



# **LAYER 2 ATTACKS & MITIGATION TECHNIQUES**

**SANOG8 – Karachi  
1<sup>st</sup> August 2006**

**Yusuf Bhaiji  
Cisco Systems**

# Agenda

- **Layer 2 Attack Landscape**
- **Attacks and Countermeasures**
  - Spanning Tree Attack**
  - VLAN “Hopping”**
  - MAC Attacks**
  - DHCP Attacks**
  - ARP Attack**
  - Spoofing Attacks**
- **Summary**

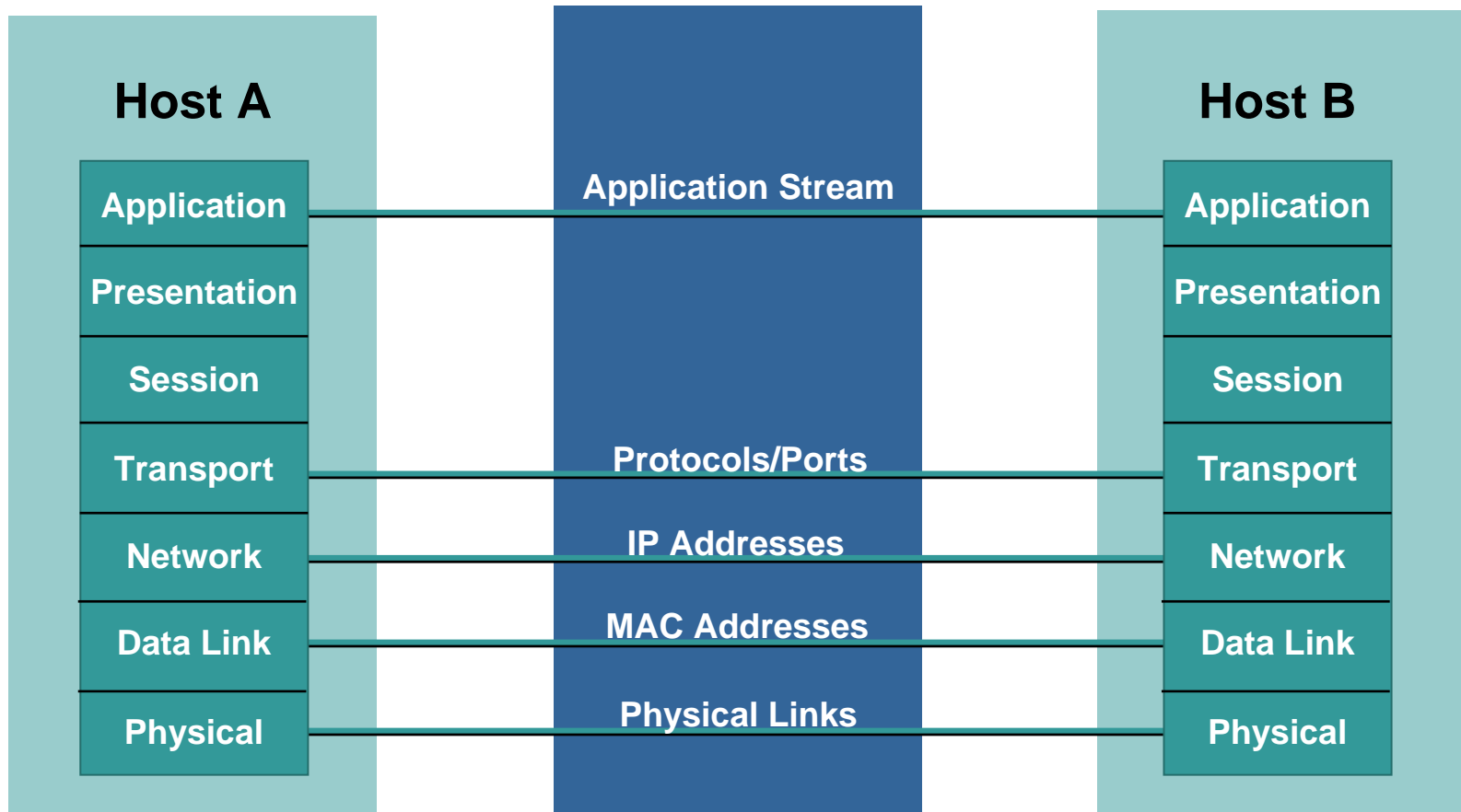
- **All attacks and mitigation techniques assume a switched Ethernet network running IP**
  - If it is a shared Ethernet access (WLAN, Hub, etc.) most of these attacks get much easier
  - If you are not using Ethernet as your L2 protocol, some of these attacks may not work, but chances are, you are vulnerable to different ones
- **New theoretical attacks can move to practical in days**
- **Ethernet switching attack resilience varies widely from vendor to vendor**
- **This is not a comprehensive talk on configuring Ethernet switches for security; the focus is mostly access L2 attacks and their mitigation**

# LAYER 2 ATTACK LANDSCAPE



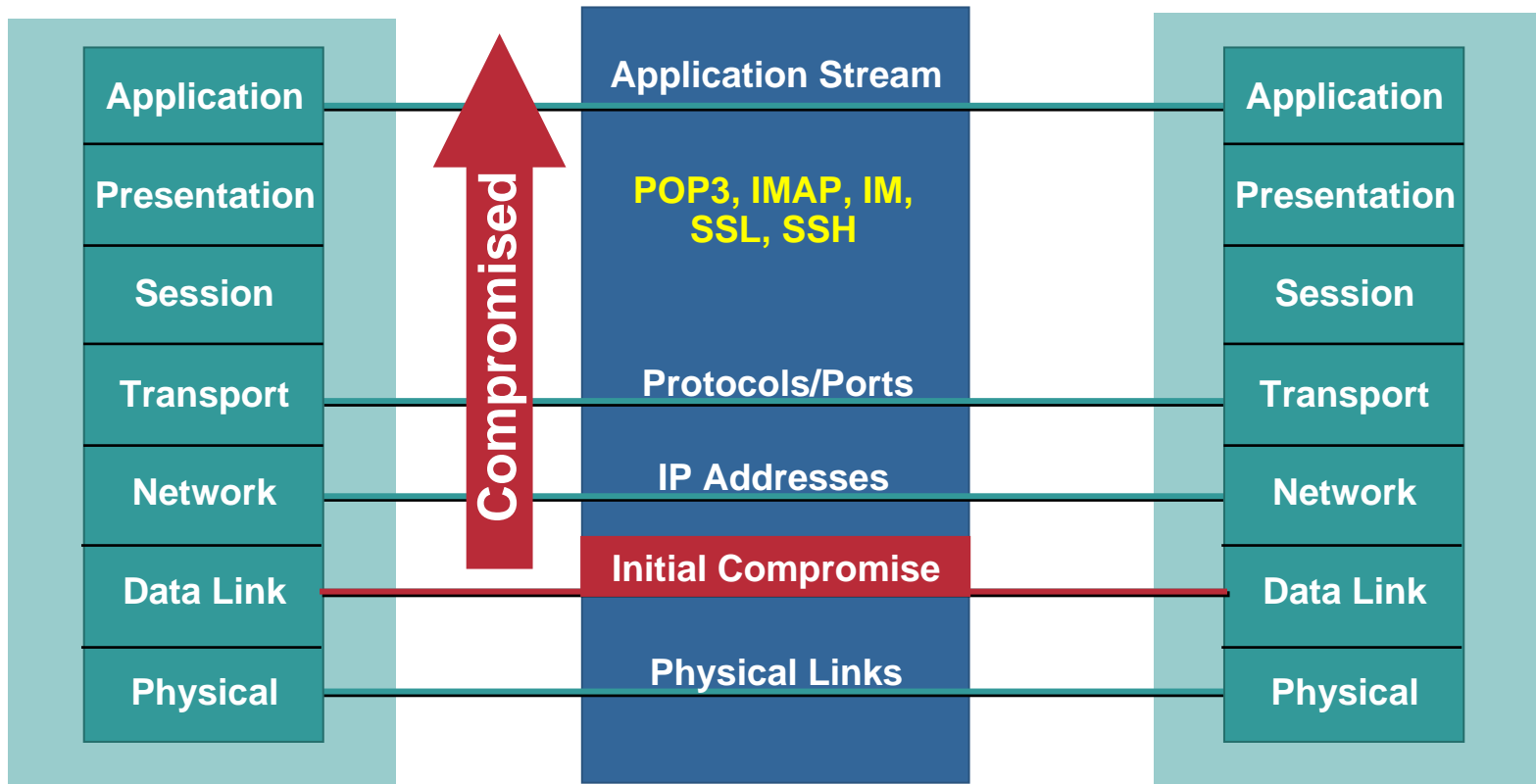
# Why Worry About Layer 2 Security?

## OSI Was Built to Allow Different Layers to Work Without the Knowledge of Each Other



# Lower Levels Affect Higher Levels

- Unfortunately this means if one layer is hacked, communications are compromised without the other layers being aware of the problem
- Security is only as strong as the weakest link
- When it comes to networking, layer 2 can be a VERY weak link



# FBI/CSI Risk Assessment\*

- **99% of all enterprises network ports are OPEN**
- **Usually any laptop can plug into the network and gain access to the network**
- **Of companies surveyed total loss was over 141 million**
- **An average of 11.4 million per incident**
- **Insider attack by disgruntled employees was listed as likely source by 59% of respondents**



\*CIS/FBI Computer Crime and Security Survey

[http://i.cmpnet.com/gocsi/db\\_area/pdfs/fbi/FBI2004.pdf](http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2004.pdf)

# Layer 2 Attacks

## There are lots of tools out there!!!

SecuriTeam.com™ (Archive) - Tools

http://www.securiteam.com/tools/archive.html

Location Search Bookmarks

Camino Info News Mac News Tabs Google

8. [new - Network Toolbox and Library](#)

9. [netUstad, Network Management Tool](#)

**Jun / 2004**

1. [Wasabi - Log Monitoring and Alert Tool](#)

2. [Hping3 \(alpha1\) - TCL Scripting Support](#)

3. [Portjammer \(SynAckFlood\) - Port Scanner](#)

4. [URCS - Unmanarc Remote Control Server](#)

5. [CifsPwScanner - CIFS/SMB Password Scanner](#)

6. [Weplan - WEP Testing Lab](#)

7. [vthrottle - SMTP Virus Throttling Engine](#)

8. [Garuda - Wireless Intrusion Detection System](#)

**May / 2004**

1. [cPanel Multiple Vulnerabilities Testing Script](#)

2. [Auditor Security Collection](#)

3. [Xpooft - Spoofed Packet Generator for Windows](#)

4. [AIRE - 802.11 Network Discovery for Windows](#)

5. [NetSQUID - IP Tables Snort Integration](#)

6. [Ettercap NG - Switched Network Sniffer](#)

7. [TCPTrack - TCP Tracker](#)

8. [Perfect Keylogger Password Cracker](#)

9. [RKDetect - Behaviour Based Rootkit Detector](#)

10. [Gwee \(Generic Web Exploitation Engine\)](#)

**Apr / 2004**

1. [OllyUni Plugin for OllyDbg](#)

2. [SSH Private Keys Cracker](#)

3. [Windows ARP Spoofer](#)

4. [Detect BIND Version without Banner Information](#)

5. [PLEM - Perl Loadable Exploiting Module](#)

6. [PSK Cracking Using IKE Aggressive Mode](#)

7. [FSTools - FileSystem Investigator](#)

8. [Hatchet - PF Firewall Log Parser](#)

9. [KnockD - Port Knocking Daemon](#)

Document: Done

ettercap

http://ettercap.sourceforge.net/

Location Search Bookmarks

Camino Info News Mac News Tabs Google

Home News Download Stuff History Forum FAQ Screenshots Authors Search...

# ETTERCAP<sup>NG</sup>

## NG-0.7.0 RELEASED !!

**Short Description:** Ettercap is a suite for man in the middle attacks on LAN. It features sniffing of live connections, content filtering on the fly and many other interesting tricks. It supports active and passive dissection of many protocols (even ciphered ones) and includes many feature for network and host analysis.

**Interface:** All this feature are integrated with a easy-to-use and pleasurable ncurses/gtk interfaces. (see [screenshots](#))

**Platform:**

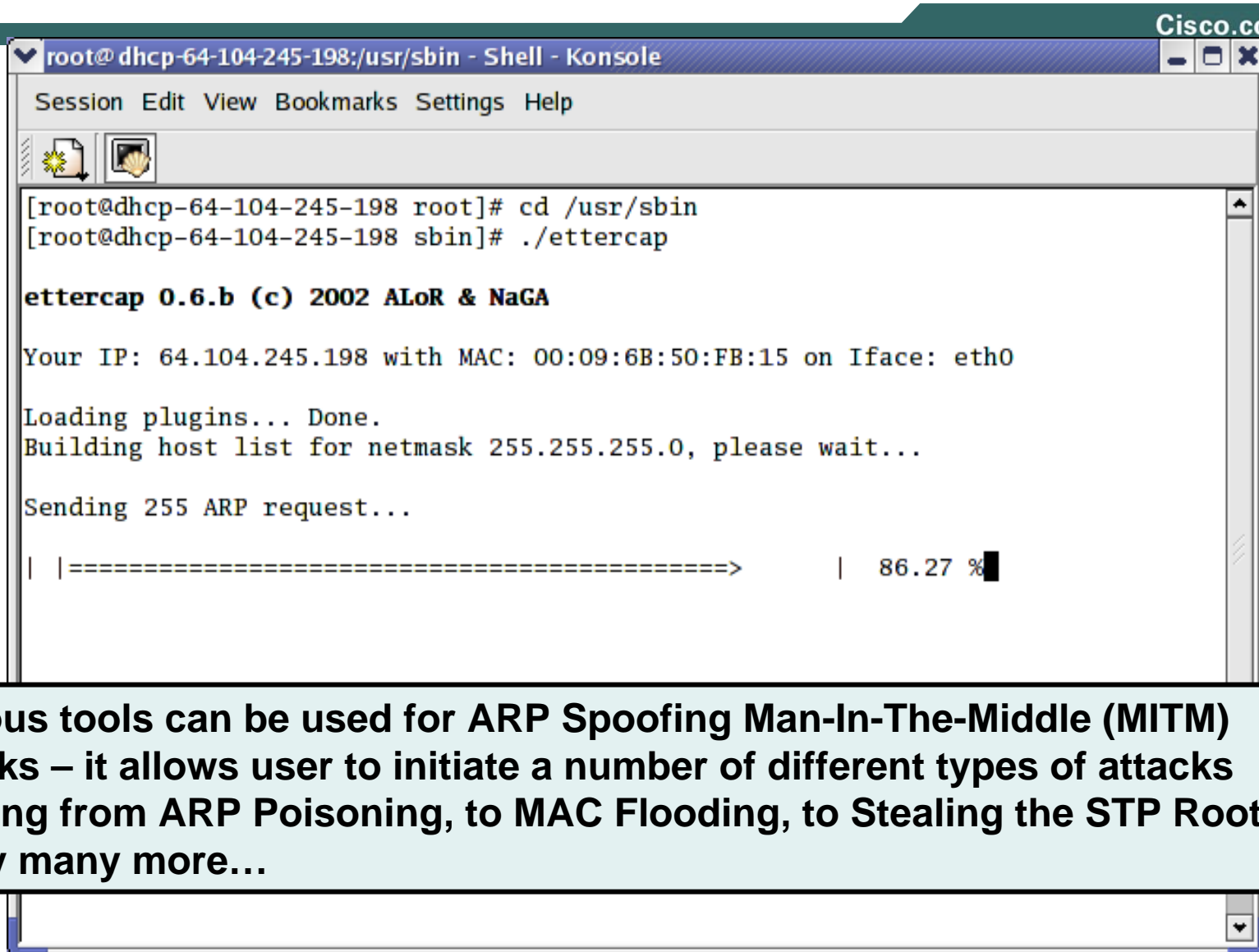
Linux 2.0.x	FreeBSD 4.x 5.x	Mac OS X (darwin 1.3 1.4 5.1 6.x 7.x)
Linux 2.2.x	OpenBSD 2.[789] 3.x	Windows 9x/NT/2000/XP
Linux 2.4.x	NetBSD 1.5	Solaris 2.x
Linux 2.6.x		

Document: Done



# Layer 2 Attacks

## ETTERCAP...



```
root@dhcp-64-104-245-198:/usr/sbin - Shell - Konsole
Session Edit View Bookmarks Settings Help
[root@dhcp-64-104-245-198 root]# cd /usr/sbin
[root@dhcp-64-104-245-198 sbin]# ./ettercap

ettercap 0.6.b (c) 2002 ALoR & NaGA

Your IP: 64.104.245.198 with MAC: 00:09:6B:50:FB:15 on Iface: eth0

Loading plugins... Done.
Building host list for netmask 255.255.255.0, please wait...

Sending 255 ARP request...

| |=====> | 86.27 %
```

**Various tools can be used for ARP Spoofing Man-In-The-Middle (MITM) attacks – it allows user to initiate a number of different types of attacks ranging from ARP Poisoning, to MAC Flooding, to Stealing the STP Root, and many many more...**

# Layer 2 Attacks

Let the fun begin...

```
root@dhcp-64-104-245-198:/usr/sbin - Shell - Konsole
Session Edit View Bookmarks Settings Help
ettercap 0.6.b
-----
30 hosts in this LAN (64.104.245.198 : 255.255.255.0)
-----
1) 64.104.245.198      1) 64.104.245.198
2) 64.104.245.1       2) 64.104.245.1
3) 64.104.245.2       3) 64.104.245.2
4) 64.104.245.3       4) 64.104.245.3
5) 64.104.245.4       5) 64.104.245.4
6) 64.104.245.5       6) 64.104.245.5
7) 64.104.245.11      7) 64.104.245.11
8) 64.104.245.12      8) 64.104.245.12
9) 64.104.245.13      9) 64.104.245.13
10) 64.104.245.14     10) 64.104.245.14
11) 64.104.245.15     11) 64.104.245.15
12) 64.104.245.16     12) 64.104.245.16
13) 64.104.245.17     13) 64.104.245.17
14) 64.104.245.18     14) 64.104.245.18
-----
Your IP: 64.104.245.198 MAC: 00:09:6B:50:FB:15 Iface: eth0 Link: SWITCH
Host: dhcp-64-104-245-198.cisco.com (64.104.245.198) : 00:09:6B:50:FB:15
Host: dhcp-64-104-245-198.cisco.com (64.104.245.198) : 00:09:6B:50:FB:15
```

