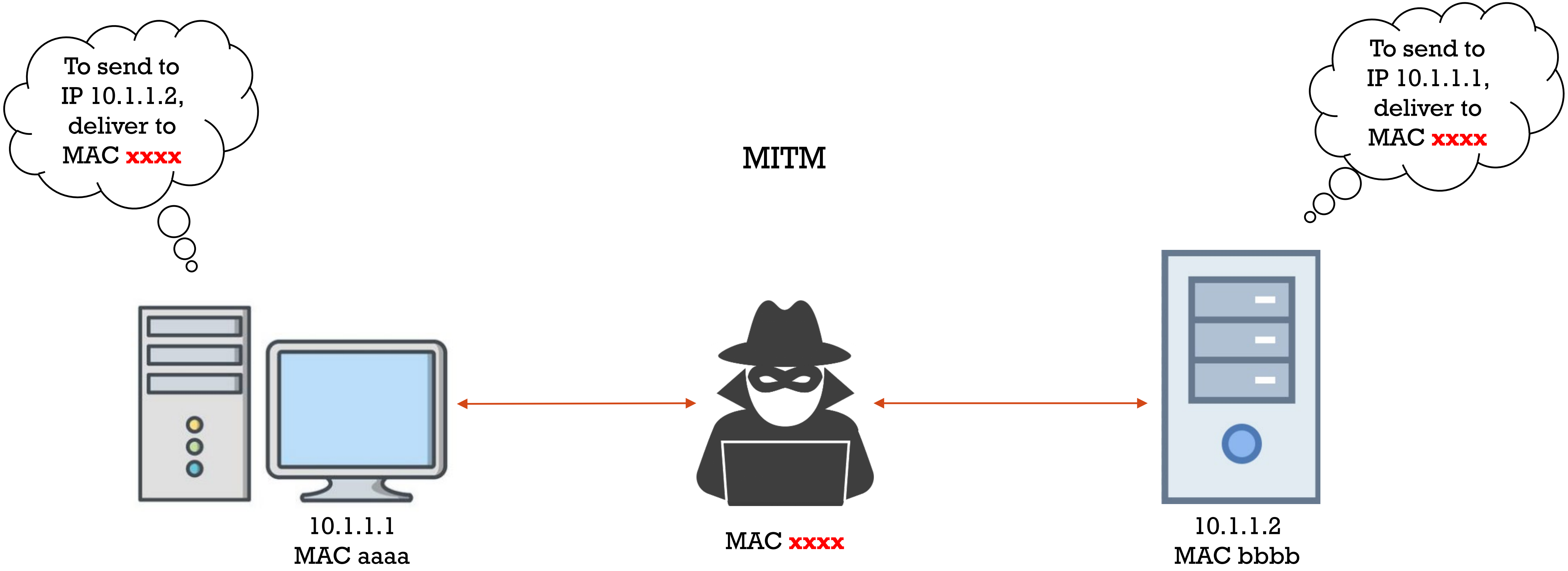
- The deliberate effort to corrupt another device's ARP cache
- Send fake ARP replies that associate attacker's MAC with target's IP
- Used for man-in-the-middle attacks
  - Corrupt both sides of a conversation (client - server / sender - gateway)
  - Each node thinks the other has your MAC address
  - The two sides will unknowingly relay their conversation through you

You use your own MAC address, but associate it with the target's IP address, to fool other devices

ARP POISONING MITM EXAMPLE


To send to IP 10.1.1.2, deliver to MAC bbbb
NORMAL
To send to IP 10.1.1.1, deliver to MAC aaaa
10.1.1.1
MAC aaaa
10.1.1.2
MAC bbbb

ARP POISONING MITM EXAMPLE



To send to IP 10.1.1.2, deliver to MAC xxxx

MITM

To send to IP 10.1.1.1, deliver to MAC xxxx

10.1.1.1
MAC aaaa

MAC xxxx

10.1.1.2
MAC bbbb

8.4 NAME RESOLUTION POISONING

- Name Resolution Process
- DNS Poisoning
- Poisoning Tools
- Poisoning Defense
- NBNS
- LLMNR

WINDOWS NAME RESOLUTION PROCESS

1. Check if the destination is self
2. Check if the name is currently in the DNS resolver cache
3. Check if the name is in the %systemroot%\system32\drivers\etc\hosts file
4. Query the DNS server
5. Send an LLMNR multicast to 224.0.0.252 (IPv6 FF02::1:3), UDP port 5355
6. Send a NetBIOS name query broadcast to 255.255.255.255, UDP port 137

