# LAYER 2 ATTACKS: MAC ADDRESS FLOODING, DHCP STARVATION, STP ATTACK

SONYA LAO
CCNP PER. 1/2

# Purpose

The purpose of this lab is to explore the vast number of possibilities to attack a network at the Layer 2 level. We were required to choose 3 attacks and understand how to execute the attack and how to prevent it from occurring again. To demonstrate our complete understanding of each attack, we were also required to create 3 videos detailing the steps required for execution and mitigation of each attack.

# Background Information

MAC address flooding: Every device has a permanent address, called the MAC address, or physical address. In a normal functioning network, when a PC requests information on another PC, the switch sends out a broadcast message on all ports except the one it originated from. Imagine you are in a grocery store with your daughter. While you are shopping, your child wanders off and you are unable to locate her. You go to the store manager and ask that he makes an announcement on the intercom asking for your child's location. Only she will know to respond to the intercom message, and the message can help facilitate your union. The same is true in a network. The switch is like the store manager; it facilitates the transfer of information. In a compromised network, the switch has no control over where information is directed. Imagine that instead of one parent looking for their child, there are suddenly one thousand new customers all with missing children, expecting the manager to make an announcement over the intercom. Because the manager cannot keep track of all the requests, he decides to announce everyone's requests and replies to everyone. Similarly, in a MAC address table flooding attack, the attacker sends thousands of fake MAC address requests to the switch to flood the table, putting the switch in fail-safe mode. In this mode, the switch acts like a hub, and sends all frames received to every port.

DHCP Starvation: If a device wants to connect to the Internet, it must have an IP address. Most PCs receive their addresses through Dynamic Host Configuration Protocol, which automatically assigns each end device an IP address from the network. Normal DHCP is like issuing school ID cards to each student in the school. As long as they attend Newport High School, the student will receive an ID number, which they can use to purchase lunch, borrow library books, and more. Every new student can also request an ID number. When the DHCP system is compromised, the router receives thousands of requests for IP addresses. When there are no more addresses in the network, the end devices that actually need the IP addresses are denied that service. Similarly, if the school administration receives thousands of requests for new student ID cards, they have no choice but to fulfill the order, without knowing that all the requests came from 1 student. When a new student registers, the school will no longer have any IDs for the student, since they were all assigned based on the requests of 1 student.

STP attack: When multiple switches are connected together, one switch is always elected as the root bridge, or the main switch to direct the traffic, through spanning tree. Spanning tree chooses the root bridge based on the lowest priority on each switch. When the network is compromised, an attacking software like Yersinia can act as the root bridge and significantly reduce network speeds as well as glean information.

# Lab Summary

MAC address flooding: First, I configured each PC with an IP address, all in the same network of 192.168.1.0/24. Then I verified that the switch had learned the MAC addresses of the PCs and VM by issuing the `show mac-address table` command. Next, in my Kali Linux VM, I issued the `macof -d 192.168.1.2 -n 1000` command in the terminal to send 1000 fake MAC address frames to the switch. To confirm that the MAC address table had been flooded, I re-issued the `show mac-address table` command and checked that packets were being sent in Wireshark. To mitigate the attack, I configured all ports as access ports and shutdown all unused ports. Then, I configured port security on the connected ports, setting the maximum number of addresses received on each port to 4 and setting the violation mode to shutdown so that if any port received more than 4 addresses, it would be shutdown. To check that my mitigation technique was valid, I re-issued the macof attack in my VM and checked to make sure the connected port was placed in Error Disable mode and shutdown.