# ATTACKS AND COUNTERMEASURES: SPANNING TREE ATTACK

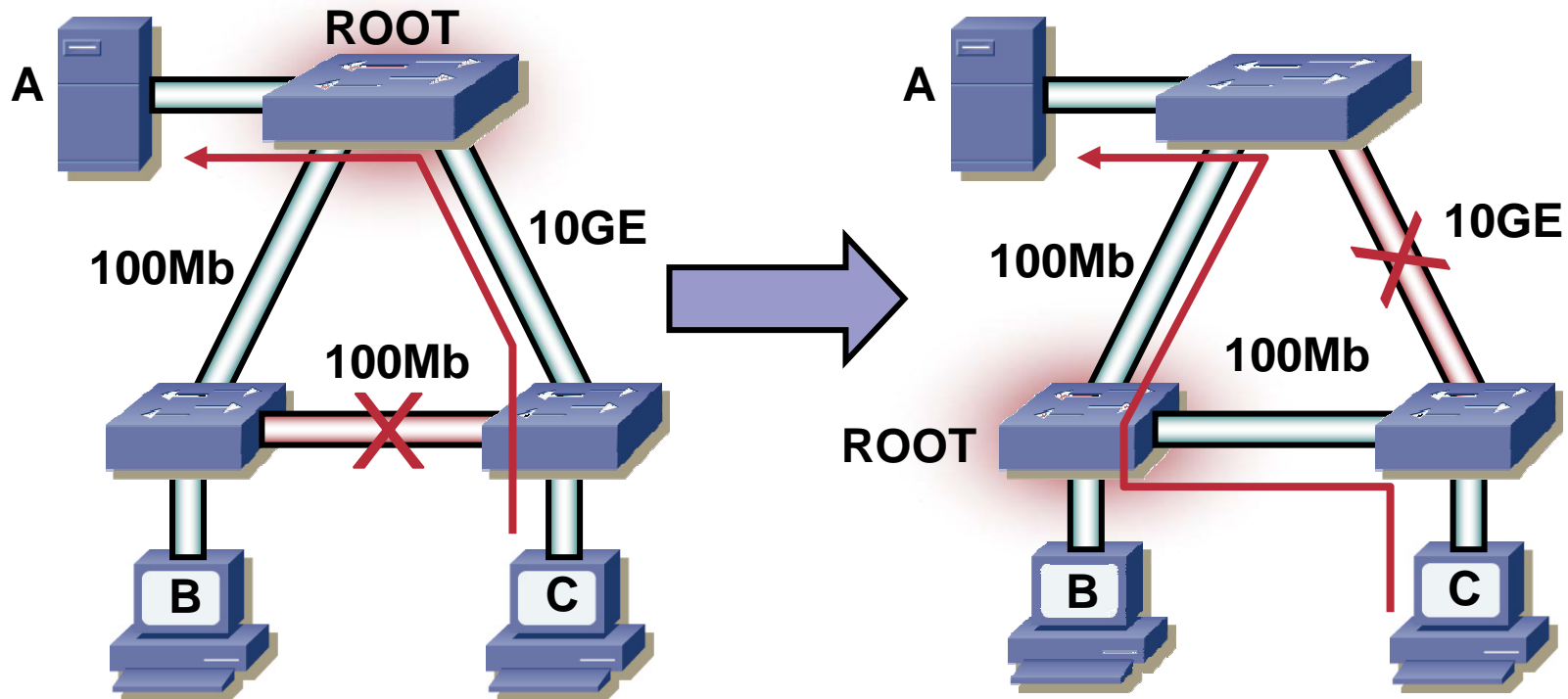


# Spanning Tree Attacks

Re-directing traffic while causing disruption...

Cisco.com

Injecting BPDU packets into the STP domain can cause the entire domain to reconfigure (30-45 second outage) and end up using a less than optimal path for data forwarding – form of Denial of Service attack...



# Spanning Tree Attacks

## Using Plug-ins...

```
root@dhcp-64-104-245-198:/usr/sbin - Shell - Konsole
Session Edit View Bookmarks Settings Help
ettercap 0.6.b
Help Window
[qQ][F10] - quit
[return] - select the IP
3 [space] - deselect the IPs
0) [tab] - switch between source and dest
[aA] - ARP poisoning based sniffing
    . for sniffing on switched LAN
    . for man-in-the-middle technique
[sS] - IP based sniffing
[mM] - MAC based sniffing
[jJ] - Only poisoning - no sniffing
[dD] - delete an entry from the list
[xX] - Packet Forge
[pP] - run a plugin
[iI] - OS fingerprint
[oO] - passive host identification
[cC] - check for other poisoner...
[rR] - refresh the list
[kK] - save host list to a file
[hH] - this help screen
Your IP: 6
Host: dhcp-
Host: dhcp-
ink: SWITCH
6B: 50:FB:15
6B: 50:FB:15
```

# Spanning Tree Attacks

## Taking over STP ROOT...

```
root@dhcp-64-104-245-198:/usr/sbin - Shell - Konsole
Session Edit View Bookmarks Settings Help
ettercap 0.6.b
-----
30 hosts in this LAN (64.104.245.198 : 255.255.255.0)
1) 64.104.245.198      1) 64.104.245.198
-----
24) hunter      1.0 E -- Search promisc NICs
25) imp         1.2 E -- Retrieves some Windows names
26) lamia       1.1 E -- Become root of a switches spanning tree (STP)
27) leech       2.2 E -- Isolate a host from the LAN
28) ooze        1.4 E -- Ping a host
29) phantom     1.6 E -- Sniff/Spoof DNS requests
30) shadow      1.8 E -- A very simple SYN/TCP port scanner
31) spectre     1.3 E -- Flood the LAN with random MAC addresses
32) triton      2.1 E -- Try to discover the LAN's gateway
-----
14) 64.104.245.18    14) 64.104.245.18
-----
Your IP: 64.104.245.198 MAC: 00:09:6B:50:FB:15 Iface: eth0 Link: SWITCH
Host: dhcp-64-104-245-198.cisco.com (64.104.245.198) : 00:09:6B:50:FB:15
Host: dhcp-64-104-245-198.cisco.com (64.104.245.198) : 00:09:6B:50:FB:15
```

# Spanning Tree Attacks

## Taking over STP ROOT...

```
root@dhcp-64-104-245-198:/usr/sbin - Shell - Konsole
Session Edit View Bookmarks Settings Help
ettercap 0.6.b
-----
30 hosts in this LAN (64.104.245.198 : 255.255.255.0)
1) 64.104.245.198 1) 64.104.245.198
Starting lamia plugin...
Priority? [0]:
14) 64.104.245.18 14) 64.104.245.18
Your IP: 64.104.245.198 MAC: 00:09:6B:50:FB:15 Iface: eth0 Link: SWITCH
Host: dhcp-64-104-245-198.cisco.com (64.104.245.198) : 00:09:6B:50:FB:15
Host: dhcp-64-104-245-198.cisco.com (64.104.245.198) : 00:09:6B:50:FB:15
```

# Spanning Tree Attacks

## Bingo - you have now become STP Root...

```
root@localhost:/usr/sbin - Shell - Konsole
Session Edit View Bookmarks Settings Help

----- ettercap 0.6.b -----
SOURCE: 10.66.240.41 <----- Filter: OFF
DEST  : 10.66.240.42 <----- doppleganger - illithid (ARP Based) - ettercap
Active Dissector: ON

----- 4 hosts in this LAN (10.66.240.44 : 255.255.255.248) -----
27) 10.66.240.42:51481 <--> 64.104.14.184:53 | UDP | domain
Starting lamia plugin...
Priority? [0]:
If it doesn't work...
...try to set your MAC address to a lower value
Sending BPDUs with priority=0...(press return to stop)
40) 10.66.240.42:51507 <--> 64.104.200.248:53 | UDP | domain

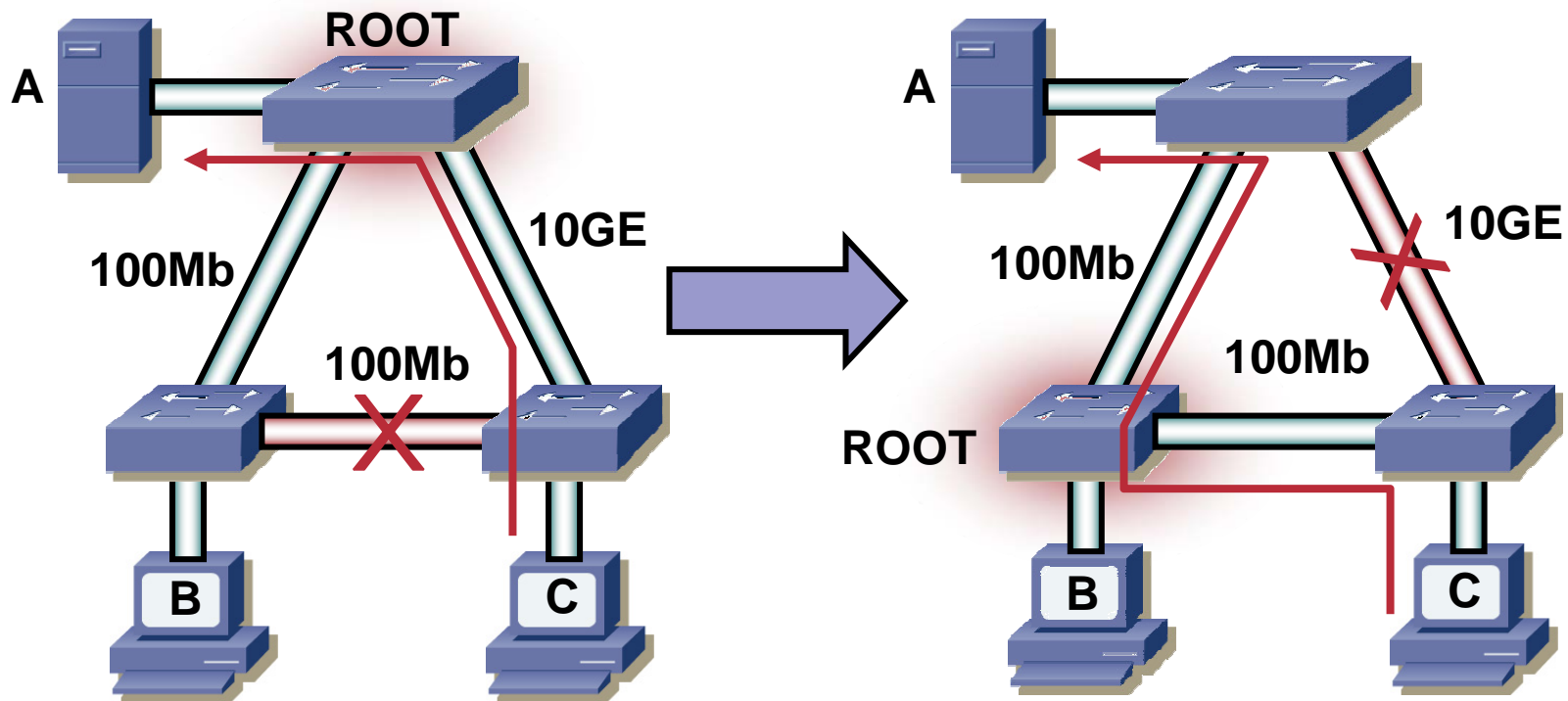
----- Your IP: 10.66.240.44 MAC: 00:09:6B:50:FB:15 Iface: eth0 Link: SWITCH -----
```



# Spanning Tree Attacks

Now traffic is re-directed to you...

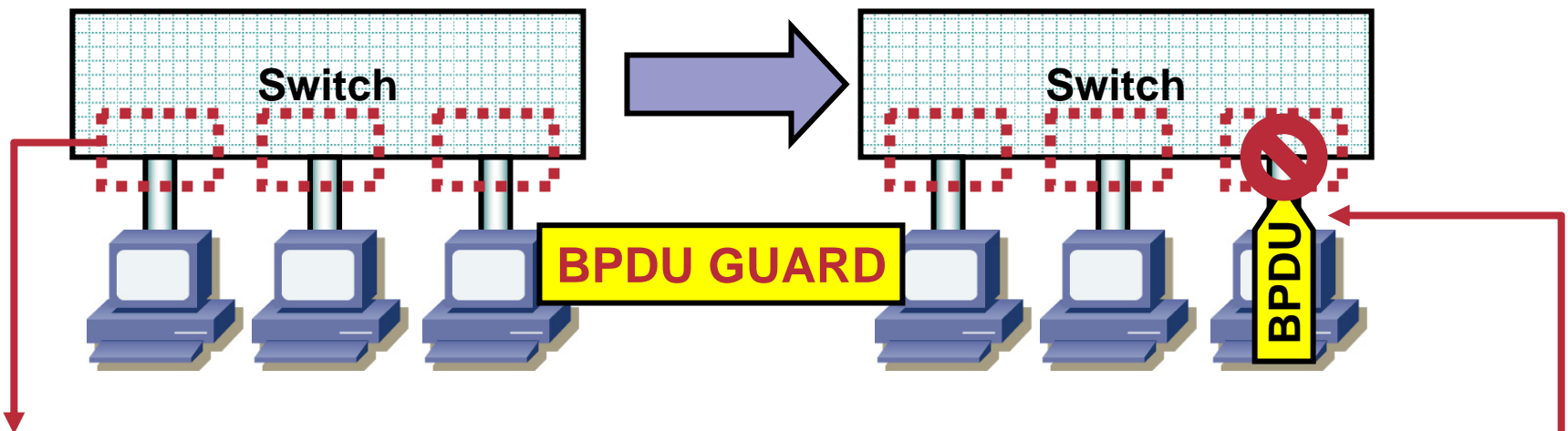
This attack has resulted in the STP root being moved - now traffic will be diverted via our switch...



# Mitigating Spanning Tree Attacks

## BPDU Guard...

BPDU Guard is one mechanism to avoid an attacker injecting BPDU packets and becoming the STP Root... applied globally on the switch...



```
CatOS> (enable) set spantree portfast bpdu-guard enable  
IOS(config)# spanning-tree portfast bpduguard
```

If a BPDU is detected on a BPDU Guard port, the switchport is shutdown

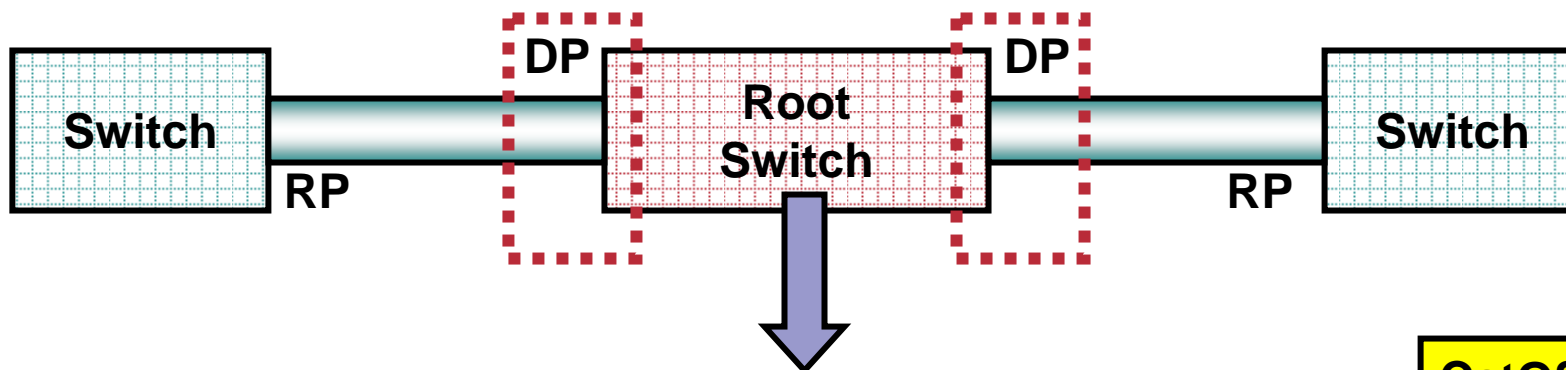
# Mitigating Spanning Tree Attacks

## ROOT Guard...

Cisco.com

Root Guard is another mechanism to avoid an attacker becoming the STP Root... applied globally on the switch... forces local ports to become “Designated” Ports

DP = Designated Port – Points away from the Root  
RP = Root Port – Points towards to Root



```
Switch> (enable) set spantree guard root 3/1
```

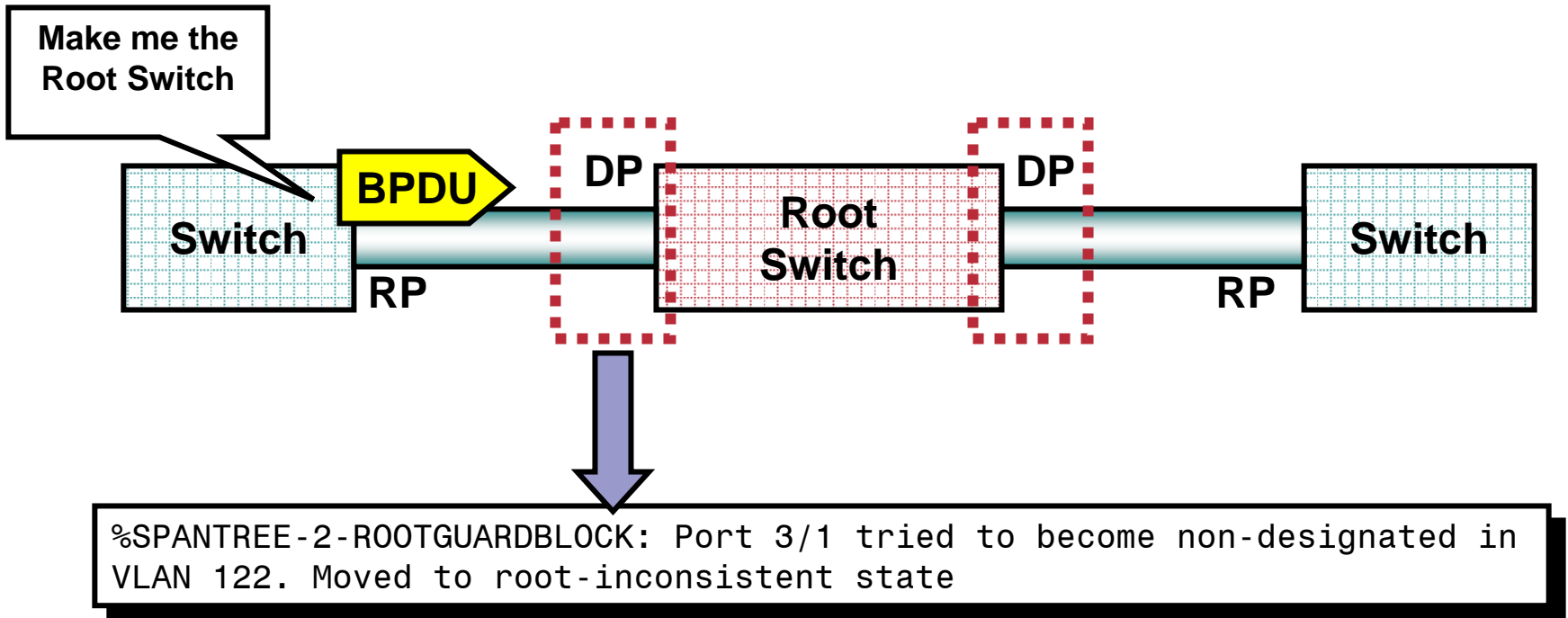
CatOS

```
Switch(config)# spanning-tree guard root (or rootguard)
```