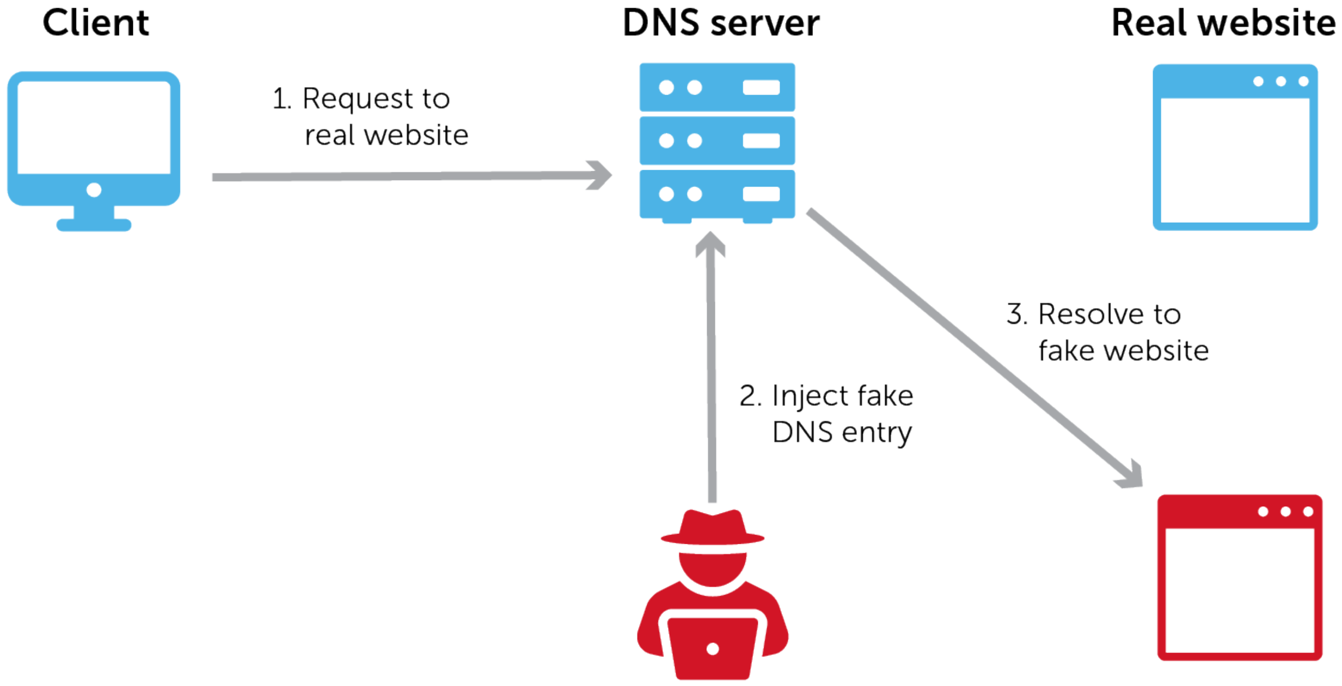
DNS POISONING

- Most DNS servers allow dynamic updates
- Attacker updates a DNS server with a fake A record
  - Destination name is the same
  - IP address has been changed to the attacker's IP
- Server thinks update is legitimate
- When clients perform an A lookup, they are given the wrong IP address
- Can be performed against both Internet and intranet DNS servers

DNS CACHE POISONING

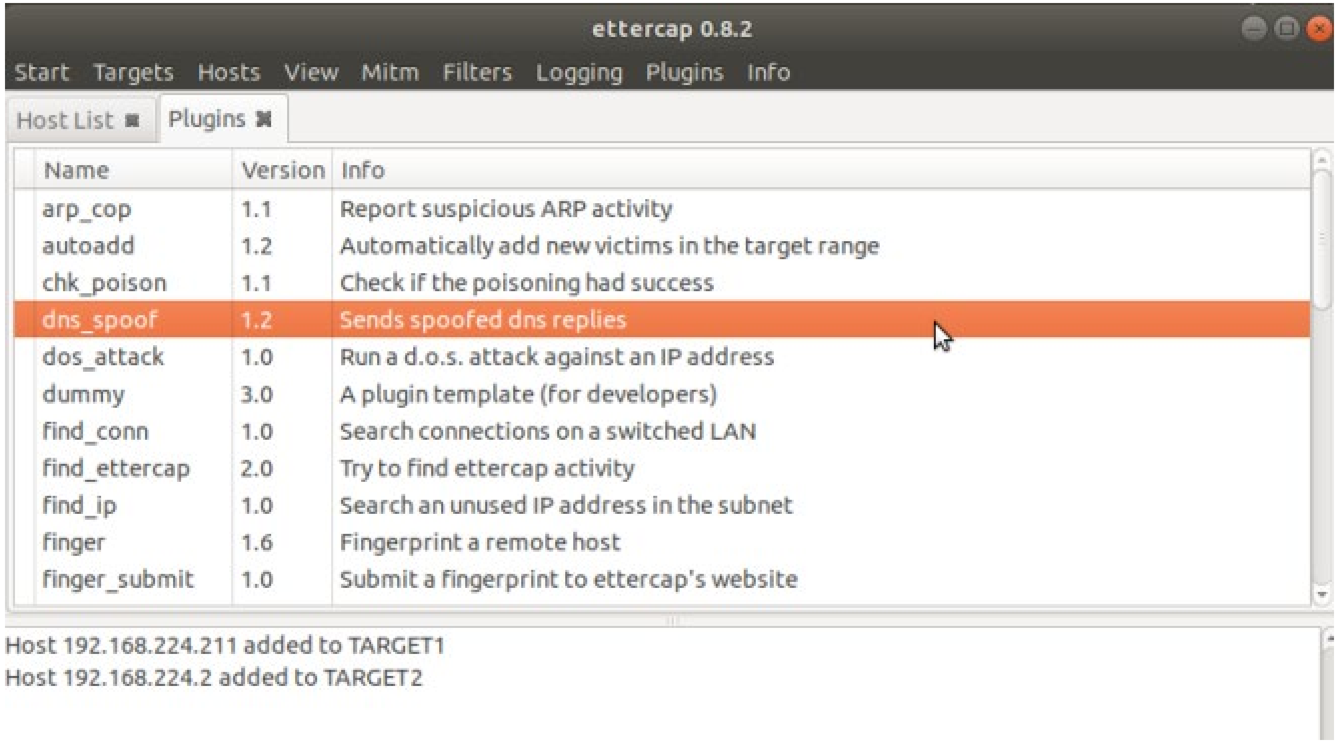
- False DNS records are inserted into a DNS server's cache
  - These records are then given to clients and other DNS servers
- Most DNS servers query other servers to resolve host names
- One false record can propagate to many DNS servers and clients
- Digital signatures and DNSSEC can help, and should be implemented
  - In DNSSEC, a digital signature accompanies each DNS record to prove its authenticity and integrity
  - Reduce the threat of DNS poisoning, spoofing, and similar types of attacks
  - Clients that cannot utilize DNSSEC will ignore the signature files