DHCP Starvation: In order to execute DHCP Starvation, DHCP must be setup first. To do so, I issued the correct commands on R1, excluding 10 addresses from the network and making R1 the default router. Then, I assigned PC1 and my VM to get addresses via DHCP and used the `show ip dhcp binding` command to confirm that 2 addresses were leased. Next, in my Kali Linux VM, I typed `yersinia -G` in the terminal to open the graphical version of Yersinia. In Yersinia, I selected the DHCP tab, then launched the DHCP Discover attack. To verify that the attack was executed, I issued the `show ip dhcp binding` command again to see the number of fake addresses that were leased. Next, on PC 2, I tried to get an address via DHCP, and as expected, it failed. To mitigate the attack, I configured standard port security for all ports on the switch. Then, I set the port connected to PC 2 to be trusted by DHCP to ensure that it would receive an address. Next, I set up DHCP spoofing on VLAN 1 to verify the MAC addresses of each end device. To check if my mitigation was successful, I launched the attack in Yersinia again, and upon issuing the `show ip dhcp binding` command again, no new addresses were leased besides the addresses to PC 1, PC 2, and the VM.

STP attack: To set up the network, I connected three switches together with 2 PCs on either end of the network. I decided to make Switch 3 the root bridge, so I configured a priority of 0 on that switch. Then, I configured Switch 2 with a priority of 4096, which is second highest, and Switch 1 with a priority of 8192. To verify that the spanning tree network had converged and was using S3 as the root bridge, I issued the `show spanning-tree` command. Next, in Kali Linux, I used the graphical version of Yersinia to launch the attack. I selected the STP tab, and launched the "Claiming Root Role" attack. To verify that the attack had been successful, I re-issued the `show spanning-tree` command on S3 to show that it was no longer the root bridge. To mitigate the attack, I configured all ports on S3 to be prevented from becoming root ports and enabled BPDU guard so that the spanning tree domain borders would remain the same. To verify that the mitigation was successful, I launched the attack again in Yersinia, and port 2 was put in error-disable state.

# Lab Commands

Mac Address Flooding:

Attack:

```
Macof –d [destination IP address] –n [number of frames to send]
```

This command is issued in the Kali Linux terminal to launch a series of fake MAC address frames to the switch. The –d refers to the destination of the attack and the –n refers to the number of fake frames to send to the switch.

Mitigation:

```
S1(config-if)#switchport port-security maximum [number]
```

This command is configured on the interface of the switch. It sets the maximum number of secure MAC addresses on a port. The default number of addresses is 1.

```
S1(config-if)#switchport port-security violation
[protect/restrict/shutdown]
```

This command sets the violation mode, informing the switch how to respond when there is a security violation. There are three modes, protect, restrict, and shutdown. Protect mode simply drops the unknown source address packets, Restrict mode drops the packets and increments the Security Violation counter, and Shutdown mode places the port in error-disable state.

DHCP Starvation:

Pre-attack:

```
R1(config)#ip dhcp excluded-address [ip address range]
```

This command allows one to exclude a range of addresses from the DHCP pool.

```
R1(config)#ip dhcp pool name [name]
```

This command creates a name for the DHCP pool and allows one to enter the DHCP configuration mode.

```
R1(dhcp-config)#network [address] [subnet mask]
```

In DHCP configuration mode, this command specifies the network and the subnet mask to be used in the DHCP address pool.

```
R1(dhcp-config)#default-router [ip address]
```

Through this command, the desired router is configured as the default gateway for all end devices that will be leased addresses through DHCP.

Verification:

```
R1#show ip dhcp binding
```

Command used to display the addresses currently leased by DHCP. Contains information on the address leased, MAC address, Expiration date, and type of lease.

Mitigation:

```
S1(config-if)#ip dhcp snooping trust
```

This command configures the desired interface as a trusted source for DHCP assignment and makes sure that the traffic from the desired interface is not filtered out.

```
S1(config)#ip dhcp snooping vlan [number]
```

This command enables DHCP snooping on the configured VLAN to screen between trusted sources and untrusted hosts.

```
S1(config)#ip dhcp snooping verify mac-address
```

By enabling mac address verification, if the switch receives a packet on an unfamiliar interface, and the source MAC address and DHCP hardware address are not the same, the packet will be dropped.