IOS

# Mitigating Spanning Tree Attacks

## ROOT Guard...



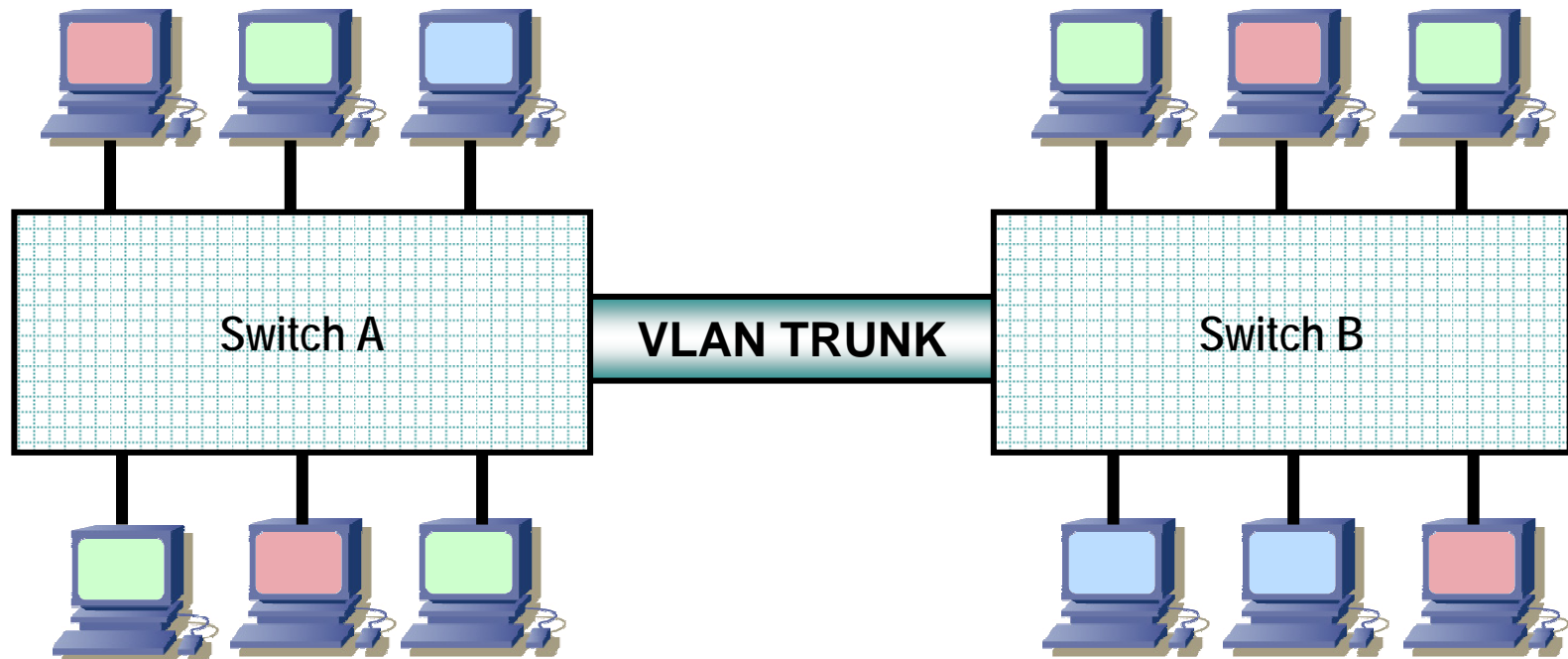
If a root guarded port receives a superior BPDU, the port is moved to a “Root inconsistent STP State” (similar to STP Listening state) and no traffic will be forwarded across that port – thus root switch position maintained

# ATTACKS AND COUNTERMEASURES: VLAN HOPPING ATTACKS



# VLAN Trunk Refresher

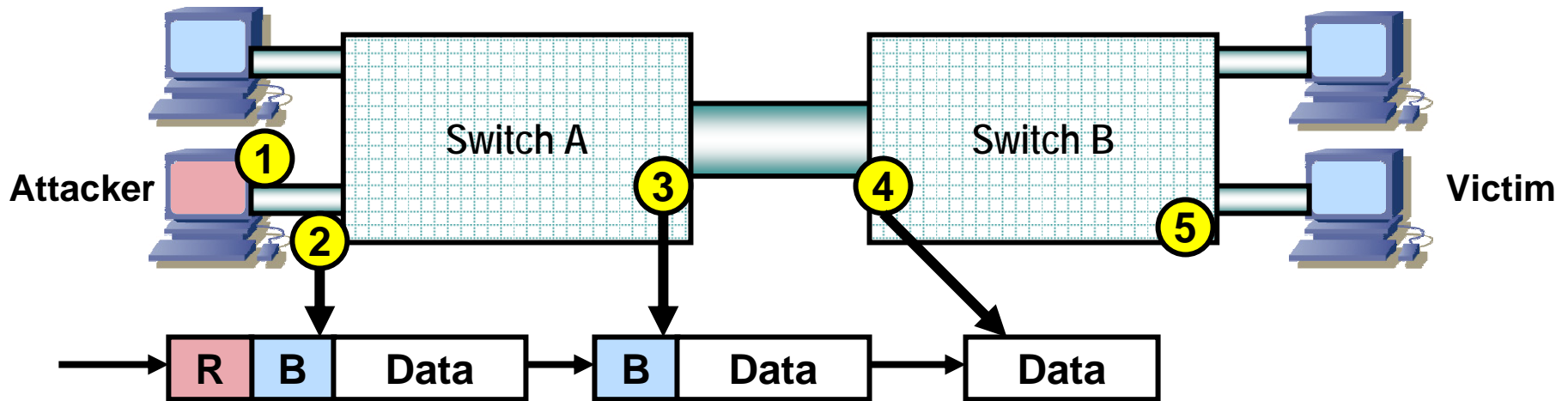
**VLAN Trunks are used to forward traffic from multiple VLAN's across a single physical link – Two encapsulation options are available for VLAN Trunking – Cisco ISL and IEEE 802.1Q**



**In this network, the VLAN Trunk is configured to carry VLAN traffic for the RED, BLUE and GREEN VLAN's**

# VLAN Hopping Attack

Idea behind the VLAN Hopping attack is to negotiate a VLAN trunk between the switch and host and then send a double tagged packet to the switch



- ① Host negotiates Trunk
- ② Host sends double tagged frame
- ③ Switch strips off first frame and forwards to next hop

- ④ Switch strips off VLAN tag
- ⑤ Switch forwards frame in BLUE VLAN to host

**Result: VLAN Hopped!!!**

# Ethereal Capture

No.	Time	Source	Destination	Protocol	Info
1	0.000000	1.2.3.9	1.2.3.4	ICMP	Echo (ping) request

Frame 1 (64 on wire, 64 captured)

Arrival Time: Jul 27, 2002 19:40:39.934687000  
Time delta from previous packet: 0.000000000 seconds  
Time relative to first packet: 0.000000000 seconds  
Frame Number: 1  
Packet Length: 64 bytes  
Capture Length: 64 bytes

Ethernet II  
Destination: 00:03:47:b9:6f:ae (Intel\_b9:6f:ae)  
Source: 00:03:47:20:0b:26 (Intel\_20:0b:26)

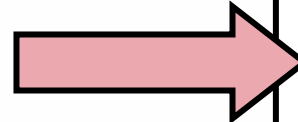
Type: 802.1Q Virtual LAN (0x8100)

802.1q Virtual LAN  
000. .... = Priority: 0  
...0 .... = CFI: 0  
... 0000 0000 0001 = ID: 1  
Type: 802.1Q Virtual LAN (0x8100)

802.1q Virtual LAN  
111. .... = Priority: 7  
...0 .... = CFI: 0  
... 0000 0000 0010 = ID: 2  
Type: IP (0x0800)  
Trailer: 0000000000000000000000000081C1A10F

Internet Protocol, Src Addr: 1.2.3.9 (1.2.3.9), Dst Addr: 1.2.3.4 (1.2.3.4)  
Version: 4  
Header length: 20 bytes  
Differentiated Services Field: 0x00 (DSCP 0x00; Default; ECN: 0x00)  
Total Length: 28  
Identification: 0x00f2  
Flags: 0x00  
Fragment offset: 0  
Time to live: 64  
Protocol: ICMP (0x01)  
Header checksum: 0x71df (correct)  
Source: 1.2.3.9 (1.2.3.9)  
Destination: 1.2.3.4 (1.2.3.4)

Internet Control Message Protocol



**First tag shown here is the outside VLAN or the attackers VLAN...**



**Second tag shown is the inside VLAN or the target VLAN that the attacker wants to hop to...**



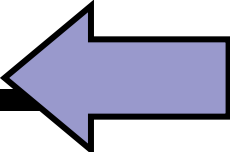
# Mitigating VLAN Hopping

**Easiest way to stop this from happening is to DISABLE trunking on unnecessary ports**

```
CatOS> (enable) set trunk <mod/port> off  
Or  
IOS(config-if)# switchport mode access
```

**CatOS also has a neat feature – a macro command that sets in place a suitable set of parameters for a host port – this command is shown below...**

```
C6500> (enable) set port host 6/22  
Port(s) 6/22 channel mode set to off.  
  
Warning: Connecting Layer 2 devices to a fast start port can cause  
temporary spanning tree loops. Use with caution.  
  
Spantree port 6/22 fast start enabled.  
Dot1q tunnel feature disabled on port(s) 6/22.  
Port(s) 6/22 trunk mode set to off.
```



# Security Best Practices for VLANs and Trunking

- **Always use a dedicated VLAN ID for all trunk ports**
- **Disable unused ports and put them in an unused VLAN**
- **Be paranoid: Do not use VLAN 1 for anything**
- **Disable auto-trunking on user facing ports (DTP off)**
- **Explicitly configure trunking on infrastructure ports**
- **Use all tagged mode for the Native VLAN on trunks**

# ATTACKS AND COUNTERMEASURES: MAC ATTACKS



# MAC Address/CAM Table Review

48 Bit Hexadecimal Number Creates Unique Layer Two Address

**1234.5678.9ABC**

First 24 bits = Manufacture Code  
Assigned by IEEE

**0000.0cXX.XXXX**

Second 24 bits = Specific Interface,  
Assigned by Manufacture

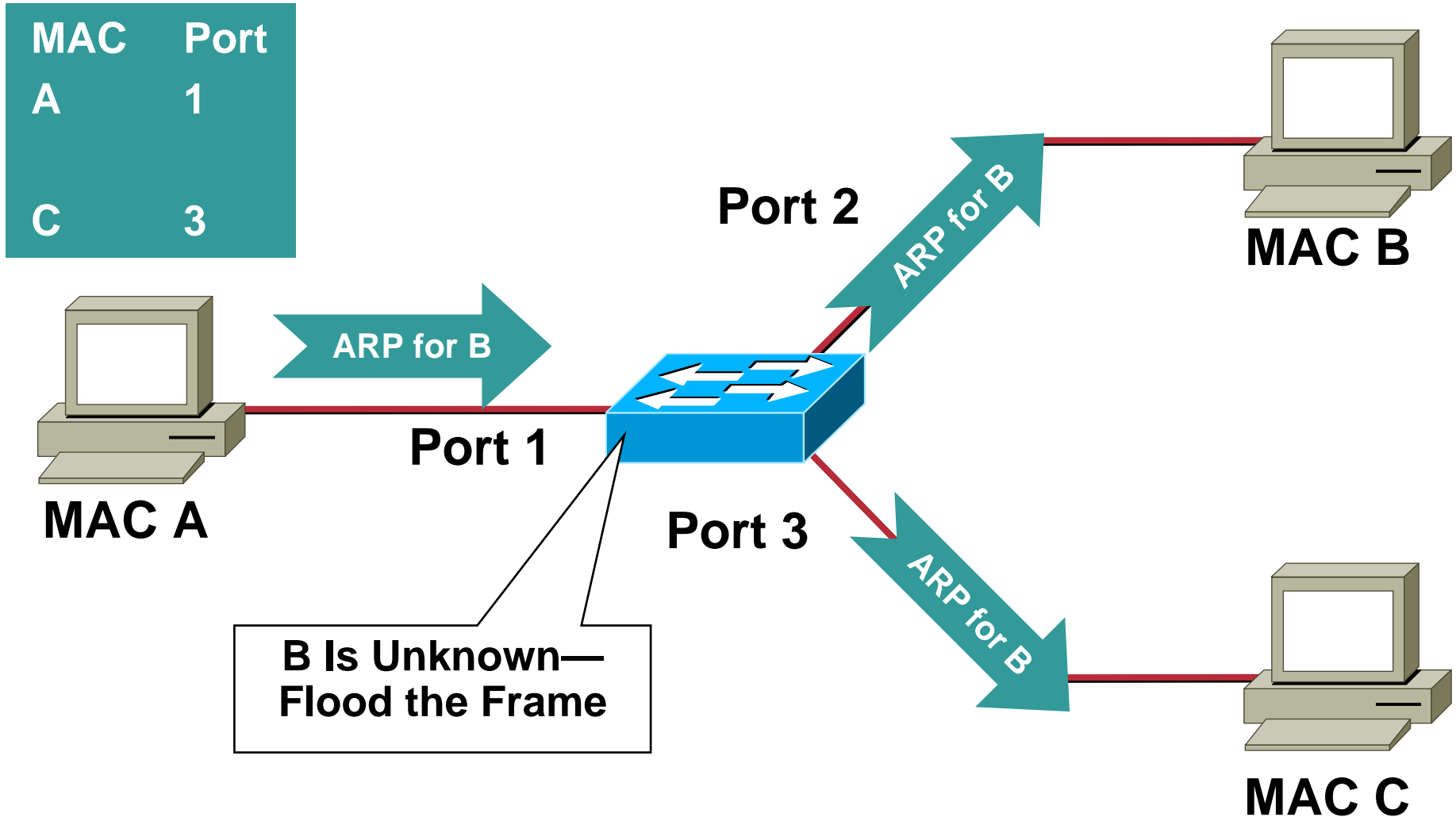
**0000.0cXX.XXXX**

All F's = Broadcast

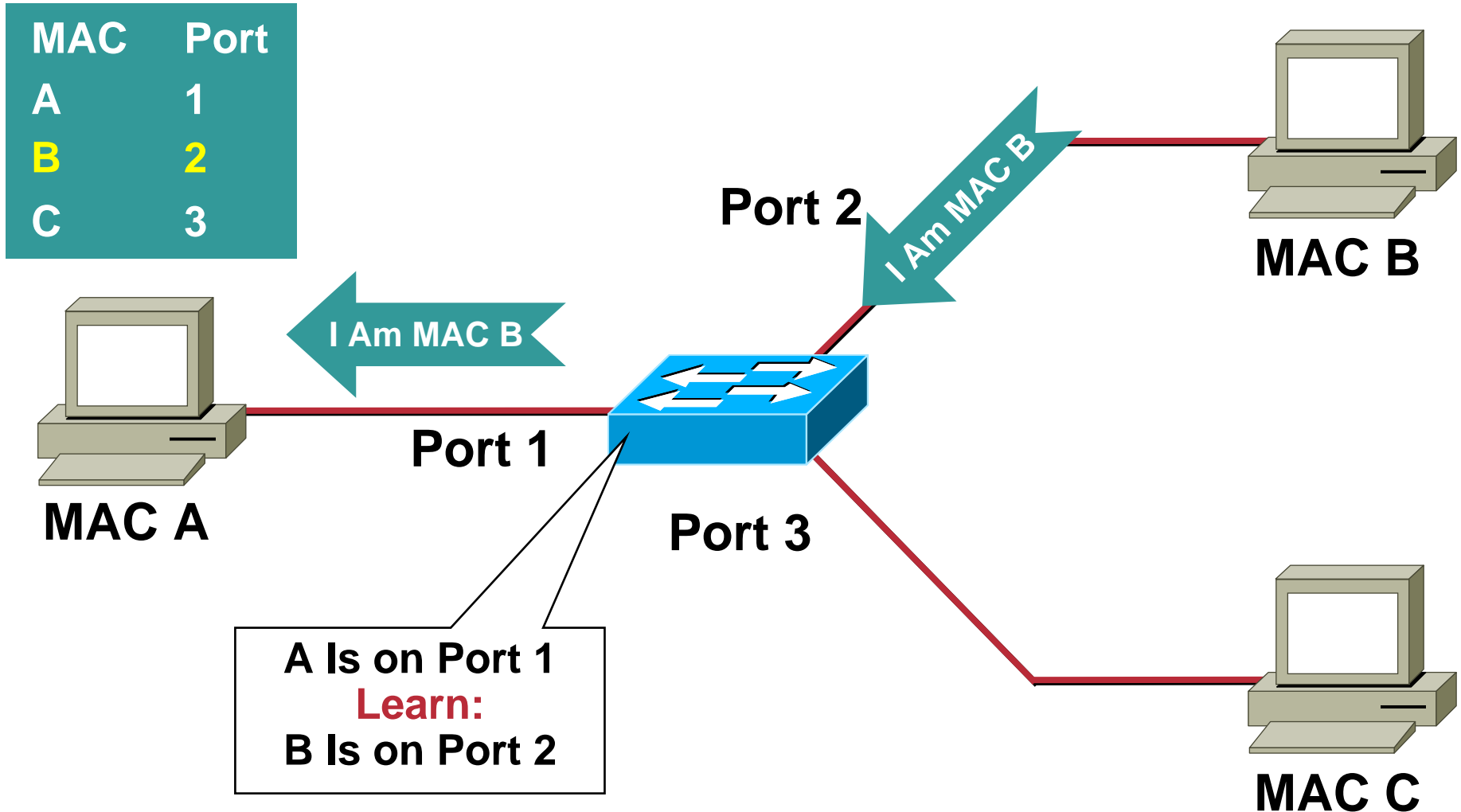
**FFFF.FFFF.FFFF**

- CAM table stands for Content Addressable Memory
- The CAM table stores information such as MAC addresses available on physical ports with their associated VLAN parameters
- CAM tables have a fixed size