DNS CACHE POISONING EXAMPLE



DNS POISONING TOOLS

- dns-poisoning-tool (https://github.com/gr3yc4t/dns-poisoning-tool)
- Ettercap
- Bettercap
- dnsspoof



Name	Version	Info
arp_cop	1.1	Report suspicious ARP activity
autoadd	1.2	Automatically add new victims in the target range
chk_poison	1.1	Check if the poisoning had success
dns_spoof	1.2	Sends spoofed dns replies
dos_attack	1.0	Run a d.o.s. attack against an IP address
dummy	3.0	A plugin template (for developers)
find_conn	1.0	Search connections on a switched LAN
find_ettercap	2.0	Try to find ettercap activity
find_ip	1.0	Search an unused IP address in the subnet
finger	1.6	Fingerprint a remote host
finger_submit	1.0	Submit a fingerprint to ettercap's website

```
Host 192.168.224.211 added to TARGET1
Host 192.168.224.2 added to TARGET2
```

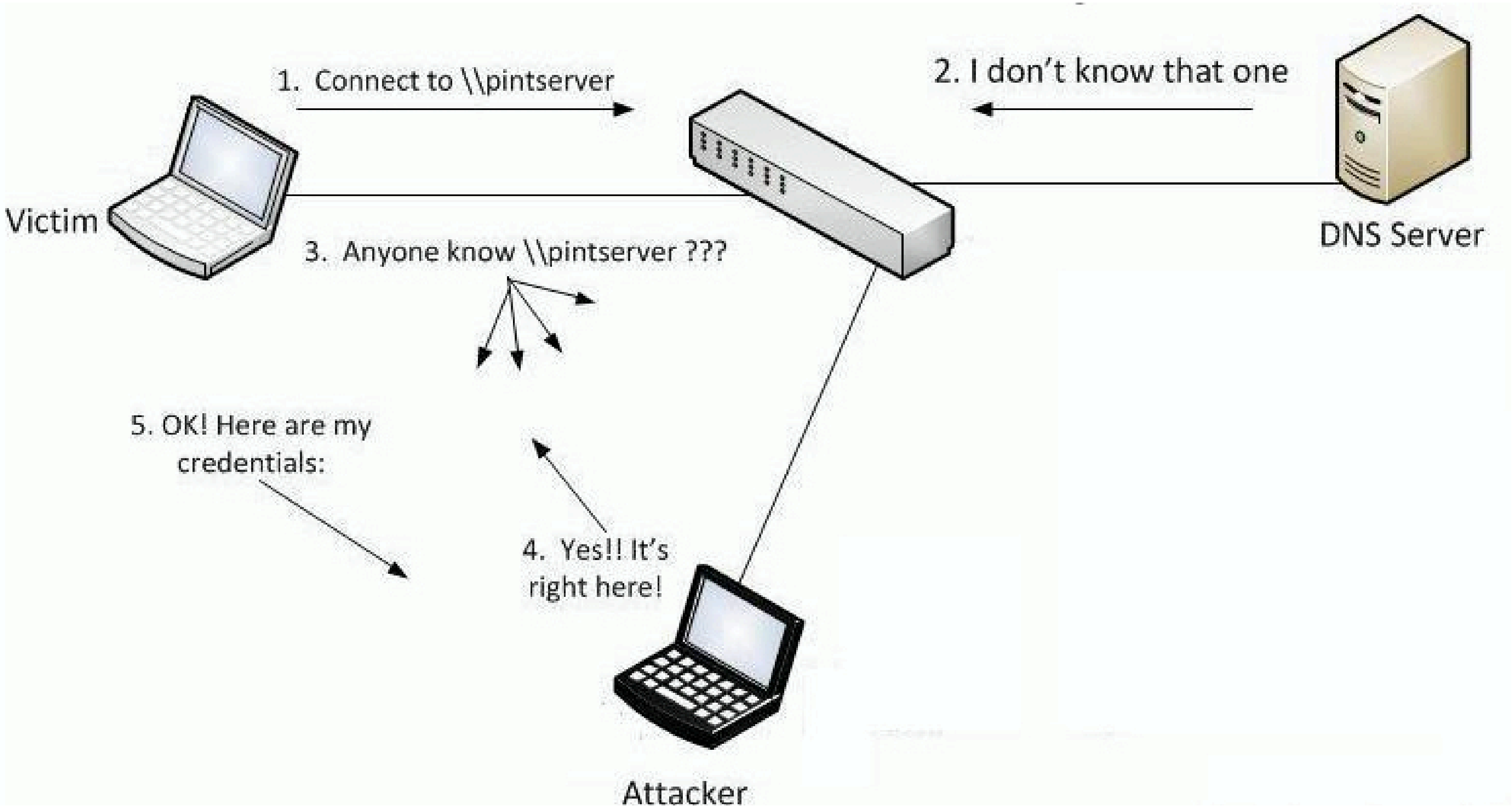
DEFEND AGAINST DNS SPOOFING

- Test your DNS server for poisoning vulnerabilities at:
  - www.dns-oarc.net/oarc/services/dnsentropy
- Keep DNS servers patched
- Configure clients to use your internal DNS server
  - As opposed to Google - you can reduce the risk of DNS MITM
- Hard-code DNS A records where practical (especially server A records)
- Disallow anonymous updates to DNS
  - Client updates
  - Incoming zone transfers
- Configure local DNS server against cache pollution
- Implement IDS to watch for inappropriate update sources
- Implement DNSSEC

NETBIOS NAME RESOLUTION (NBNS)

- Pre-Windows 2000 clients and servers