STP MiTM:

Pre-attack:

```
S1(config)#spanning-tree vlan [number] priority [multiple of 4096]
```

This command configures the spanning tree priority of the switch on the desired VLAN. The priority is the first consideration in determining the root bridge. The number must be a multiple of 4096.

Verification:

```
S1#show spanning-tree
```

Used to display the spanning tree network, with information on the root bridge of the network and the states of each port in the network (blocking or forwarding). It also displays the type of each port on the switch (designated, root, blocked, alternate)

Mitigation:

```
S1(config-if)# spanning-tree guard root
```

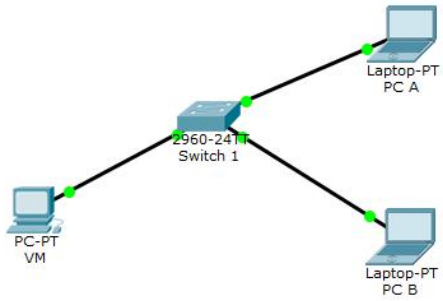
Through this command, the interface is protected from becoming a root port. This command is used on ports that are known to be designated ports. If the port receives BPDUs that are higher, the port will be moved to root-inconsistent state, and no traffic is forwarded from that port.

```
S1(config-if)# spanning-tree bpduguard enable
```

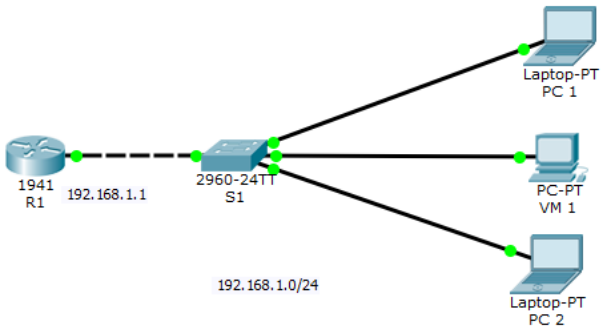
With BPDU guard enabled, if the port receives a BPDU, the port will be disabled and placed in error-disabled state.

# Network Diagram

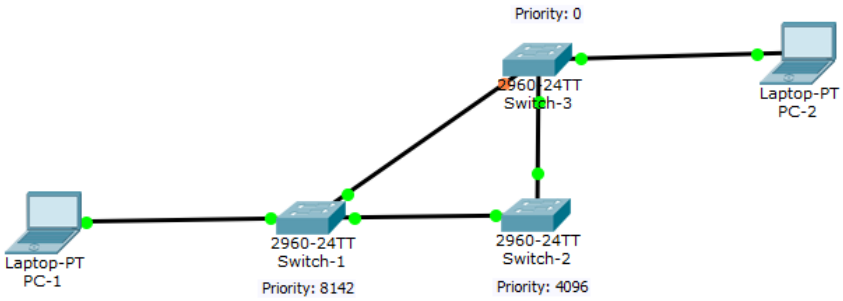
MAC address table flooding:




DHCP Starvation:




Spanning Tree Attack:



# Configurations

MAC Address Flooding:

**S1 show mac address table Pre-Attack:**

```
S1#show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
All    0100.0ccc.cccc    STATIC      CPU
All    0100.0ccc.cccd    STATIC      CPU
All    0180.c200.0000    STATIC      CPU
All    0180.c200.0001    STATIC      CPU
All    0180.c200.0002    STATIC      CPU
All    0180.c200.0003    STATIC      CPU
All    0180.c200.0004    STATIC      CPU
All    0180.c200.0005    STATIC      CPU
All    0180.c200.0006    STATIC      CPU
All    0180.c200.0007    STATIC      CPU
All    0180.c200.0008    STATIC      CPU
All    0180.c200.0009    STATIC      CPU
All    0180.c200.000a    STATIC      CPU
All    0180.c200.000b    STATIC      CPU
All    0180.c200.000c    STATIC      CPU
All    0180.c200.000d    STATIC      CPU
All    0180.c200.000e    STATIC      CPU
All    0180.c200.000f    STATIC      CPU
All    0180.c200.0010    STATIC      CPU
All    ffff.ffff.ffff    STATIC      CPU
   1    000a.b82d.10e0    DYNAMIC     Fa0/16
   1    0012.80b6.4cd8    DYNAMIC     Fa0/3
Total Mac Addresses for this criterion: 22
```