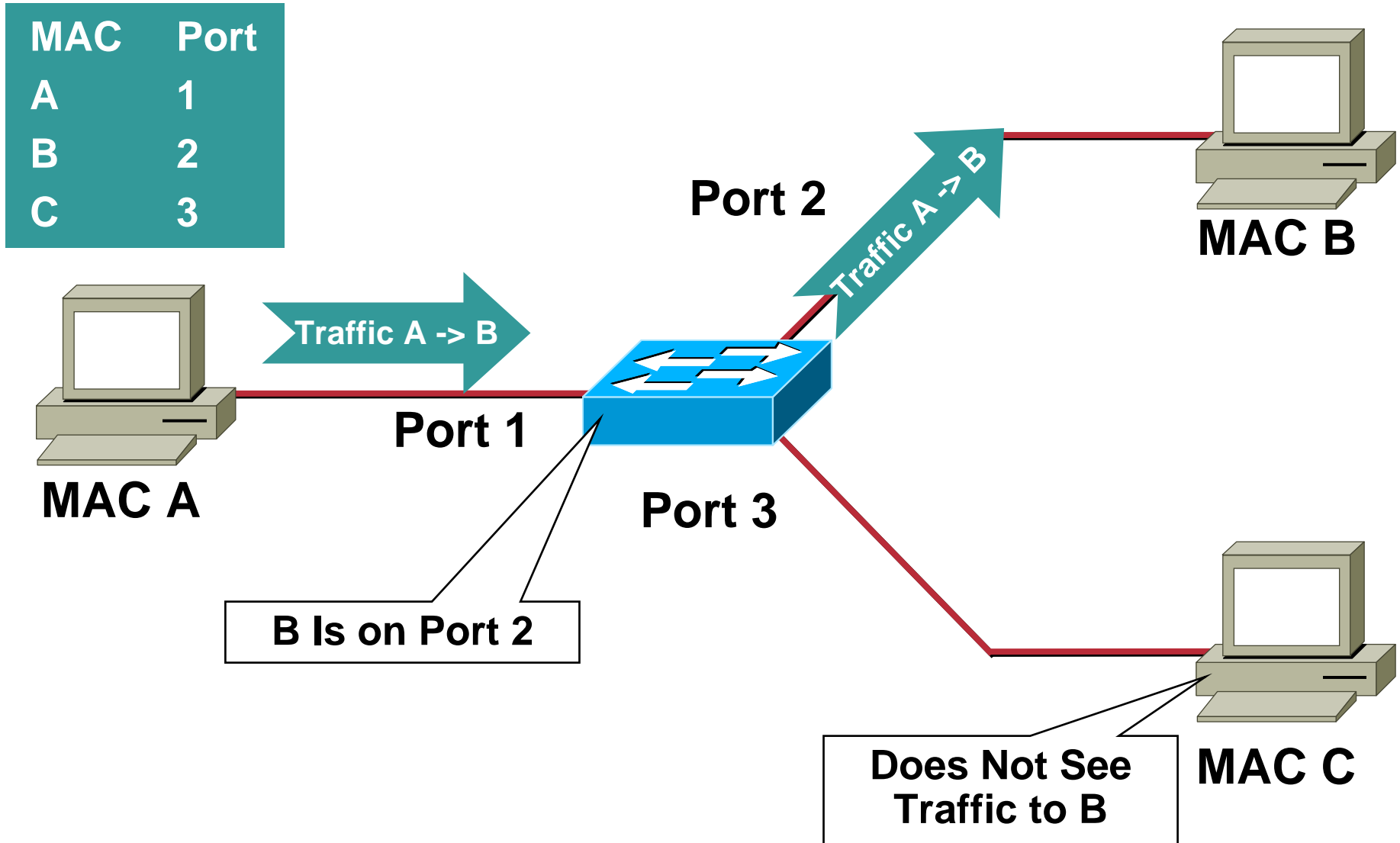
# Normal CAM Behavior 1/3



# Normal CAM Behavior 2/3



# Normal CAM Behavior 3/3

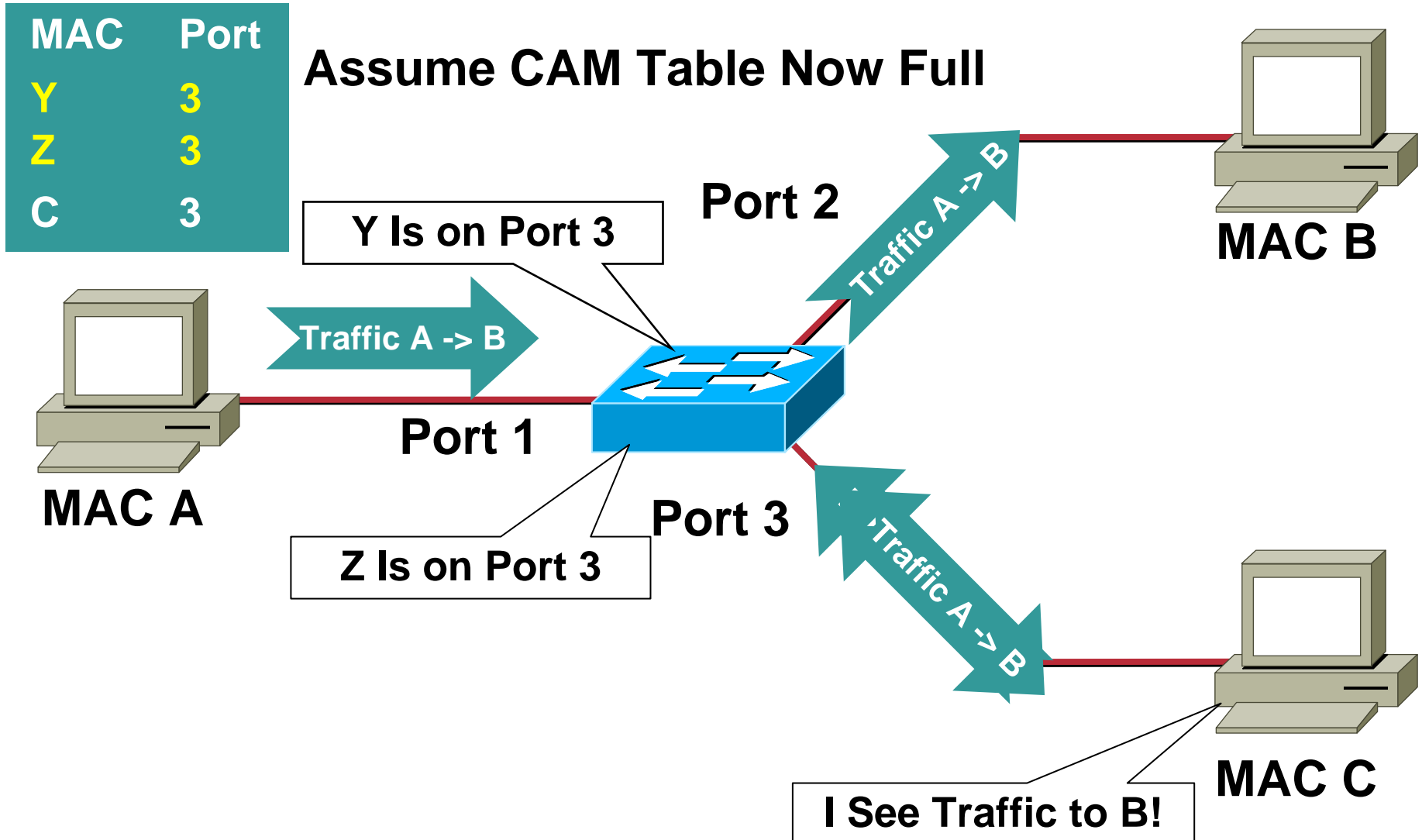


# CAM Overflow 1/3

- **macof tool since 1999**
  - About 100 lines of perl
  - Included in “dsniff”
- **Attack successful by exploiting the size limit on CAM tables**



# CAM Overflow 2/3



# MAC/CAM Attacks

## Flooding the MAC Table?

Cisco.com

```
root@dhcp-64-104-245-198:/usr/sbin - Shell - Konsole
Session Edit View Bookmarks Settings Help

----- ettercap 0.6.b -----

----- 30 hosts in this LAN (64.104.245.198 : 255.255.255.0) -----
18) 64.104.245.28 18) 64.104.245.28

24) hunter      1.0 E -- Search promisc NICs
25) imp         1.2 E -- Retrieves some Windows names
26) lamia       1.1 E -- Become root of a switches spanning tree (STP)
27) leech       2.2 E -- Isolate a host from the LAN
28) ooze        1.4 E -- Ping a host
29) phantom     1.6 E -- Sniff/Spoof DNS requests
30) shadow      1.8 E -- A very simple SYN/TCP port scanner
31) spectre     1.3 E -- Flood the LAN with random MAC addresses
32) triton      2.1 E -- Try to discover the LAN's gateway

----- Your IP: 64.104.245.198 MAC: 00:09:6B:50:FB:15 Iface: eth0 Link: SWITCH -----
Host: printer-oz-per-bw02.cisco.com (64.104.245.36) : 00:01:E6:A3:B0:14
```

# MAC/CAM Attacks

## Flooding the MAC Table?

```
root@dhcp-64-104-245-198:/usr/sbin - Shell - Konsole
Session Edit View Bookmarks Settings Help
ettercap 0.6.b
-----
30 hosts in this LAN (64.104.245.198 : 255.255.255.0)
18) 64.104.245.28 18) 64.104.245.28
Starting spectre plugin...
Are you sure you want to Flood the LAN with random MAC addresses ? (yes/n
o)
-----
Your IP: 64.104.245.198 MAC: 00:09:6B:50:FB:15 Iface: eth0 Link: SWITCH
Host: printer-oz-per-bw02.cisco.com (64.104.245.36) : 00:01:E6:A3:B0:14
```

# MAC/CAM Attacks

## Flooding the MAC Table?

```
root@dhcp-64-104-245-198:/usr/sbin - Shell - Konsole
Session Edit View Bookmarks Settings Help
ettercap 0.6.b
-----
30 hosts in this LAN (64.104.245.198 : 255.255.255.0)
18) 64.104.245.28 18) 64.104.245.28
Starting spectre plugin...
Are you sure you want to Flood the LAN with random MAC addresses ? (yes/n
o) yes
Flooding the lan... (press return to exit)
-----
Your IP: 64.104.245.198 MAC: 00:09:6B:50:FB:15 Iface: eth0 Link: SWITCH
Host: printer-oz-per-bw02.cisco.com (64.104.245.36) : 00:01:E6:A3:B0:14
```

# MAC/CAM Attacks

## MACOF Attack Tool?

MACOF is one of a number of tools available with “DSNIFF”

Dynamically generates MAC addresses to fill the Switch CAM table...

Three main development platforms  
Red Hat Linux, Solaris and Open BSD  
(Also on Win2K/XP, FreeBSD, Debian, AIX, and HPUX)



<http://www.monkey.org/~dugsong/dsniff>

```
[root@macattack]# macof -i eth0
```

```
36:a1:48:63:81:70 15:26:8d:4d:28:f8 0.0.0.0.26413 > 0.0.0.0.49492: S 1094191437:1094191437(0) win 512
16:e8:8:0:4d:9c da:4d:bc:7c:ef:be 0.0.0.0.61376 > 0.0.0.0.47523: S 446486755:446486755(0) win 512
18:2a:de:56:38:71 33:af:9b:5:a6:97 0.0.0.0.20086 > 0.0.0.0.6728: S 105051945:105051945(0) win 512
e7:5c:97:42:ec:1 83:73:1a:32:20:93 0.0.0.0.45282 > 0.0.0.0.24898: S 1838062028:1838062028(0) win 512
62:69:d3:1c:79:ef 80:13:35:4:cb:d0 0.0.0.0.11587 > 0.0.0.0.7723: S 1792413296:1792413296(0) win 512
c5:a:b7:3e:3c:7a 3a:ee:c0:23:4a:fe 0.0.0.0.19784 > 0.0.0.0.57433: S 1018924173:1018924173(0) win 512
88:43:ee:51:c7:68 b4:8d:ec:3e:14:bb 0.0.0.0.283 > 0.0.0.0.11466: S 727776406:727776406(0) win 512
b8:7a:7a:2d:2c:ae c2:fa:2d:7d:e7:bf 0.0.0.0.32650 > 0.0.0.0.11324: S 605528173:605528173(0) win 512
e0:d8:1e:74:1:e 57:98:b6:5a:fa:de 0.0.0.0.36346 > 0.0.0.0.55700: S 2128143986:2128143986(0) win 512
```

# MAC/CAM Attacks

## MACOF Attack seen from Ethereal?

Using ETHEREAL to trace the progress of the MACOF attack...

The screenshot shows the Ethereal interface with a list of captured packets. The first packet is highlighted in blue. The details pane at the bottom shows the structure of this packet: Ethernet II, Internet Protocol, and Transmission Control Protocol.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	56.193.140.71	145.144.50.126	TCP	37603 > 11983 [SYN] Seq=101524
2	0.000009	188.78.123.38	106.134.69.40	TCP	54998 > 49433 [SYN] Seq=247924
3	0.000031	100.224.84.92	175.110.127.57	TCP	24604 > 58895 [SYN] Seq=717858
4	0.000040	144.236.195.103	69.11.95.61	TCP	13614 > 26423 [SYN] Seq=220195
5	0.000055	138.74.1.46	18.26.19.60	TCP	15963 > 20322 [SYN] Seq=182482
6	0.000065	158.68.184.76	64.237.135.45	TCP	8639 > 29042 [SYN] Seq=1222948
7	0.000081	136.102.115.5	67.72.95.5	TCP	27103 > 41257 [SYN] Seq=182495
8	0.000086	23.246.57.5	75.251.236.3	TCP	42693 > 14764 [SYN] Seq=119904
9	0.000095	163.125.67.79	14.241.212.73	TCP	34606 > 61093 [SYN] Seq=135098
10	0.000110	204.61.32.113	141.102.6.83	TCP	10778 > 39115 [SYN] Seq=311150
11	0.000131	184.6.89.20	142.179.92.18	TCP	32771 > 49622 [SYN] Seq=734385
12	0.000141	157.132.183.117	13.140.204.98	TCP	58443 > 387 [SYN] Seq=14767240
13	0.000156	147.155.142.1	20.78.43.127	TCP	37047 > 51004 [SYN] Seq=954248
14	0.000164	83.51.161.45	0.140.245.68	TCP	46773 > 24824 [SYN] Seq=162871
15	0.000180	141.188.130.127	26.183.116.7	TCP	16484 > 41051 [SYN] Seq=944146
16	0.000189	69.19.122.111	12.198.122.52	TCP	10079 > 45431 [SYN] Seq=527207

Frame 1 (60 bytes on wire, 60 bytes captured)  
Ethernet II, Src: f2:0f:13:22:f6:4f, Dst: 82:26:30:3b:8a:4f  
Internet Protocol, Src Addr: 56.193.140.71 (56.193.140.71), Dst Addr: 145.144.50.126 (145.144.50.126)  
Transmission Control Protocol, Src Port: 37603 (37603), Dst Port: 11983 (11983), Seq: 10152419

# CAM Table Sizes

- **Each switch has a limit on CAM tables**
- **Size by basic switch**
  - 3xxx—16,000**
  - 4xxx—32,000**
  - 6xxx—128,000**

# CAM Table FULL!

- Once you have flooded the CAM table, packets from any new session will be forwarded out all switch ports (default switch behavior), allowing a malicious user to run a packet sniffer to capture users traffic...
- This will turn a VLAN on a switch basically into a hub
- This attack will also fill the CAM tables of adjacent switches

10.1.1.22 -> (broadcast) ARP C Who is 10.1.1.1, 10.1.1.1 ?

10.1.1.22 -> (broadcast) ARP C Who is 10.1.1.19, 10.1.1.19 ?

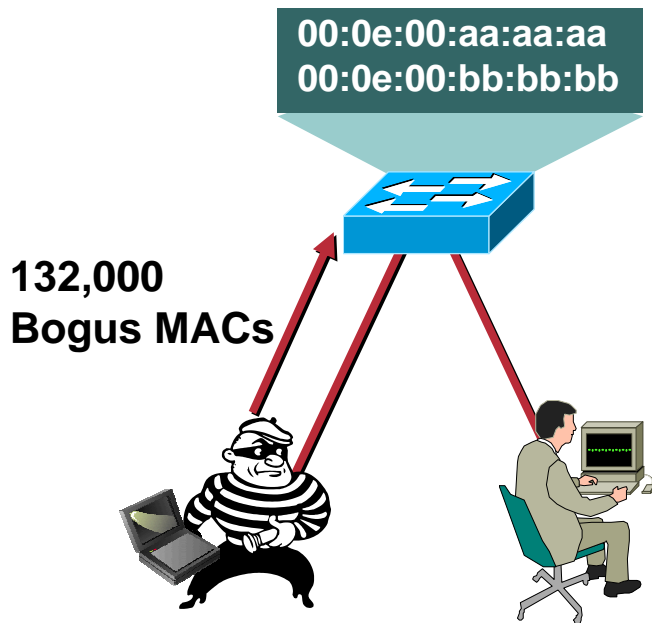
10.1.1.26 -> 10.1.1.25 ICMP Echo request (ID: 256 Sequence number: 7424) ← OOPS

10.1.1.25 -> 10.1.1.26 ICMP Echo reply (ID: 256 Sequence number: 7424) ← OOPS

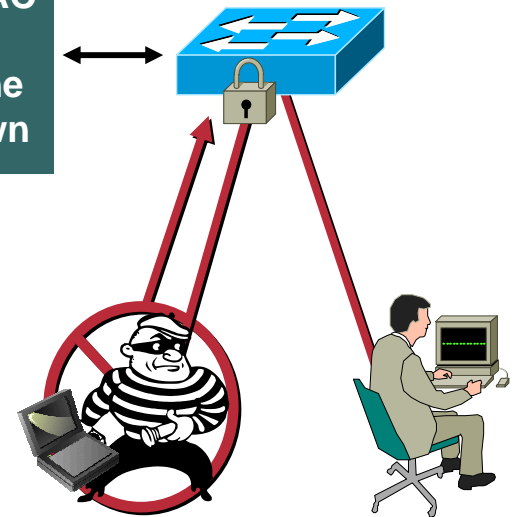


# Countermeasures for MAC Attacks

## Port Security Limits the Amount of MAC's on an Interface



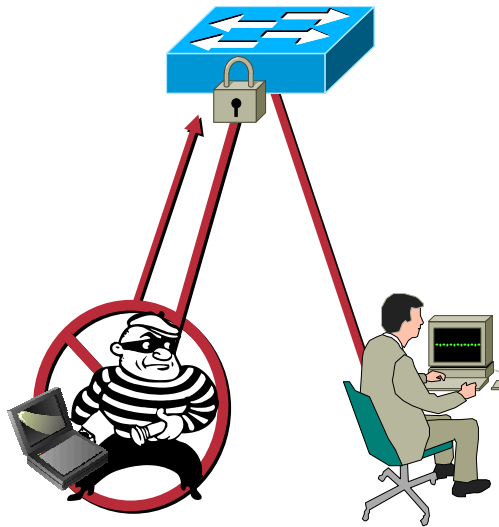
Only Three MAC  
Addresses  
Allowed on the  
Port: Shutdown



### Solution:

- Port security limits MAC flooding attack and locks down port and sends an SNMP trap

# Port Security: Example Config



## CatOS

```
set port security 5/1 enable
set port security 5/1 port max 3
set port security 5/1 violation restrict
set port security 5/1 age 2
set port security 5/1 timer-type inactivity
```

## IOS®

```
switchport port-security
switchport port-security maximum 3
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
```

- Three MAC addresses encompass the phone, the switch in the phone, and the PC
- “Restrict” rather than “error disable” to allow only three, and log more than three
- Aging time of two and aging type inactivity to allow for phone CDP of one minute

If Violation Error-Disable, the Following Log Message Will Be Produced: 4w6d: %PM-4-ERR\_DISABLE: Psecure-Violation Error Detected on Gi3/2, Putting Gi3/2 in Err-Disable State

## Not All Port Security Created Equal

- In the past you would have to type in the **ONLY MAC** you were going to allow on that port
- You can now put a **limit** to how many MAC address a port will learn
- You can also put **timers** in to state how long the MAC address will be bound to that switch port
- You might still want to do **static MAC entries** on ports that there should be no movement of devices, as in server farms
- If you are going to be running **Cisco IPT**, you will need a minimum of three MAC addresses on each port if you are running voice VLANs
- New feature called “**Sticky Port Security**”, settings will survive reboot (not on all switches)