- Name resolution was performed by querying Microsoft's NetBIOS name server WINS (aka NetBIOS over TCP Name Server)
- NetBIOS name resolution order (configurable)
  1. Check local NetBIOS resolver cache (nbtstat -c)
  2. Query WINS server (UDP 139)
  3. Check local LMHOSTS file
  4. Send NetBIOS broadcast message (UDP 137)
  5. Check DNS resolver cache
  6. Query DNS server
- Link-Local Multicast Name Resolution (LLMNR) replaced NetBIOS
  - Uses multicasting instead of broadcasting
  - Supports IPv4 and IPv6

LLMNR / NBT-NS POISONING EXAMPLE


1. Connect to \\pintserver
2. I don't know that one
Victim
DNS Server
3. Anyone know \\pintserver ???
5. OK! Here are my credentials:
4. Yes!! It's right here!
Attacker

LLMNR / NBT-NS POISONING COUNTERMEASURES

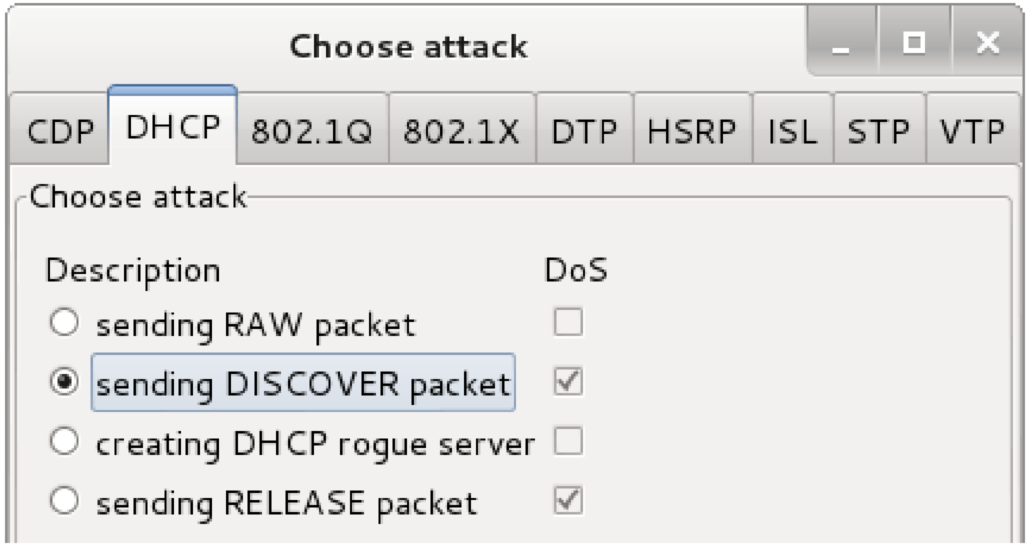
- Disable LLMNR/NetBIOS name queries
- Require all clients to use DNS
- Secure DNS against spoofing

8.5 OTHER LAYER 2 ATTACKS

- DHCP Starvation
- Spanning-Tree Protocol Attacks
- VLAN Hopping

DHCP STARVATION ATTACK

- A flood of fake DHCP Discover messages with spoofed MAC addresses
- The DHCP server makes an Offer to each of the fake clients
- All available IP addresses quickly become reserved for “potential” DHCP clients
- DHCP starvation is often accompanied by a rogue DHCP server and MITM attack



Choose attack

Description	DoS
○ sending RAW packet	□
● sending DISCOVER packet	☑
○ creating DHCP rogue server	□
○ sending RELEASE packet	☑

DHCP STARVATION TOOLS AND MITIGATION

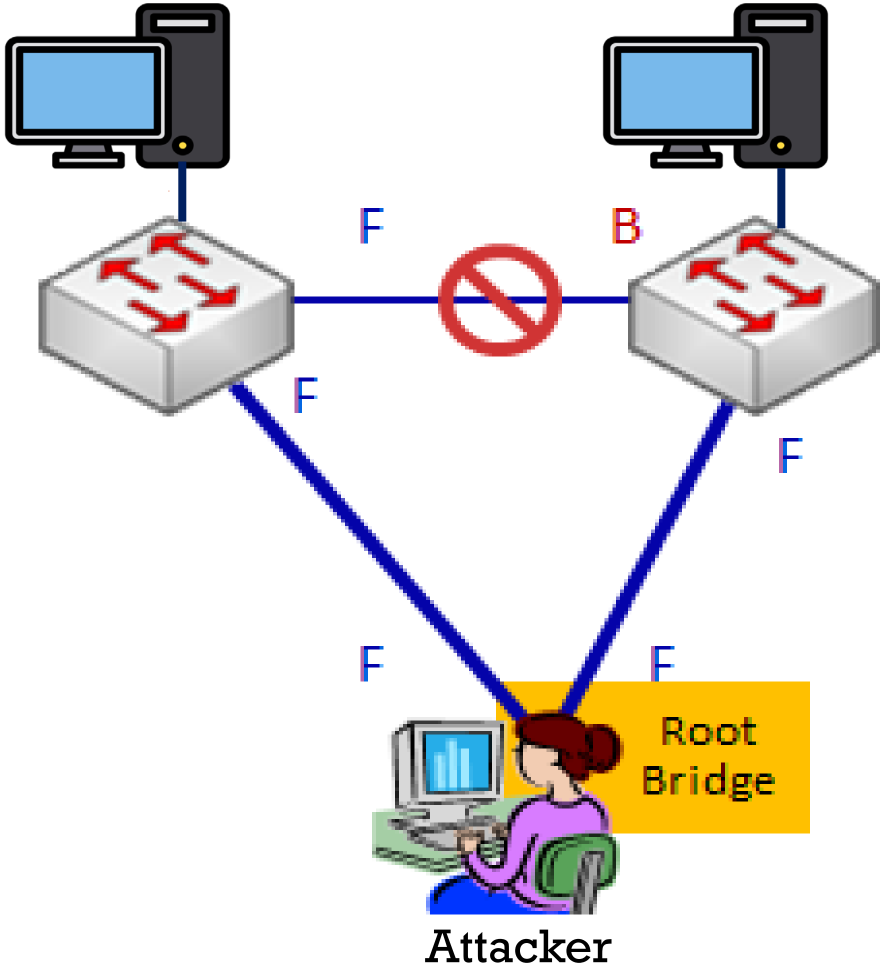
- Attack Tool Examples:
  - Yersenia
  - DHCPstarv
  - A variety of GitHub tools
- Mitigation:
  - Switchport security (restricting the port to only allow one MAC address) may not help
    - Switches monitor nodes on their ports by examining source MAC addresses
  - The DHCP protocol does not use source MAC addresses to identify clients
    - It uses the DHCP DISCOVER CHADDR field in the payload
  - You can configure DHCP snooping on the switch
    - Will block rogue DHCP servers
    - The verify mac-address parameter will also only allow client requests whose payload matches the actual source MAC in the frame
    - `ip dhcp snooping verify mac-address`

SPANNING-TREE PROTOCOL (STP)

- Switching loops are caused by uncontrolled redundant links
- Switching loops will almost instantly bring the network segment to a standstill
  - Links will be flooded with endlessly looping and repeating frames
  - The switch CPU utilization will shoot up to near 100%
  - The switch MAC table will become unstable by constant rapid changes
- Spanning-tree protocol (STP) eliminates switching loops in a switched network
- Switches us it to identify redundant links
- The switches agree upon one switch becoming the primary point of reference (root bridge) for the entire network
- All redundant links to the root bridge are put in a blocked state to break any loops
- If a primary link goes down, then the redundant link will assume its place and start forwarding traffic.

STP ATTACKS

- The attacker can send spoofed root bridge messages (BPDUs) to a switch, advertising a better link to the root bridge
- The switch will redirect traffic from its normal path to the attacker instead
- The attacker can then sniff the incoming traffic
- The attacker can also choose to discard the traffic or redirect it back into the network



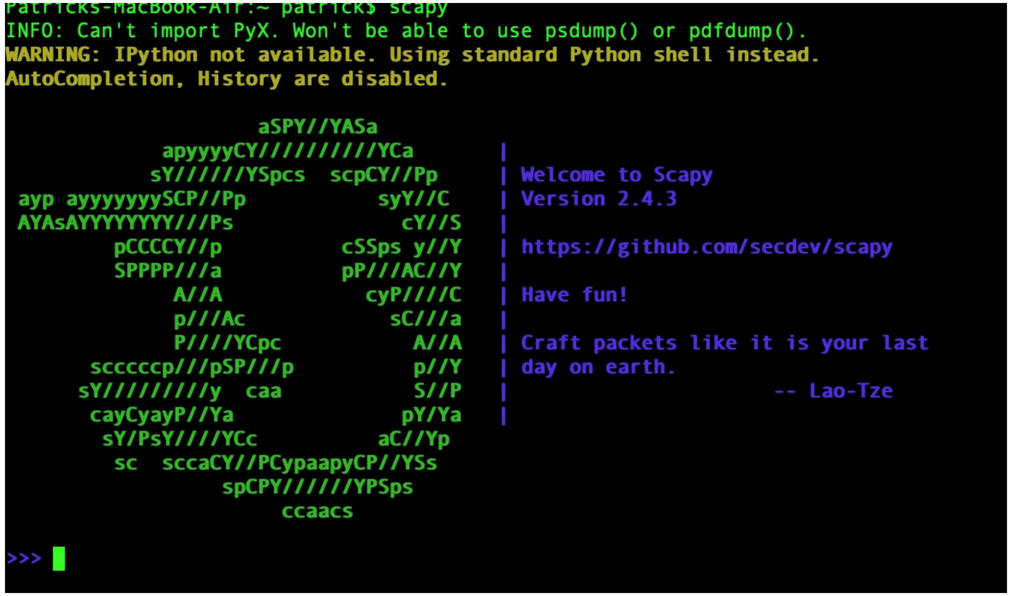
STP ATTACK TOOLS AND MITIGATION

Tools:

- Scapy
- Yersinia
- Various GitHub projects

Mitigation:

- Enable Root Guard on the switchports
- `spanning-tree guard root`



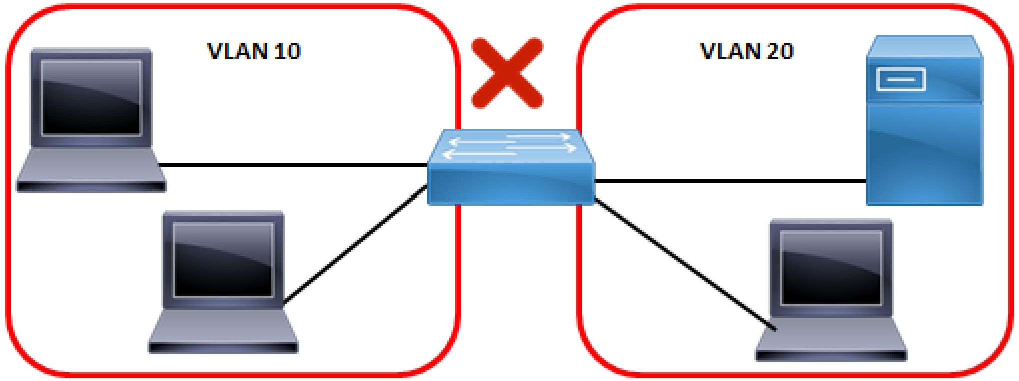
```
Patricks-MacBook-Air:~ patrick$ scapy
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: IPython not available. Using standard Python shell instead.
AutoCompletion, History are disabled.

                     aSPY//YASa
             apyyyyCY//////////YCa       |
            sY//////YSpcs  scpCY//Pp     | Welcome to Scapy
 ayp ayyyyyyySCP//Pp           syY//C    | Version 2.4.3
 AYAsAYYYYYYYY///Ps              cY//S   |
         pCCCCY//p          cSSps y//Y   | https://github.com/secdev/scapy
         SPPPP///a          pP///AC//Y   |
              A//A            cyP////C   | Have fun!
              p///Ac            sC///a   |
              P////YCpc           A//A   | Craft packets like it is your last
       scccccp///pSP///p          p//Y   | day on earth.
      sY/////////y  caa           S//P   |                      -- Lao-Tze
       cayCyayP//Ya              pY/Ya   |
        sY/PsY////YCc          aC//Yp
         sc  sccaCY//PCypaapyCP//YSs
                  spCPY//////YPSps
                       ccaacs

>>> 
```

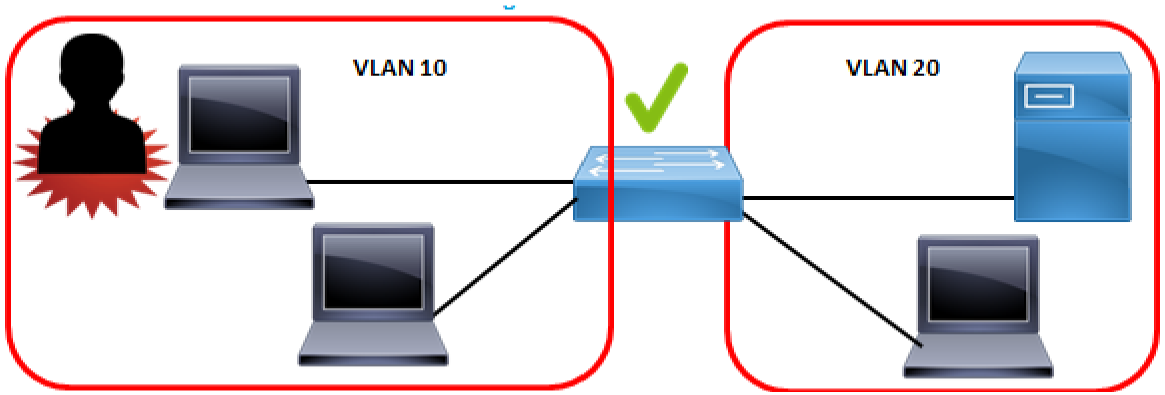
VIRTUAL LAN (VLAN)

- A logical grouping of switch ports
- Used to segregate end devices and their traffic based on various business criteria:
  - Location
  - Device type
  - Security level
- Each VLAN becomes its own broadcast domain
  - Traffic cannot not leave that VLAN unless routed by a router/Layer 3 switch
  - Devices can only communicate with other devices in the same VLAN
  - Generally, a switch access port (that an end device is plugged into) can only belong to one VLAN at any one time
- VLANs can extend across any number of switches on an Ethernet or Wi-Fi network

VLAN HOPPING

- The illegal movement of traffic from one VLAN to another
  - Traffic is not routed properly between VLANs
  - Traffic jumps over the VLAN "barricade" and ends up in another VLAN