# Building the Layers



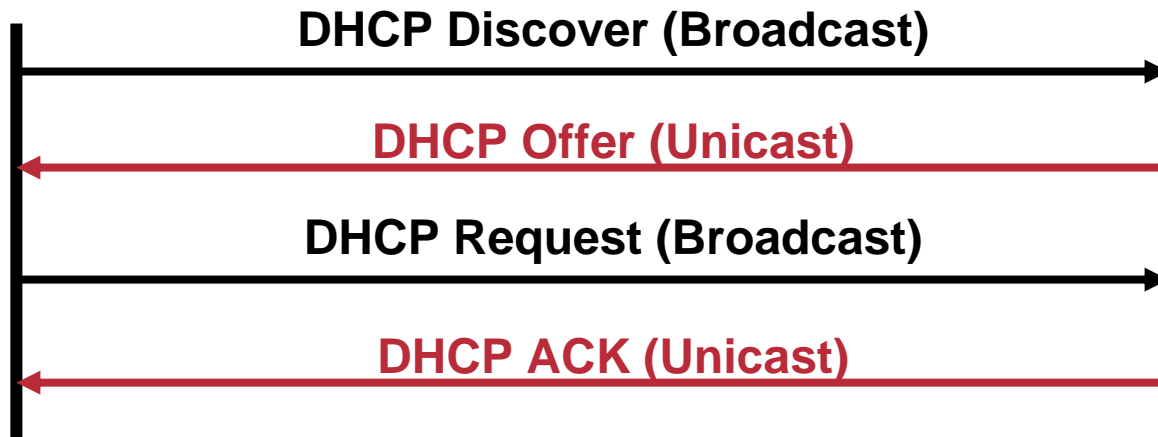
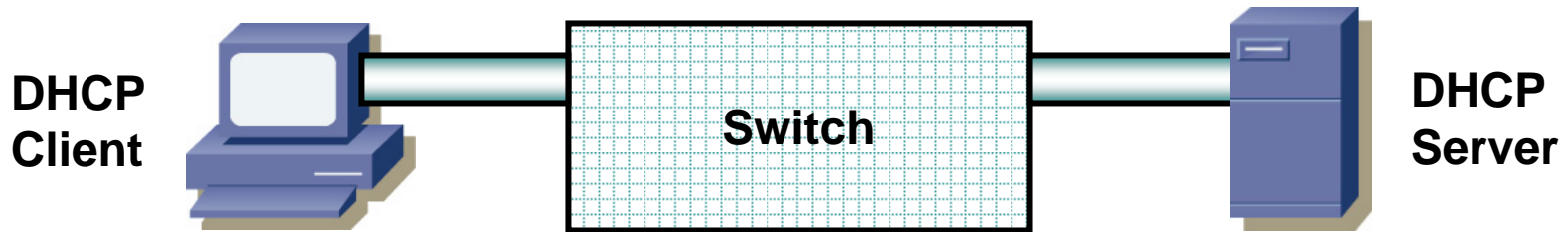
- **Port Security prevents CAM attacks and DHCP starvation attacks**

# ATTACKS AND COUNTERMEASURES: DHCP ATTACKS



# DHCP Refresher

Dynamic Host Configuration Protocol is designed to serve IP addresses to requesting hosts from a pool of addresses setup by an administrator...



# DHCP Request

No. .	Time	Source	Destination	Protocol	Info
96	39.746002	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transac
97	39.750603	10.66.227.33	10.66.227.34	DHCP	DHCP ACK - Transac

Frame 96 (342 bytes on wire, 342 bytes captured)

Ethernet II, Src: 00:09:6b:90:f5:70, Dst: ff:ff:ff:ff:ff:ff

Internet Protocol, Src Addr: 0.0.0.0 (0.0.0.0), Dst Addr: 255.255.255.255 (255.255.255.255)

User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)

Bootstrap Protocol

- Message type: Boot Request (1)
- Hardware type: Ethernet
- Hardware address length: 6
- Hops: 0
- Transaction ID: 0xb04db74d
- Seconds elapsed: 0
- Bootp flags: 0x0000 (Unicast)
- Client IP address: 0.0.0.0 (0.0.0.0)
- Your (client) IP address: 0.0.0.0 (0.0.0.0)
- Next server IP address: 0.0.0.0 (0.0.0.0)
- Relay agent IP address: 0.0.0.0 (0.0.0.0)
- Client hardware address: 00:09:6b:90:f5:70
- Server host name not given
- Boot file name not given
- Magic cookie: (OK)
- Option 53: DHCP Message Type = DHCP Request
- Option 50: Requested IP Address = 10.66.227.34
- Option 55: Parameter Request List
  - 1 = Subnet Mask
  - 28 = Broadcast Address
  - 2 = Time Offset
  - 3 = Router
  - 15 = Domain Name
  - 6 = Domain Name Server
  - 12 = Host Name
  - 40 = Network Information Service Domain
  - 41 = Network Information Service Servers
  - 42 = Network Time Protocol Servers
- End option

**Broadcast**

**What the host is requesting**

# DHCP ACK (the reply)

No. .	Time	Source	Destination	Protocol	Info
96	39.746002	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transac
97	39.750603	10.66.227.33	10.66.227.34	DHCP	DHCP ACK - Transac

Frame 97 (343 bytes on wire, 343 bytes captured)

Ethernet II, Src: 00:02:16:0d:f3:dc, Dst: 00:09:6b:90:f5:70

Internet Protocol, Src Addr: 10.66.227.33 (10.66.227.33), Dst Addr: 10.66.227.34 (10.66.227.34)

User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)

Bootstrap Protocol

Message type: Boot Reply (2)  
Hardware type: Ethernet  
Hardware address length: 6  
Hops: 0  
Transaction ID: 0xb04db74d  
Seconds elapsed: 0

Bootp flags: 0x0000 (Unicast)  
Client IP address: 0.0.0.0 (0.0.0.0)  
Your (client) IP address: 10.66.227.34 (10.66.227.34)  
Next server IP address: 0.0.0.0 (0.0.0.0)  
Relay agent IP address: 0.0.0.0 (0.0.0.0)  
Client hardware address: 00:09:6b:90:f5:70  
Server host name not given  
Boot file name not given  
Magic cookie: (OK)  
Option 53: DHCP Message Type = DHCP ACK  
Option 54: Server Identifier = 10.66.227.33  
Option 51: IP Address Lease Time = 1 day  
Option 58: Renewal Time value = 12 hours  
Option 59: Rebinding Time value = 21 hours  
Option 1: subnet Mask = 255.255.255.248

Option 6: Domain Name Server  
IP Address: 64.104.200.248  
IP Address: 171.70.168.183

Option 3: Router = 10.66.227.33  
Option 15: Domain Name = "cisco.com"

End Option

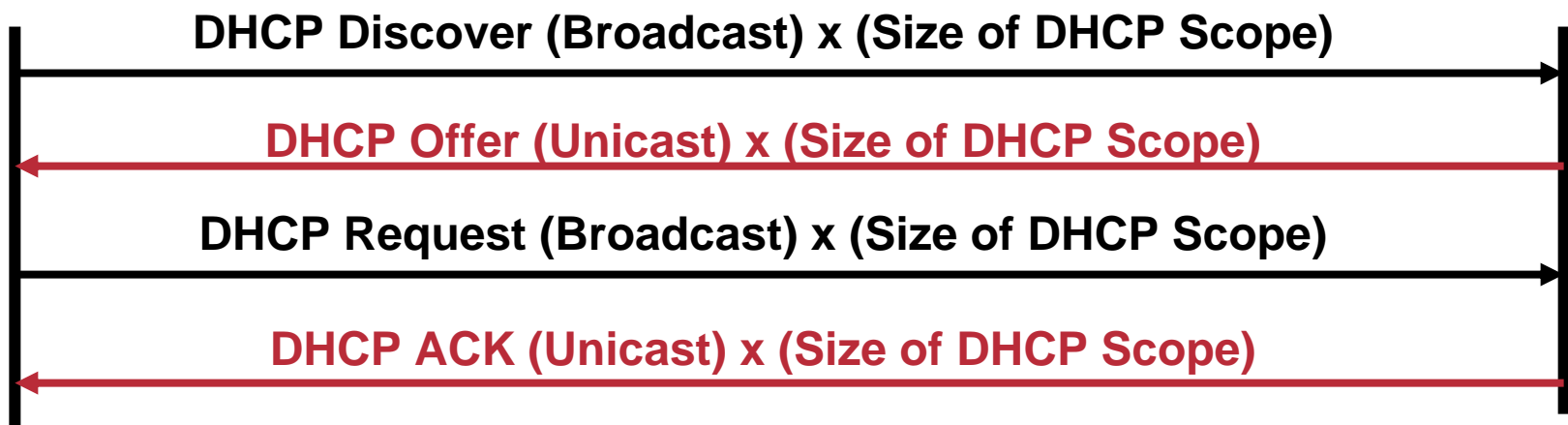
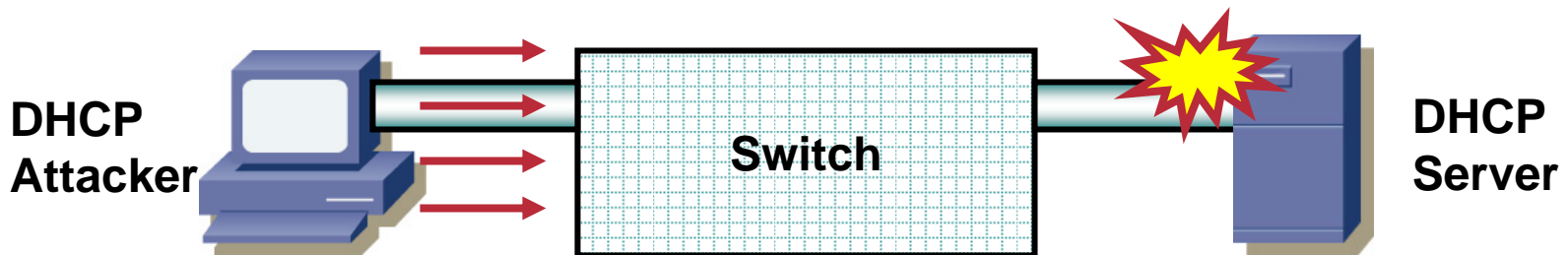
Unicast

DHCP Address Details for the requesting host



# DHCP Starvation Attack

Denial of Service (DoS) attack that can be used to grab **ALL** the addresses from the DHCP server – Now all other hosts requesting DHCP address will be denied from accessing network due to no available addresses...



# DHCP Attack Tools

Denial of Service (DoS) attack that can be used to grab ALL the addresses from the DHCP server so that no other host can get onto the network...



```
DHCP gobbler v0.66 from www.networkpenetration.com
```

## Attack Options

```
-g to gobble all available IP's
```

```
-m <#> Start a man in the middle attack, # indicates number of connections to initially gobble
```

## Man in the middle attack options

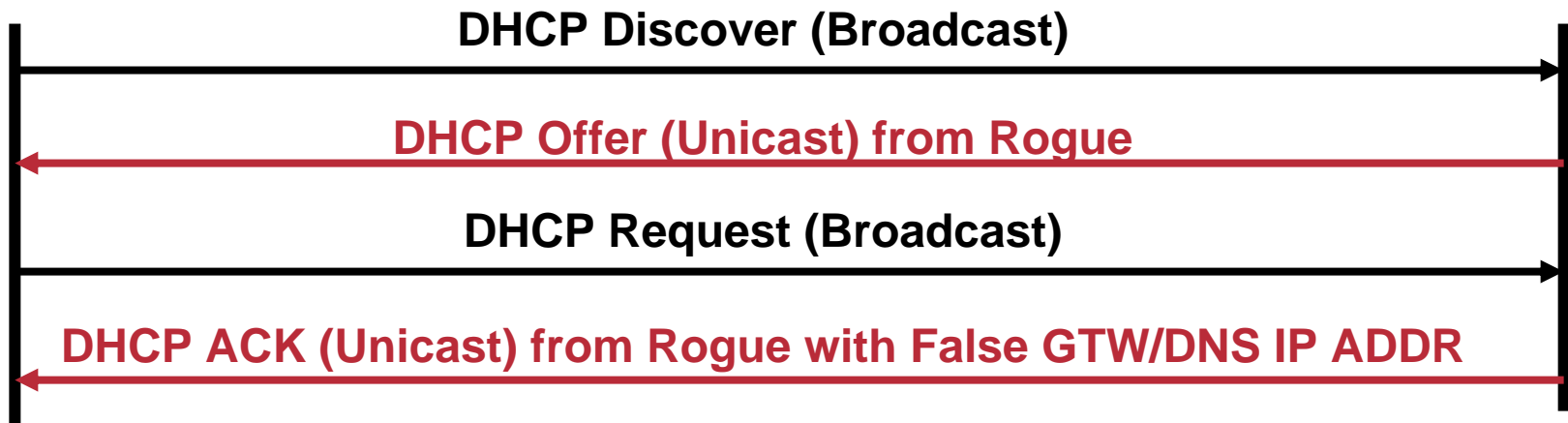
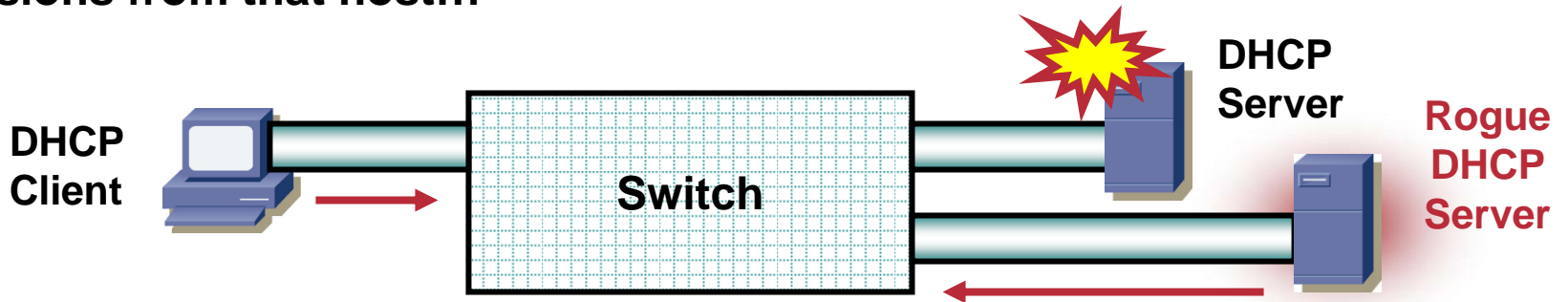
```
-D <IP Address> IP address of spoofed DNS server
```

```
-G <IP Address> IP address of spoofed default gateway
```

```
-A <IP Address> IP address of fake DHCP server (can be spoofed)
```

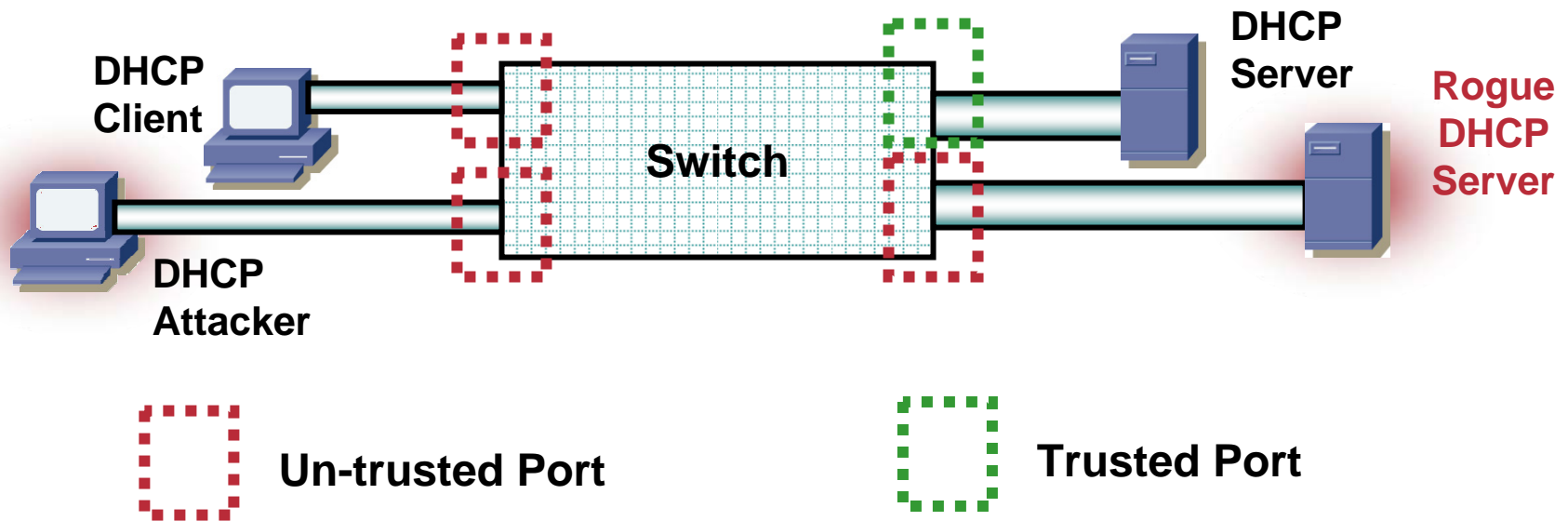
# DHCP Rogue Server Attack

The Rogue DHCP Server can issue DHCP Requests pointing to false DNS or Gateway (maybe use its own IP for this) – allowing it to snoop on sessions from that host...