COMMON VLAN HOPPING TECHNIQUES

- MAC flood a vulnerable switch
  - When this occurs, the switch defaults to operating as a hub
  - Repeats all frames out all ports
  - VLANs become meaningless
  - This "fail open" method ensures the network can continue to operate, but it is a security risk
- Configure an attacker's NIC as a "trunk port"
  - Encourage the switch to negotiate a trunk link
  - All VLAN traffic is then sent across that link to the attacker
- Double-tagging
  - A frame header is specially crafted with two VLAN tags, one embedded inside another
  - The outside tag must belong to the native (default) VLAN of the switch
  - The switch accepts the frame, discards the outer tag, reads the second tag, and then forwards the frame to that target VLAN

VLAN HOPPING COUNTERMEASURES

- Patch/update switch operating system
- Shut down unused ports and put them in an unused VLAN
- Explicitly configure ports for end devices as "access ports"
  - `switchport mode access`
- Disable Dynamic Trunking Protocol
  - An attacker will not be able to trick a switchport into establishing a trunk link with them
- Change the switch's native VLAN and ensure no port directly uses the native VLAN
  - This prevents a switch from accepting double-tagged frames

8.6 SNIFFING COUNTER-MEASURES

- Countermeasures
- Tools & Techniques

SNIFFING COUNTERMEASURES

- Use encrypted versions of protocols
- Require HTTP Strict Transport Security (HSTS) to prevent MITM downgrade attacks
- Prefer switches over hubs
- Configure port security on switches
- Consider using host-to-host (transport mode) VPNs
- Use strong encryption WPA3/2 for Wi-Fi
- Scan for NICs in promiscuous mode.

SNIFFING COUNTERMEASURES (CONT'D)

- Avoid public Wi-Fi spots
- Check DNS logs for Reverse DNS lookups
  - By default, sniffers will attempt to resolve IP addresses to names
- Ping suspected clients with the their correct IP but the wrong MAC address
  - If suspect accepts the packet, its interface is in promiscuous mode
  - A good indication of sniffing
- Use Nmap sniffer detection script:

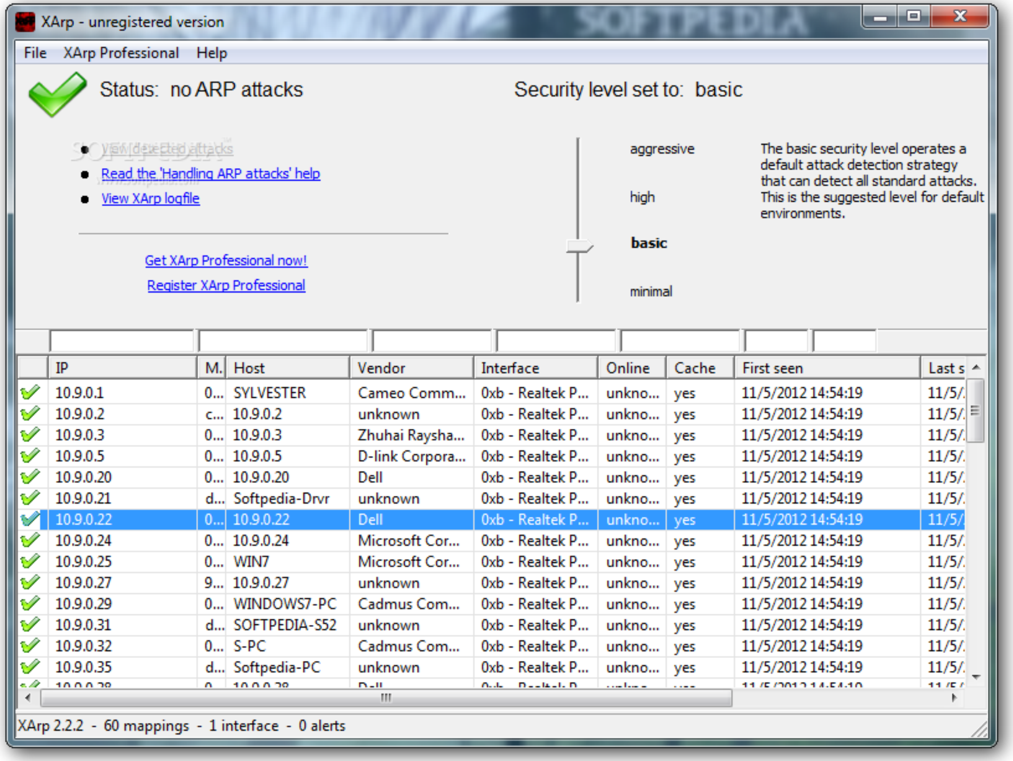
```
nmap --script=sniffer-detect <target>
```

PROMISCUOUS MODE DETECTION

- Transmit an ARP request with the fake broadcast address FF:FF:FF:FF:FF:FE
  - This will be blocked by all NIC's operating in normal mode
  - Will be allowed by NIC operating in promiscuous mode and thus it will respond to the message
- Promiscuous mode detection tools:
  - PromqryUI
  - Ifchk.

ARP SPOOFING DETECTION

- Use tools like Xarp to identify ARP attacks
- Hard code ARP-IP mappings
- Implement IDS
- Use host-to-host VPNs.


XArp - unregistered version
File
XArp Professional
Help
View detected attacks
Read the 'Handling ARP attacks' help
View XArp logfile
Get XArp Professional now!
Register XArp Professional
aggressive
high
basic
minimal

Status: no ARP attacks

Security level set to: basic

The basic security level operates a default attack detection strategy that can detect all standard attacks. This is the suggested level for default environments.