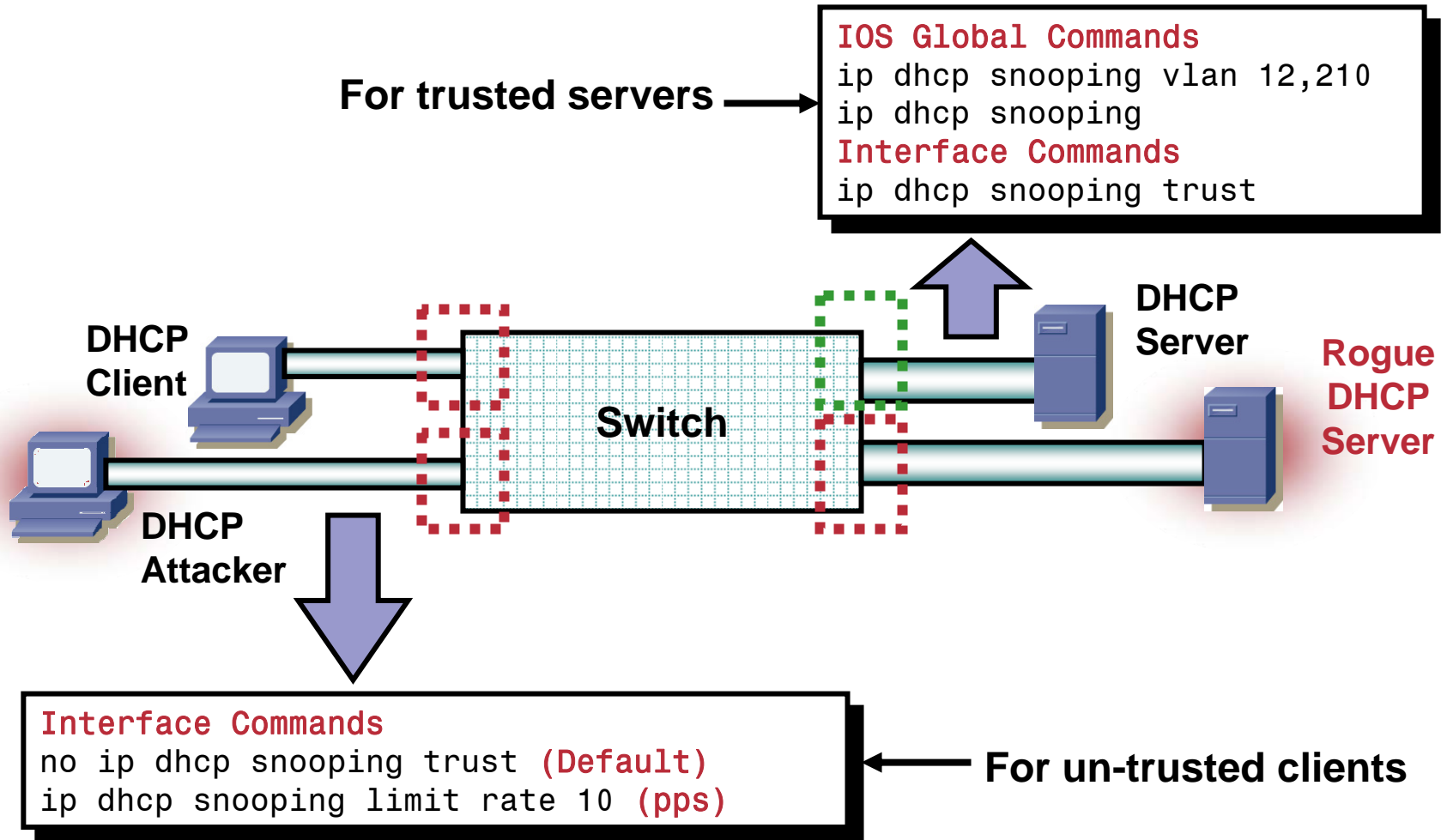
# Mitigating DHCP Attacks

The new DHCP Snooping Feature prevents MITM attacks and DoS attacks on the DHCP server

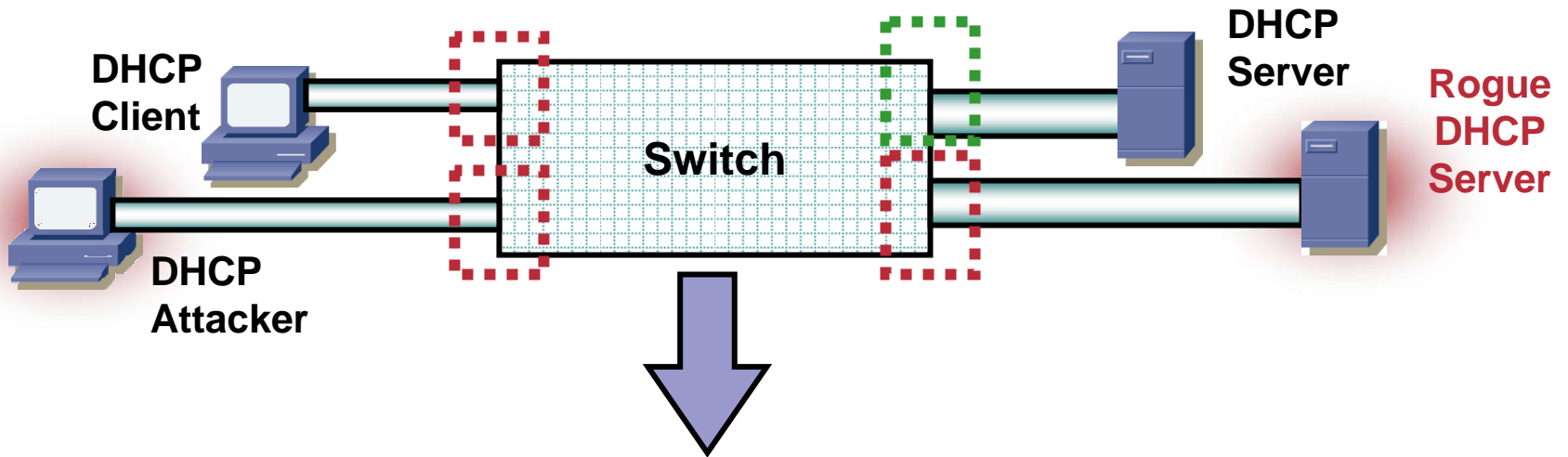


When DHCP Packets originate from an UNTRUSTED port – only Client Requests are forwarded – all other DHCP packets are dropped (DHCP Offer, DHCP ACK, NACK or other DHCP server orientated packets...

# Mitigating DHCP Attacks



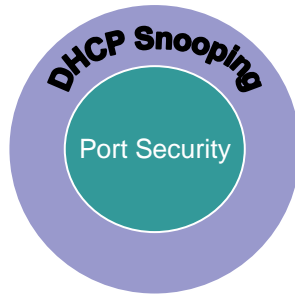
# DHCP Binding Table



```
Switch# show ip dhcp snooping binding
MacAddress      IpAddress      Lease(sec)    Type           VLAN  Interface
-----
00:00:09:f3:2a:19  10.66.227.33  138225       dhcp-snooping  12   GigabitEthernet4/12
```

**This binding table is the same one used for other security features like IP Source Guard and Dynamic ARP Inspection**

# Building the Layers



- **Port Security prevents CAM Attacks and DHCP Starvation attacks**
- **DHCP Snooping prevents Rogue DHCP Server attacks**

# ATTACKS AND COUNTERMEASURES: ARP ATTACKS



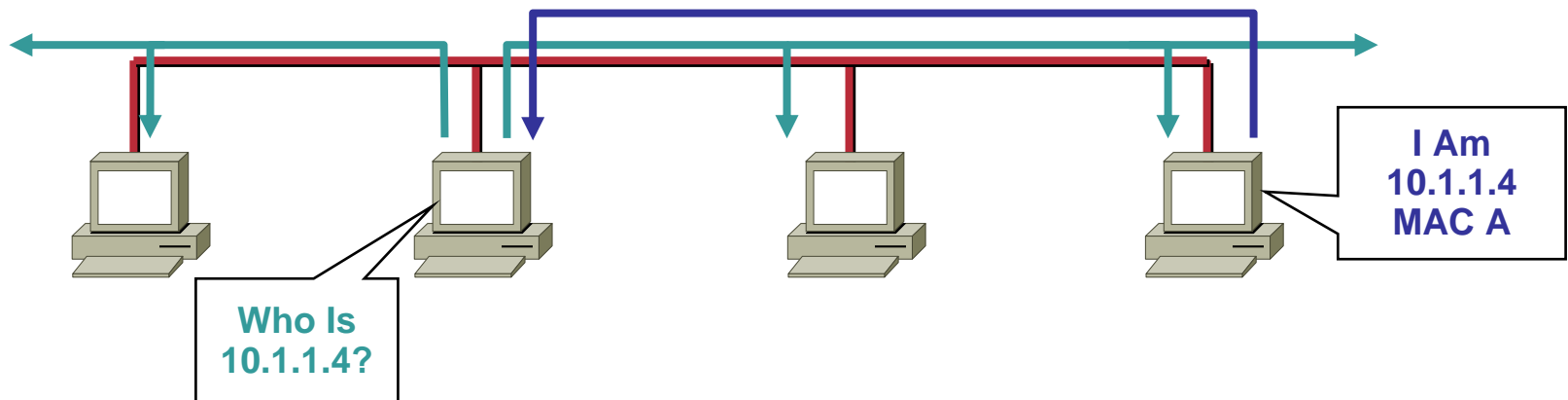


# ARP Function Review

- Before a station can talk to another station it must do an ARP request to map the IP address to the MAC address

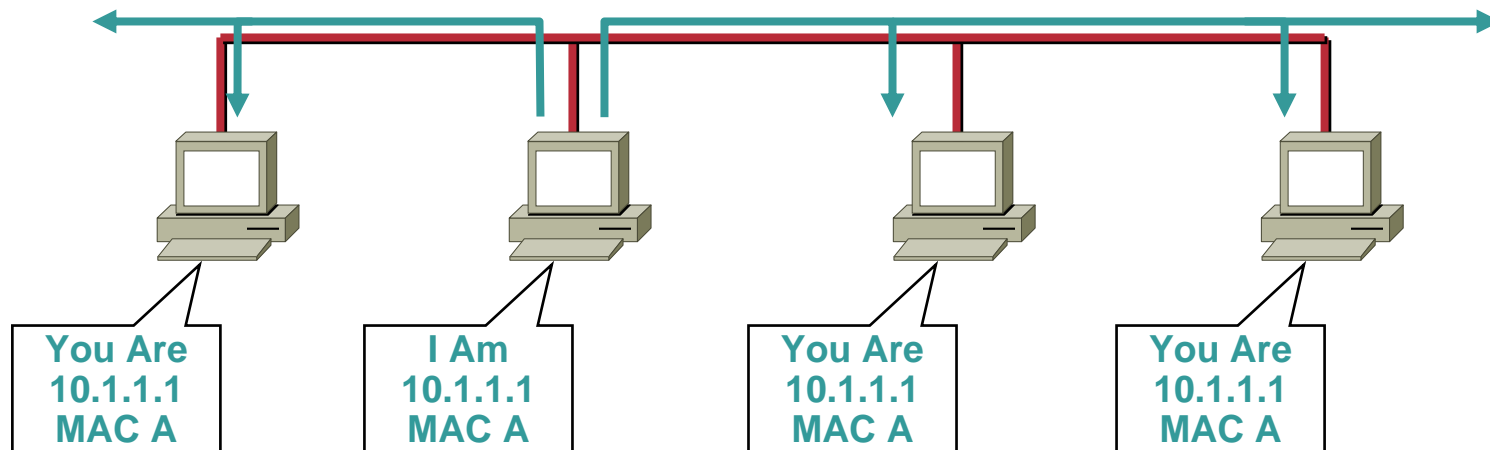
This **ARP request** is broadcast using **protocol 0806**

- All computers on the subnet will receive and process the ARP request; the station that matches the IP address in the request will send an ARP reply



# ARP Function Review

- According to the ARP RFC, a client is allowed to send an unsolicited ARP reply; this is called a gratuitous ARP; other hosts on the same subnet can store this information in their ARP tables
- Anyone can claim to be the owner of any IP/MAC address they like
- ARP attacks use this to redirect traffic



# ARP Attack Tools

- **Two major tools on the Net for ARP man-in-the-middle attacks**

**dsniff**—<http://monkey.org/~dugsong/dsniff/>

**ettercap**—<http://ettercap.sourceforge.net/index.php>

Both “tools” function similar to each other

- **ettercap is the second generation of ARP attack tools**

ettercap has a nice GUI, and is almost point and click

Interesting features of ettercap

Packet Insertion, many to many ARP attack

- **Both capture the traffic/passwords of applications (over 30)**

FTP, Telnet, SMTP, HTTP, POP, NNTP, IMAP, SNMP, LDAP, RIP, OSPF, PPTP, MS-CHAP, SOCKS, X11, IRC, ICQ, AIM, SMB, Microsoft SQL

# ARP Attack Tools

- Ettercap in action
- As you can see runs in Window, Linux, Mac
- Decodes passwords on the fly
- This example, telnet username/ password is captured

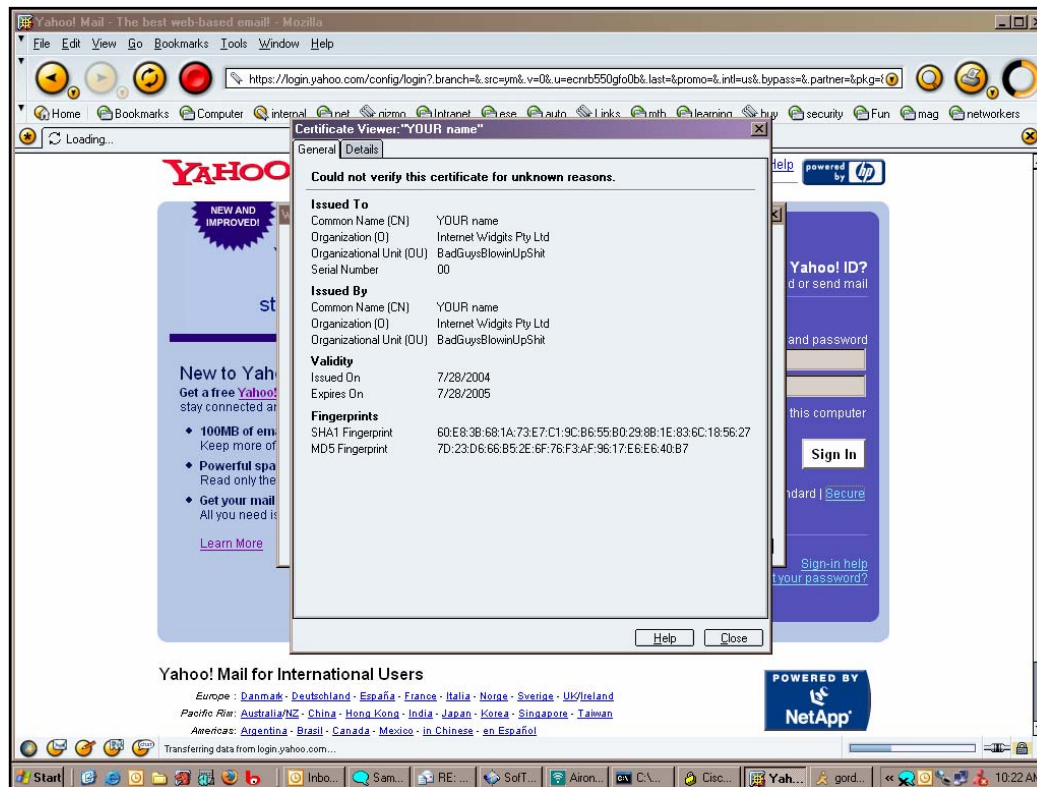
```
root@ngcs-p01:~# ettercap 0.6.b
SOURCE: 10.10.10.20 <-- Filter: OFF
DEST : 10.10.10.64 <-- doppleganger - illithid (ARP Based) - ettercap
Active Dissector: ON

-----
4 hosts in this LAN (10.10.10.62 : 255.255.255.0)
-----
1) 10.10.10.64:137 <--> 10.10.10.20:137 UDP netbios-ns
2) 10.10.10.20:1687 <--> 10.10.10.64:139 CLOSED netbios-ssn
3) 10.10.10.20:1688 <--> 10.10.10.64:23 silent telnet

-----
Your IP: 10.10.10.62 MAC: 00:03:47:2D:8B:0F Iface: eth1 Link: SWITCH
USER: administrator
PASS: cisco
```

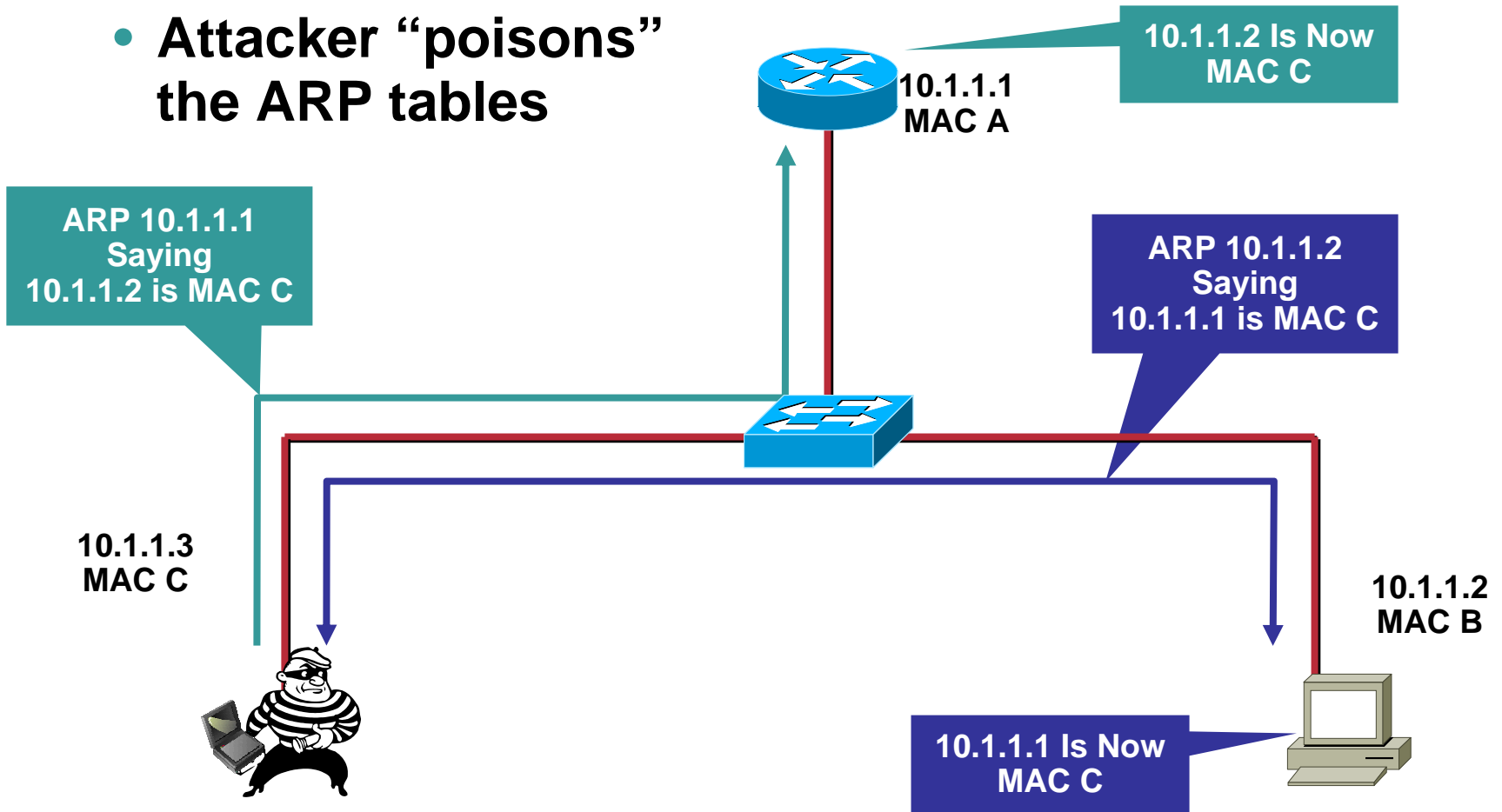
# ARP Attack Tools: SSH/SSL

- Using these tools SSL/SSH sessions can be intercepted and bogus certificate credentials can be presented
- Once you have accepted the certificate, all SSL/SSH traffic for all SSL/SSH sites can flow through the attacker



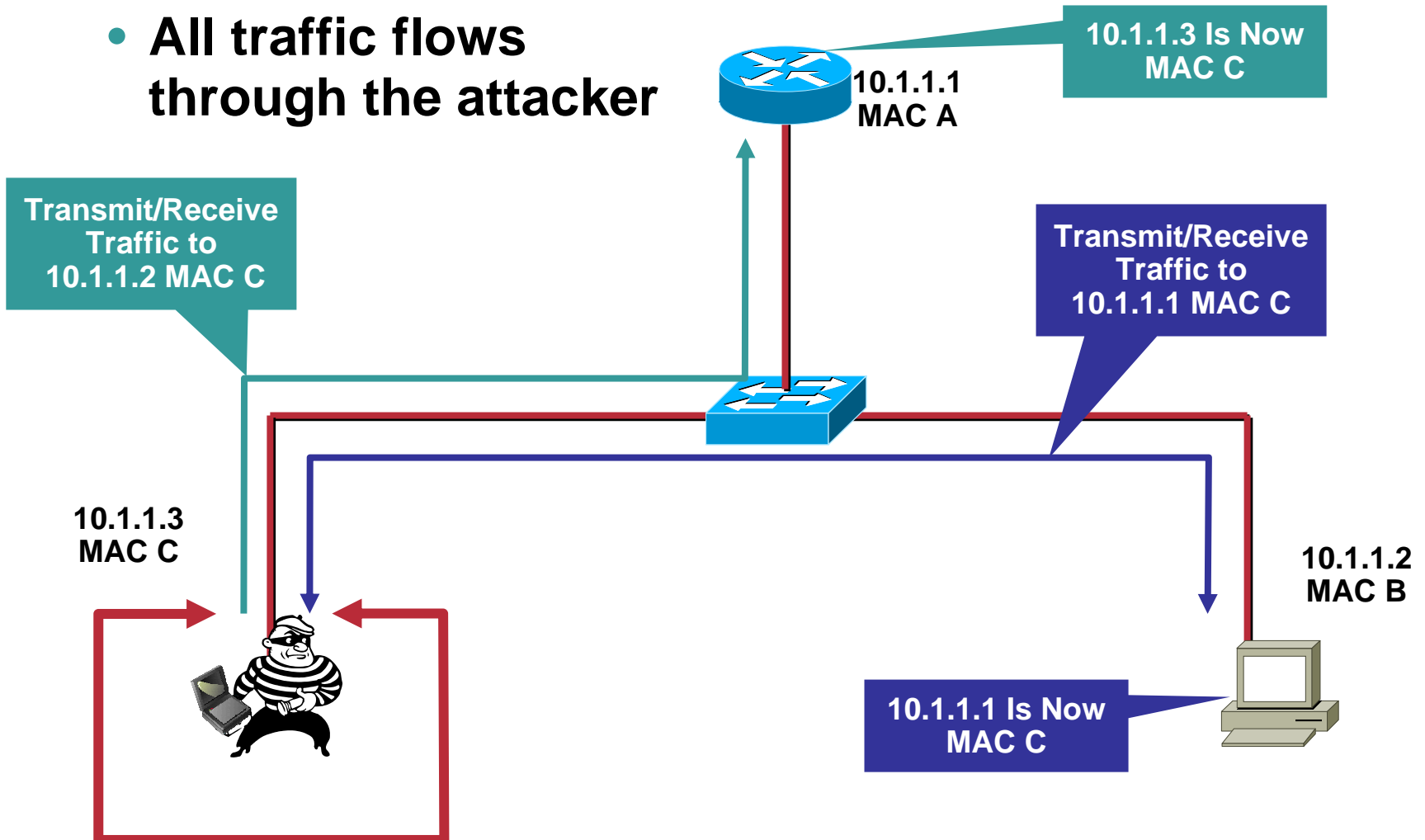
# ARP Attack in Action

- Attacker “poisons” the ARP tables



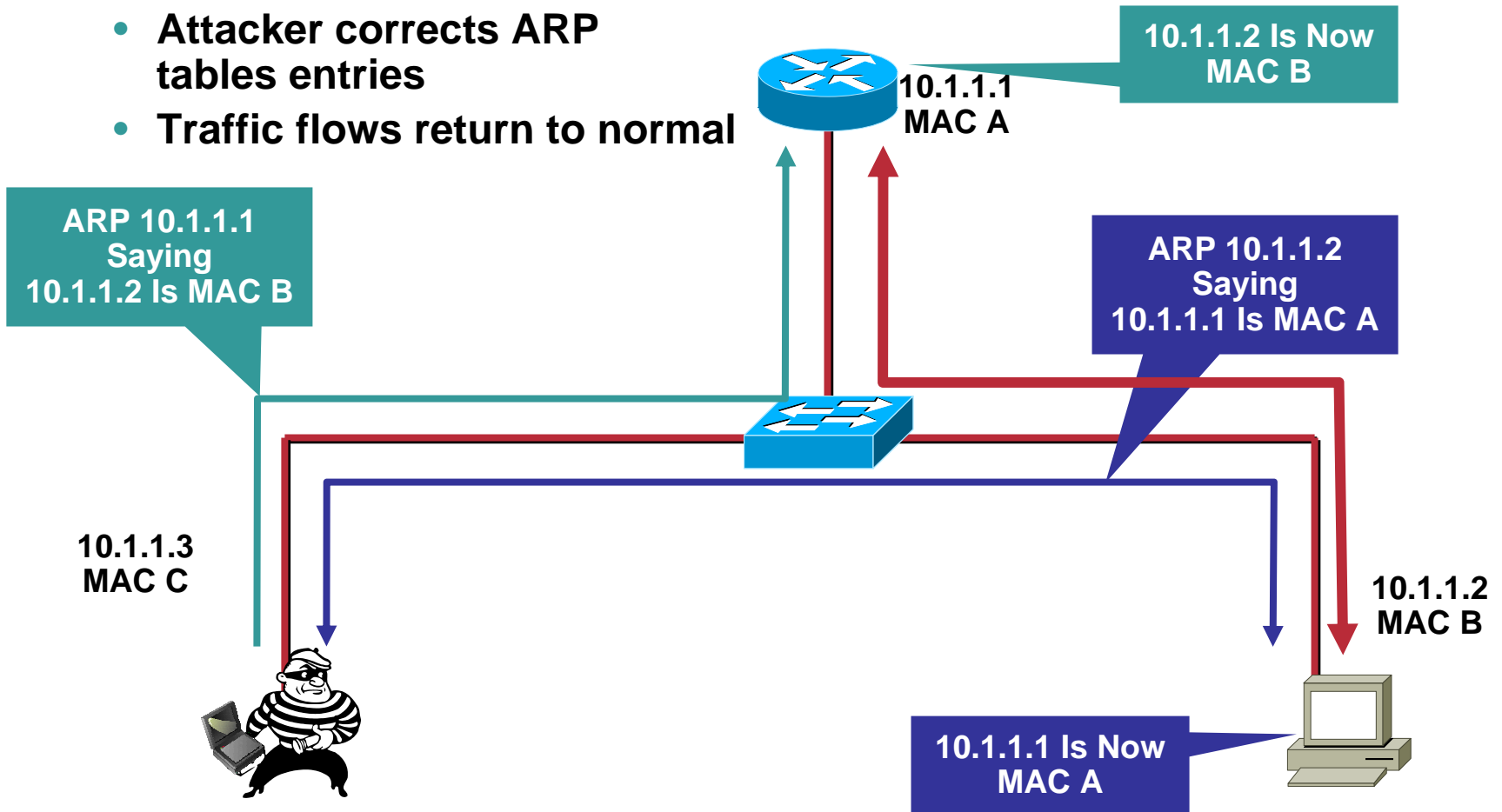
# ARP Attack in Action

- All traffic flows through the attacker



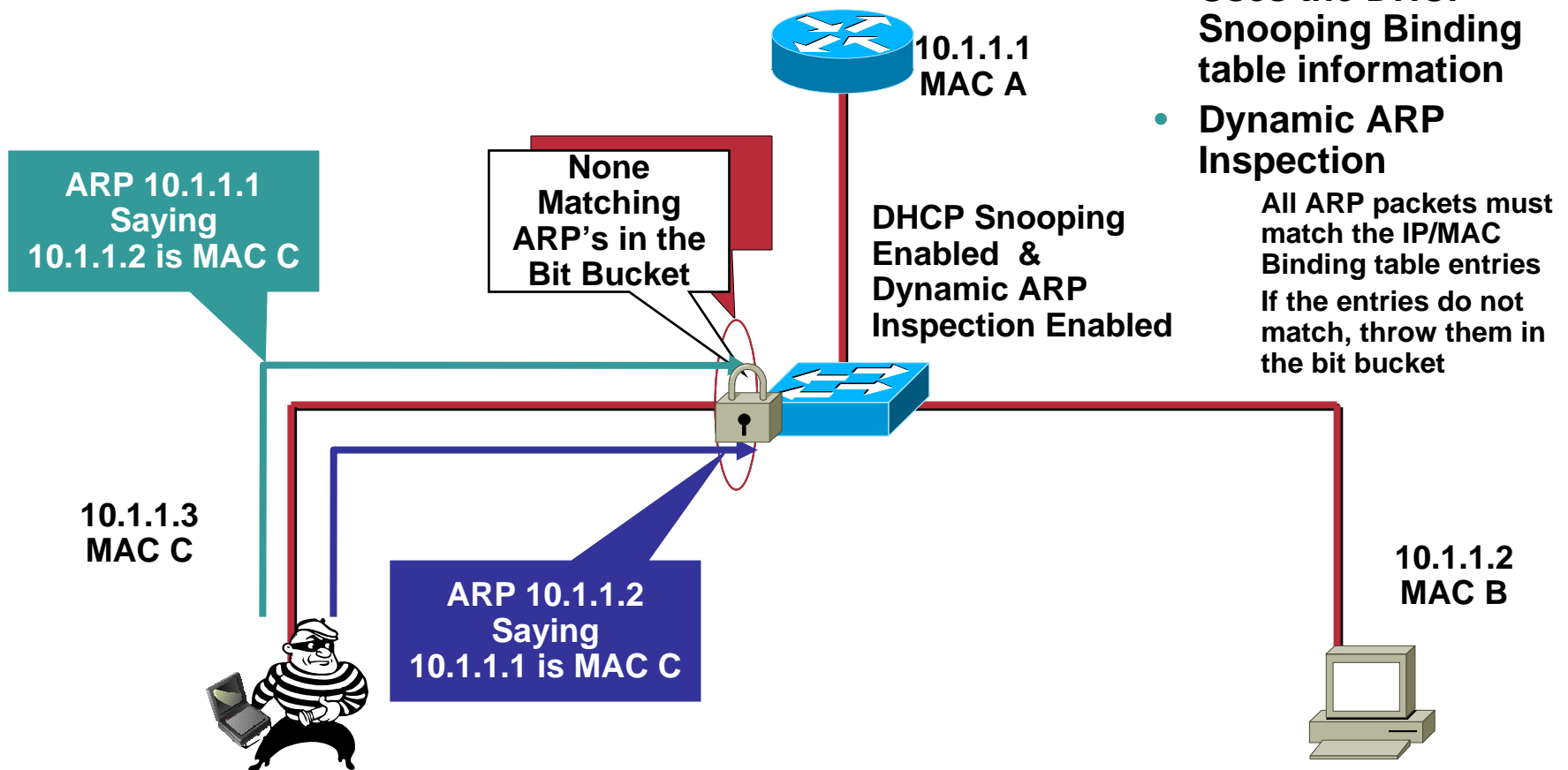
# ARP Attack Clean Up

- Attacker corrects ARP tables entries
- Traffic flows return to normal





# Countermeasures to ARP Attacks: Dynamic ARP Inspection



# Countermeasures to ARP Attacks: Dynamic ARP Inspection

- **Uses the information from the DHCP Snooping Binding table**

```
sh ip dhcp snooping binding
```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
00:03:47:B5:9F:AD	10.120.4.10	193185	dhcp-snooping	4	FastEthernet3/18
00:03:47:c4:6f:83	10.120.4.11	213454	dhcp-snooping	4	FastEthernet3/21

- **Looks at the MacAddress and IpAddress fields to see if the ARP from the interface is in the binding, if not, traffic is blocked**

# Countermeasures to ARP Attacks: Dynamic ARP Inspection

## Configuration of Dynamic ARP Inspection (DAI)

- **DHCP Snooping had to be configured so the binding table it built**
- **DAI is configured by VLAN**
- **You can trust an interface like DHCP Snooping**
- **Be careful with rate limiting—varies between platforms**
- **Suggested for voice is to set the rate limit above the default if you feel dial tone is important**

# Countermeasures to ARP Attacks: Dynamic ARP Inspection

## Dynamic ARP Inspection Commands

### *IOS*

#### *Global Commands*

```
ip dhcp snooping vlan 4,104
no ip dhcp snooping information option
ip dhcp snooping
ip arp inspection vlan 4,104
ip arp inspection log-buffer entries 1024
ip arp inspection log-buffer logs 1024 interval 10
```

#### *Interface Commands*

```
ip dhcp snooping trust
ip arp inspection trust
```

### *IOS*

#### *Interface Commands*

```
no ip arp inspection trust
(default)
ip arp inspection limit rate 15
(pps)
```

# Countermeasures to ARP Attacks: Dynamic ARP Inspection

Cisco.com

## Error Messages in Show Log

```
sh log:
4w6d: %SW_DAI-4-PACKET_RATE_EXCEEDED: 16 packets received in 296 milliseconds on Gi3/2.
4w6d: %PM-4-ERR_DISABLE: arp-inspection error detected on Gi3/2, putting Gi3/2 in err-disable state
4w6d: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Gi3/2, vlan
183.([0003.472d.8b0f/10.10.10.62/0000.0000.0000/10.10.10.2/12:19:27 UTC Wed Apr 19 2000])
4w6d: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Gi3/2, vlan
183.([0003.472d.8b0f/10.10.10.62/0000.0000.0000/10.10.10.3/12:19:27 UTC Wed Apr 19 2000])
```

# Non DHCP Devices

- **Can use Static bindings in the DHCP Snooping Binding table**

*IOS*

*Global Commands*

```
ip source binding 0000.0000.0001 vlan 4 10.0.10.200 interface fastethernet 3/1
```

- **Show static and dynamic entries in the DHCP Snooping Binding table is different**

*IOS*

*Show Commands*

```
show ip source binding
```

# Binding Table Info

- **No entry in the binding table—no traffic!**
- **Wait until all devices have new leases before turning on Dynamic ARP Inspection**
- **Entries stay in table until the lease runs out**
- **All switches have a binding size limit**
  - 3000 switches—1,000 entries**
  - 4000 switches—2,000 entries (6000 for the SupV-10GE)**
  - 6000 switches—16,000 entries**

# Summary of ARP Attacks

- **Dynamic ARP Inspection prevents ARP attacks by intercepting all ARP requests and responses**
- **DHCP Snooping must be configured first, otherwise there is no binding table for dynamic ARP Inspection to use**
- **The DHCP Snooping table is built from the DHCP request, but you can put in static entries**

**If you have a device that does not DHCP, but you would like to turn on Dynamic ARP Inspection, you would need a static entry in the table**



# More ARP Attack Information

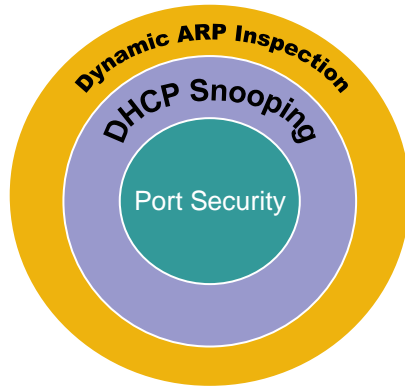
- **Some IDS systems will watch for an unusually high amount of ARP traffic**
- **ARPWatch is freely available tool to track IP/MAC address pairings**

**Caution—you will need an ARPWatch server on every VLAN**

**Hard to manage and scale**

**You can still do static ARP for critical routers and hosts (administrative pain)**

# Building the Layers



- **Port security prevents CAM attacks and DHCP Starvation attacks**
- **DHCP snooping prevents rogue DHCP server attacks**
- **Dynamic ARP inspection prevents current ARP attacks**