	IP	M.	Host	Vendor	Interface	Online	Cache	First seen	Last s
✓	10.9.0.1	0...	SYLVESTER	Cameo Comm...	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.2	c...	10.9.0.2	unknown	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.3	0...	10.9.0.3	Zhuhai Raysha...	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.5	0...	10.9.0.5	D-link Corpora...	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.20	0...	10.9.0.20	Dell	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.21	d...	Softpedia-Drvr	unknown	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.22	0...	10.9.0.22	Dell	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/
✓	10.9.0.24	0...	10.9.0.24	Microsoft Cor...	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.25	0...	WIN7	Microsoft Cor...	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.27	9...	10.9.0.27	unknown	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.29	0...	WINDOWS7-PC	Cadmus Com...	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.31	d...	SOFTPEDIA-S52	unknown	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.32	0...	S-PC	Cadmus Com...	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.35	d...	Softpedia-PC	unknown	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.
✓	10.9.0.38	0...	10.9.0.38	Dell	0xb - Realtek P...	unkno...	yes	11/5/2012 14:54:19	11/5/.

XArp 2.2.2 - 60 mappings - 1 interface - 0 alerts

SWITCHPORT SECURITY

- Limit MAC addresses that are allowed to connect to a switchport
  - Hard-code a maximum number of MACs per port
  - Hard-code the MAC-to-port mapping in the switch’s MAC table
    - Alternatively, allow “sticky MAC” learning – the switch enters the first MAC plugged into the port as the only permitted MAC
      - Better make sure you plug in an authorized device for the switch to learn!
- Set rules for switchport security violations
  - The port shuts down
  - The port is quarantined
  - The violation is logged.

ROGUE DEVICE DETECTION

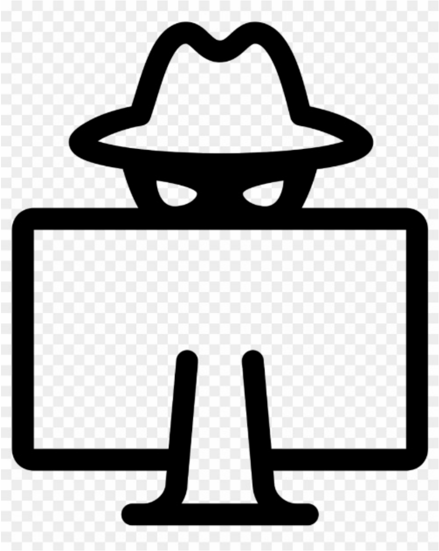
- DHCP Snooping
  - Feature that can be enabled on certain switches
  - Examines DHCP message exchanges passing through its ports
  - Detects and blocks DHCPOFFER frames from untrusted/unknown sources
- Dynamic ARP Inspection
  - Prevents malicious devices from poisoning their neighbors' ARP caches
  - Rejects invalid and malicious ARP packets
  - Relies on DHCP snooping
- Best option:
  - MAC address reporting from a source device like a router or a switch
  - You would need a management system or inventory process to capture these addresses
  - You then identify the rogue devices, and the switchports they were discovered on
- Next best option:
  - Periodic ARP scanning to list active MAC addresses
  - Check output for rogue devices.

8.7 SNIFFING REVIEW

- Review

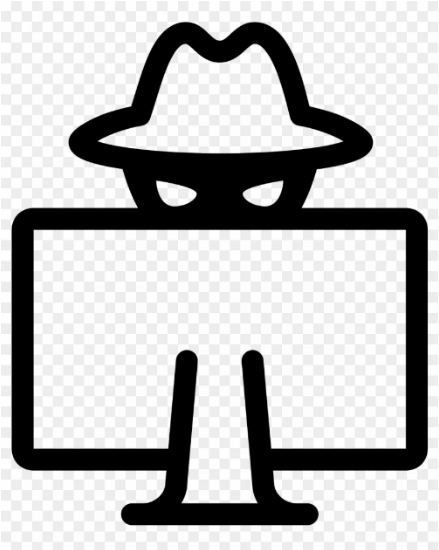
SNIFFING REVIEW

- Sniffing allows you to capture passwords, private messages, voice and video calls, files and other sensitive data from the network
- A good sniffer can capture any protocol from a variety of media types
  - Should also be able to use multiple filters, follow TCP sessions, recreate captured files from raw hex data, provide packet analysis, and save and load captures files
- Sniffing is successful when desired traffic passes a NIC in promiscuous mode
- ARP poisoning redirects local LAN segment traffic to the attacker's MAC address



SNIFFING REVIEW

- Sniffing allows you to capture passwords, private messages, voice and video calls, files and other sensitive data from the network
- A good sniffer can capture any protocol from a variety of media types
  - Should also be able to use multiple filters, follow TCP sessions, recreate captured files from raw hex data, provide packet analysis, and save and load captures files
- Sniffing is successful when desired traffic passes a NIC in promiscuous mode
- ARP poisoning redirects local LAN segment traffic to the attacker's MAC address



- MAC flooding forces a vulnerable switch to behave like a hub and flood all frames out all ports
  - Useful for VLAN hopping or when ARP poisoning is not desirable
- MAC spoofing changes the MAC address of your device's NIC
- Use DNS cache poisoning and other name resolution exploits to redirect targets when ARP poisoning isn't practical
  - Including when credential harvesting from another subnet
- Be careful when poisoning ARP and DNS caches as it could cause a denial-of-service for regular users.