# ATTACKS AND COUNTERMEASURES: SPOOFING ATTACKS



# spoofing Attacks

- **MAC spoofing**

**If MACs are used for network access an attacker can gain access to the network**

**Also can be used to take over someone's identity already on the network**

- **IP spoofing**

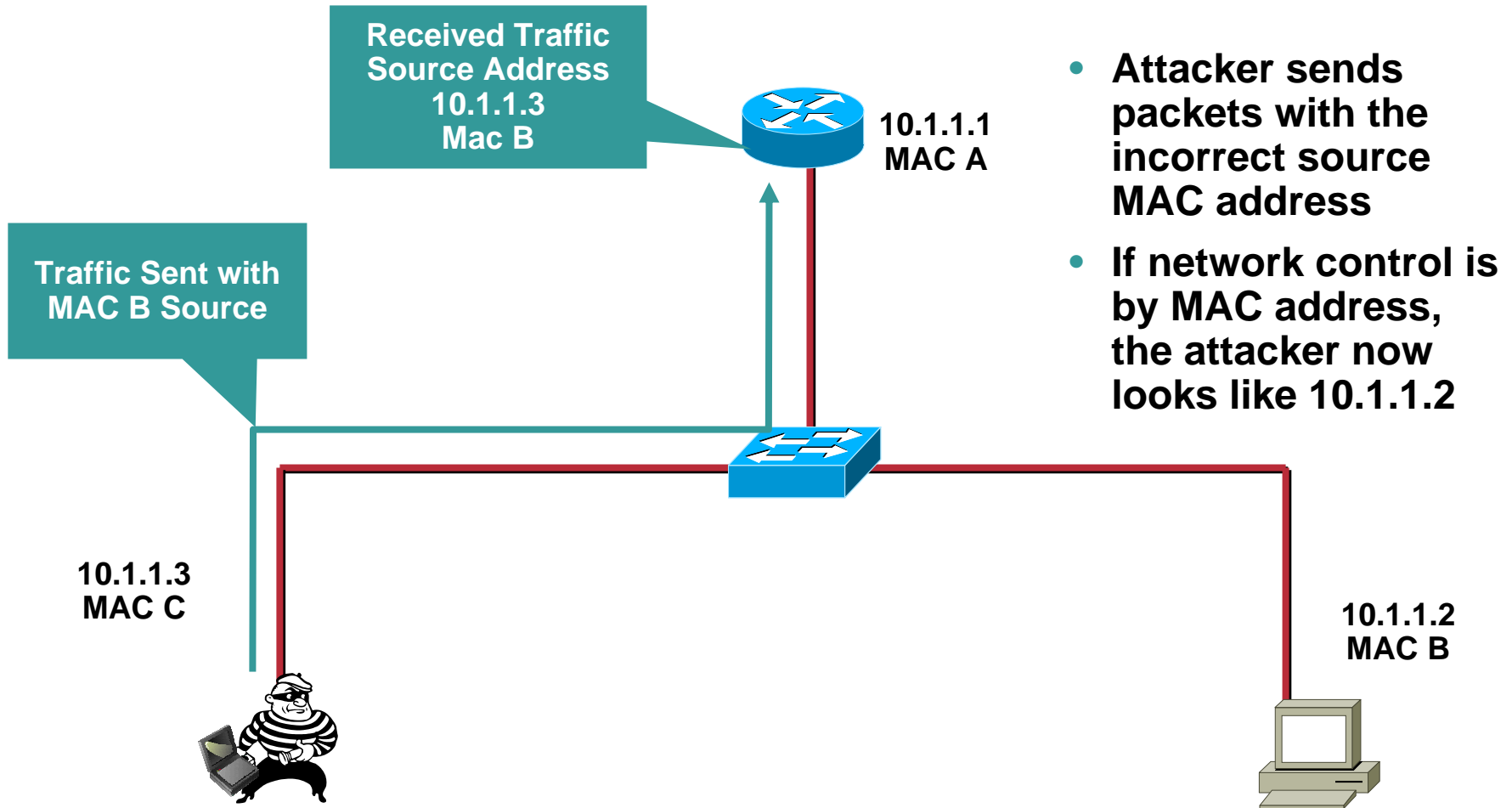
**Ping of death**

**ICMP unreachable storm**

**SYN flood**

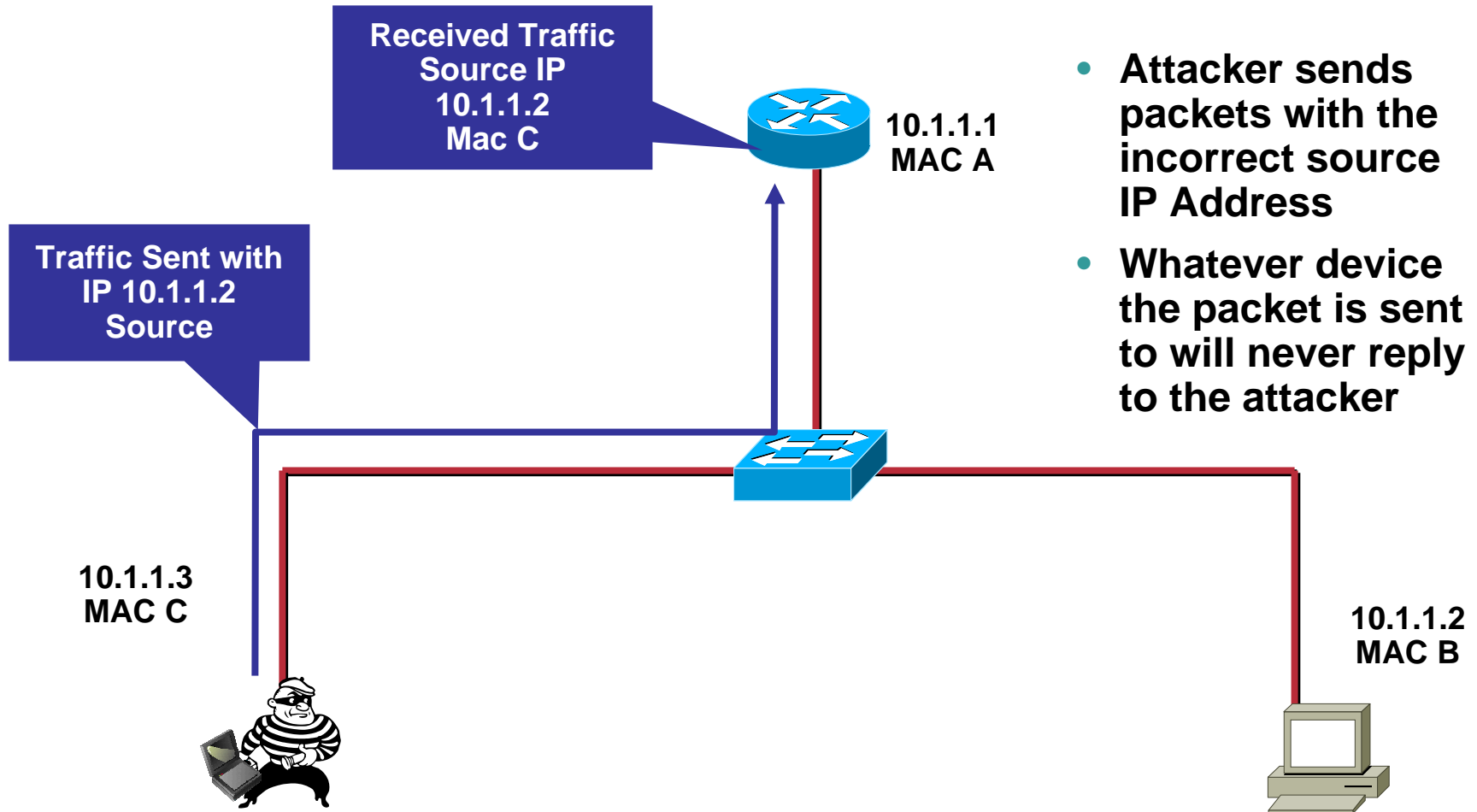
**Trusted IP addresses can be spoofed**

# Spoofing Attack: MAC



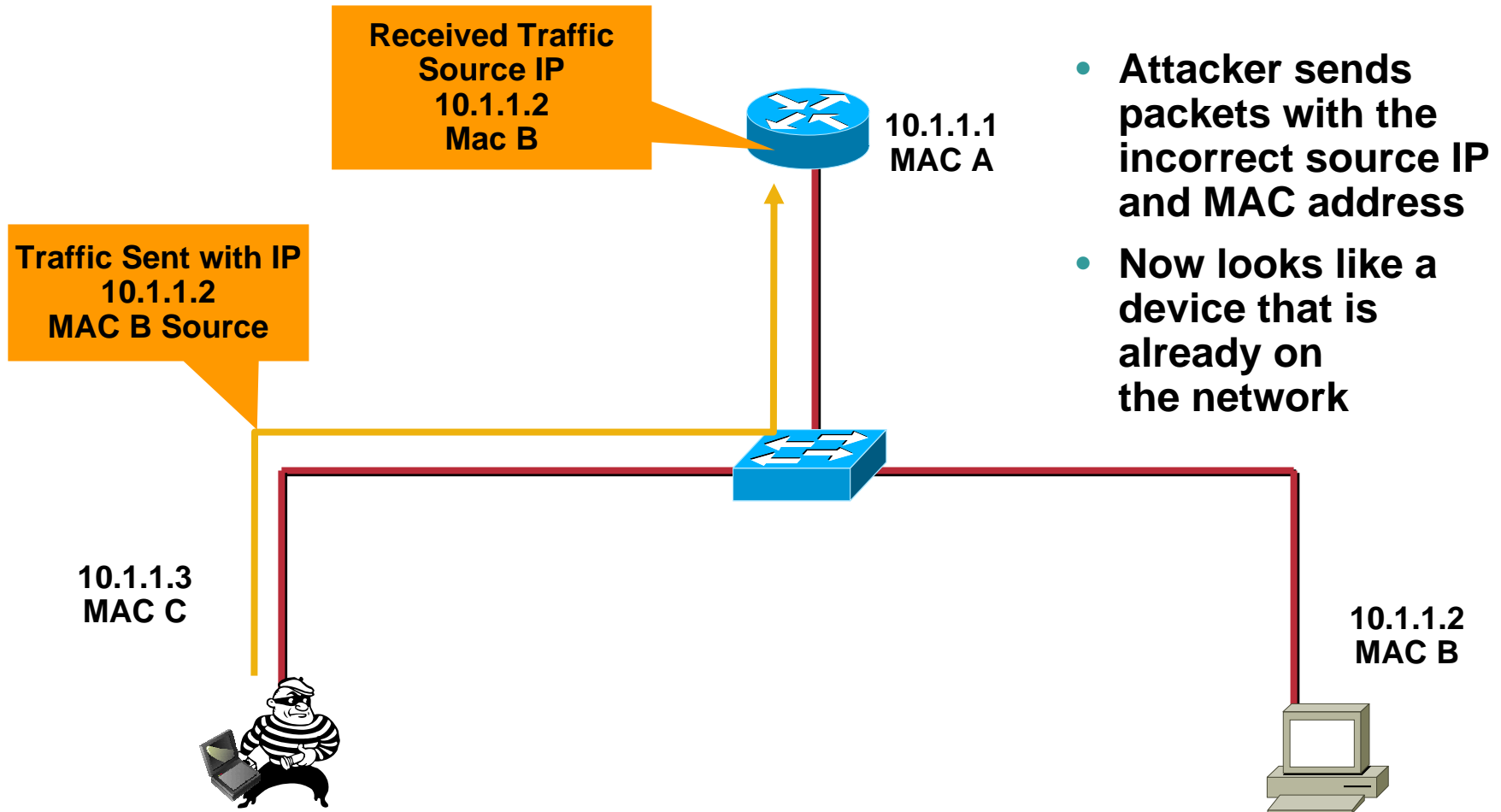
- Attacker sends packets with the incorrect source MAC address
- If network control is by MAC address, the attacker now looks like 10.1.1.2

# Spoofing Attack: IP

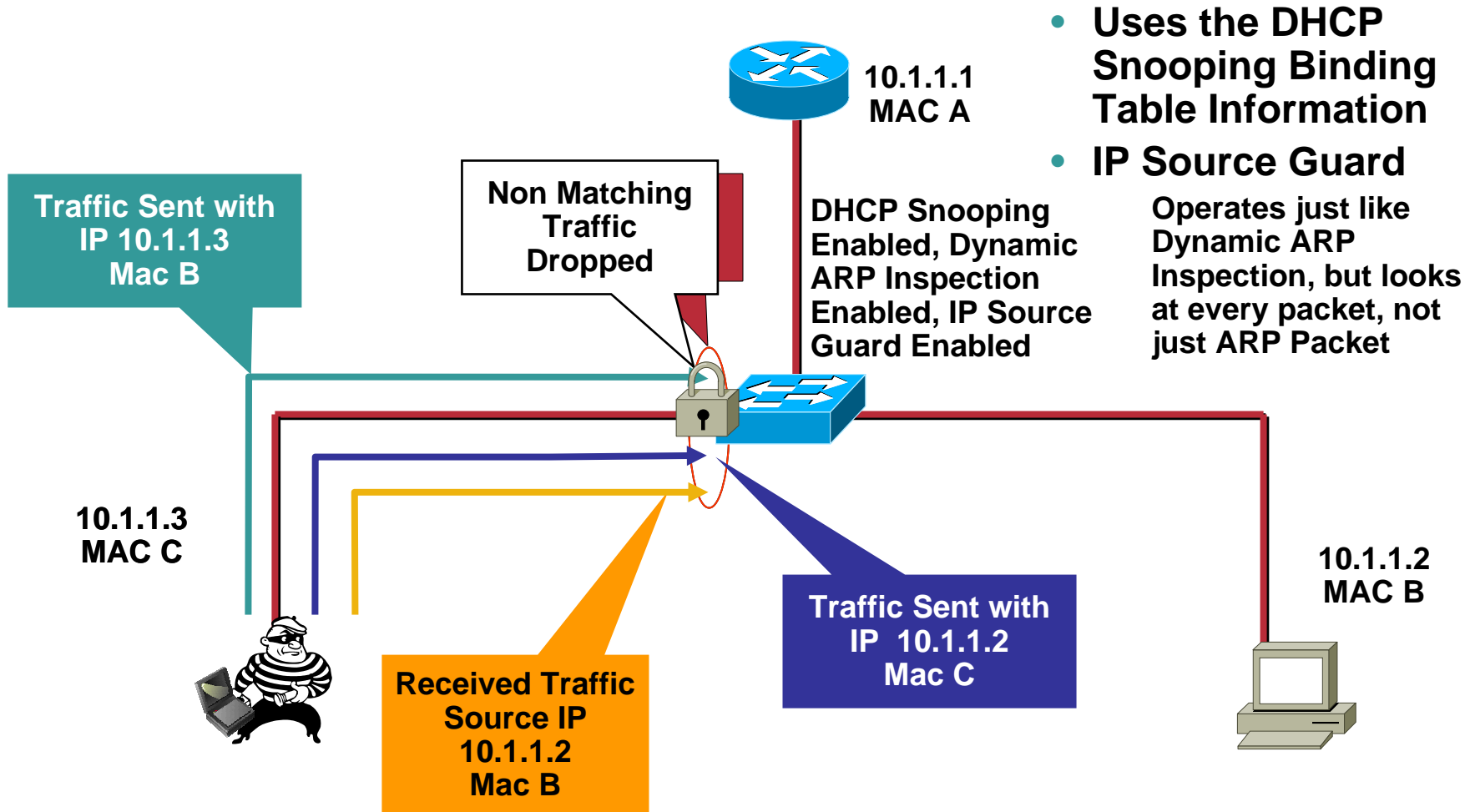


- Attacker sends packets with the incorrect source IP Address
- Whatever device the packet is sent to will never reply to the attacker

# Spoofing Attack: IP/MAC



# Countermeasures to Spoofing Attacks: IP Source Guard





# Countermeasures to Spoofing Attacks: IP Source Guard

- **Uses the information from the DHCP Snooping Binding table**

```
sh ip dhcp snooping binding
```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
00:03:47:B5:9F:AD	10.120.4.10	193185	dhcp-snooping	4	FastEthernet3/18
00:03:47:c4:6f:83	10.120.4.11	213454	dhcp-snooping	4	FastEthermet3/21

- **Looks at the MacAddress and IpAddress fields to see if the traffic from the interface is in the binding table, if not, traffic is blocked**

# Countermeasures to Spoofing Attacks: IP Source Guard

## Configuration of IP Source Guard

- DHCP Snooping had to be configured so the binding table it built
- IP Source Guard is configured by port
- IP Source Guard with MAC does not learn the MAC from the device connected to the switch, it learns it from the DHCP Offer
- MAC and IP checking can be turned on separately or together

For IP—

Will work with the information in the binding table

For MAC—

Must have an Option 82 enabled DHCP server  
(Microsoft does not support option 82)

Have to Change all router configuration to support Option 82

All Layer 3 devices between the DHCP request and the DHCP server  
will need to be configured to trust the Option 82 DHCP Request—`ip dhcp relay  
information trust`

**Note:** There are at least two DHCP servers that support Option 82 Field Cisco Network Registrar® and Avaya

# Countermeasures to Spoofing Attacks: IP Source Guard

## IP Source Guard

### IP Source Guard Configuration IP/MAC Checking Only (Opt 82)

#### *IOS*

#### *Global Commands*

```
ip dhcp snooping vlan 4,104  
ip dhcp snooping information option  
ip dhcp snooping
```

#### *Interface Commands*

```
ip verify source vlan dhcp-snooping  
port-security
```

### IP Source Guard Configuration IP Checking Only (no Opt 82)

#### *IOS*

#### *Global Commands*

```
ip dhcp snooping vlan 4,104  
no ip dhcp snooping information option  
ip dhcp snooping
```

#### *Interface Commands*

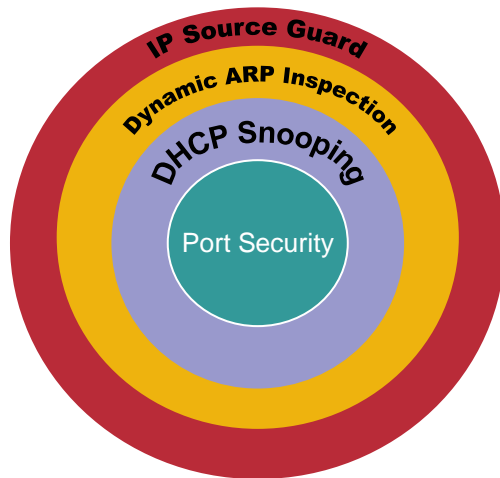
```
ip verify source vlan dhcp-snooping
```

# IP Source Guard vs. DAI

While IP Source Guard and Dynamic ARP Inspection offer similar functions, they are different in the way they operate and facilitate protection ...

Dynamic ARP Inspection	IP Source Guard
<ul style="list-style-type: none"><li>- DHCP Snooping creates IP to MAC bindings</li><li>- DAI Intercepts all ARP requests</li><li>- Intercepted ARP is validated against IP to MAC binding</li><li>- Does not switch ARP packets with invalid source address</li><li>- Used primarily to prevent MITM attacks</li></ul>	<ul style="list-style-type: none"><li>- Initially all traffic blocked</li><li>- Snoops DHCP Address</li><li>- Creates IP to MAC binding</li><li>- Installs per port VACL to deny traffic other than snooped source</li><li>- Protects against IP and MAC spoofing</li><li>- Will not prevent a MITM attack</li></ul>
Dynamic ARP Inspection	IP Source Guard

# Building the Layers

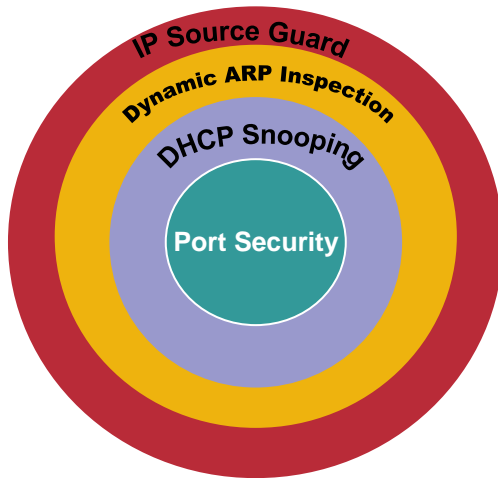


- **Port security prevents CAM attacks and DHCP Starvation attacks**
- **DHCP Snooping prevents Rogue DHCP Server attacks**
- **Dynamic ARP Inspection prevents current ARP attacks**
- **IP source guard prevents IP/MAC Spoofing**

# SUMMARY



# Building the Layers



- **Port Security prevents CAM attacks**
- **DHCP Snooping prevents Rogue DHCP Server attacks**
- **Dynamic ARP Inspection prevents current ARP attacks**
- **IP Source Guard prevents IP/MAC Spoofing**

# Best Practices for L2 Security

1. **Always use a dedicated VLAN ID for Trunk Ports**
2. **Disable unused ports and put them in an unused VLAN**
3. **Use Secure Transmission when managing Switches (SSH, OOB, Permit Lists)**
4. **Deploy Port Security**
5. **Set all host ports to Non Trunking (unless you are Cisco VoIP)**
6. **ALWAYS use a dedicated VLAN for Trunk Ports**
7. **Avoid using VLAN 1**
8. **Have a plan for ARP Security issues and implement it! (ARP Inspection, IDS, etc.)**
9. **Use SNMP V3 to secure SNMP transmission**
10. **Use STP Attack mitigation – Root Guard and BPDU Guard**
11. **Use CDP only where necessary**
12. **Use MD5 Authentication for VTP**
13. **Plan & implement DHCP Attack mitigation (DHCP Snooping, VACLs)**
14. **Use Private VLAN's to better secure guest VLAN's**
15. **Use and implement 802.1x (IBNS) to protect entry into your network**
16. **Consider using VACL's to limit access to key network resources...**



# CISCO SYSTEMS