**S1 show mac address table Post-Attack:**

```
S1#show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
All    0100.0ccc.cccc    STATIC      CPU
All    0100.0ccc.cccd    STATIC      CPU
All    0180.c200.0000    STATIC      CPU
All    0180.c200.0001    STATIC      CPU
All    0180.c200.0002    STATIC      CPU
All    0180.c200.0003    STATIC      CPU
All    0180.c200.0004    STATIC      CPU
All    0180.c200.0005    STATIC      CPU
All    0180.c200.0006    STATIC      CPU
All    0180.c200.0007    STATIC      CPU
All    0180.c200.0008    STATIC      CPU
All    0180.c200.0009    STATIC      CPU
All    0180.c200.000a    STATIC      CPU
All    0180.c200.000b    STATIC      CPU
All    0180.c200.000c    STATIC      CPU
All    0180.c200.000d    STATIC      CPU
All    0180.c200.000e    STATIC      CPU
All    0180.c200.000f    STATIC      CPU
All    0180.c200.0010    STATIC      CPU
All    ffff.ffff.ffff    STATIC      CPU
   1    000a.b82d.10e0    DYNAMIC     Fa0/16
   1    0012.80b6.4cd8    DYNAMIC     Fa0/3
   1    0012.80b6.4cd9    DYNAMIC     Fa0/16
   1    0014.6915.4100    DYNAMIC     Fa0/16
   1    0018.b921.9200    DYNAMIC     Fa0/16
   1    0018.b921.9278    DYNAMIC     Fa0/1
   1    0018.b974.528f    DYNAMIC     Fa0/16
   1    0019.0617.660f    DYNAMIC     Fa0/13
   1    0019.0617.6610    DYNAMIC     Fa0/14
   1    0019.0617.6611    DYNAMIC     Fa0/15
   1    001b.d450.970f    DYNAMIC     Fa0/19
   1    001b.d450.9710    DYNAMIC     Fa0/20
   1    001b.d450.9711    DYNAMIC     Fa0/21
   4    0018.b974.528f    DYNAMIC     Fa0/16
  56    0018.b974.528f    DYNAMIC     Fa0/16
  56    0019.069c.80e1    DYNAMIC     Fa0/19
   6    0018.b974.528f    DYNAMIC     Fa0/16
```

```
  45    0018.b945.f780    DYNAMIC     Fa0/5
  45    0018.b974.528f    DYNAMIC     Fa0/16
  56    0018.b945.f781    DYNAMIC     Fa0/16
  56    0018.b974.528f    DYNAMIC     Fa0/16
  56    0019.069c.80e1    DYNAMIC     Fa0/19
   6    0018.b974.528f    DYNAMIC     Fa0/16
   6    0019.069c.80e0    DYNAMIC     Fa0/13
Total Mac Addresses for this criterion: 42
```

**S1 show run:**

```
enable
configure terminal
hostname S1
no ip domain lookup
interface range f1/0/4 – 24
switchport mode access
shutdown
interface range f1/0/1 – 3
switchport mode access
switchport port-security
switchport port-security maximum 4
switchport port-security violation shutdown
switchport port-security mac-address sticky
```

DHCP Starvation Attack:

**R1 show ip dhcp binding Pre-Attack:**

```
R1(config)#do sh ip dhcp binding
IP address  Client-ID/  Lease expiration Type
            Hardware address
192.168.1.11 00D0.BCA8.2BEC Nov 17 2016 05:09 PM Automatic
192.168.1.12 0001.630D.B1D0 Nov 17 2016 05:11 PM Automatic
```

**R1 show ip dhcp binding Post-Attack:**

```
R1(config)#do sh ip dhcp binding
IP address  Client-ID/  Lease expiration Type
            Hardware address
192.168.1.11 00D0.BCA8.2BEC Nov 17 2016 05:09 PM Automatic
192.168.1.12 0001.630D.B1D0 Nov 17 2016 05:11 PM Automatic
192.168.1.13 D500.9C9B.F6E4 Nov 18 2016 05:12 PM Automatic
192.168.1.14 F7E6.02B3.AFD9 Nov 18 2016 05:12 PM Automatic
192.168.1.15 4C06.C111.448B Nov 18 2016 05:12 PM Automatic
192.168.1.16 6F23.FF11.C704 Nov 18 2016 05:12 PM Automatic
192.168.1.17 F5A3.D690.DBCE Nov 18 2016 05:12 PM Automatic
192.168.1.18 8367.9A9E.DBE8 Nov 18 2016 05:12 PM Automatic
```

```
192.168.1.19 2B7E.C2C9.9D03 Nov 18 2016 05:12 PM Automatic
192.168.1.20 0729.3B92.4F4C Nov 18 2016 05:12 PM Automatic
192.168.1.21 4011.4695.7FD7 Nov 18 2016 05:12 PM Automatic
192.168.1.22 4E57.B03B.B7B8 Nov 18 2016 05:12 PM Automatic
192.168.1.23 9252.6D71.2EAE Nov 18 2016 05:12 PM Automatic
192.168.1.24 42E4.E798.D4A2 Nov 18 2016 05:12 PM Automatic
192.168.1.25 5596.ECB7.53C5 Nov 18 2016 05:12 PM Automatic
192.168.1.26 2128.7CB1.49D1 Nov 18 2016 05:12 PM Automatic
192.168.1.27 42E4.E798.D4A2 Nov 18 2016 05:12 PM Automatic
192.168.1.28 4E07.8739.A5A0 Nov 18 2016 05:12 PM Automatic
192.168.1.29 4C0F.1F28.AB6E Nov 18 2016 05:12 PM Automatic
192.168.1.30 C88F.E7D1.CE0D Nov 18 2016 05:12 PM Automatic
192.168.1.31 CEA8.09DE.A369 Nov 18 2016 05:12 PM Automatic
192.168.1.32 570C.A372.5B71 Nov 18 2016 05:12 PM Automatic
192.168.1.33 6FC2.74A2.10A9 Nov 18 2016 05:12 PM Automatic
192.168.1.34 AA4B.0580.691A Nov 18 2016 05:12 PM Automatic
192.168.1.35 9DF7.E1FC.7C82 Nov 18 2016 05:12 PM Automatic
192.168.1.36 780F.1840.D58A Nov 18 2016 05:12 PM Automatic
192.168.1.37 42E4.E798.D4A2 Nov 18 2016 05:12 PM Automatic
192.168.1.38 B5A4.BD80.BC12 Nov 18 2016 05:12 PM Automatic
192.168.1.39 B808.51C0.910C Nov 18 2016 05:12 PM Automatic
192.168.1.40 9C70.6522.818C Nov 18 2016 05:12 PM Automatic
192.168.1.41 109B.6154.F19F Nov 18 2016 05:12 PM Automatic
192.168.1.42 230C.2DB4.25C1 Nov 18 2016 05:12 PM Automatic
192.168.1.43 2424.4B14.F79D Nov 18 2016 05:12 PM Automatic
192.168.1.44 E614.BD98.E028 Nov 18 2016 05:12 PM Automatic
192.168.1.45 E3F1.B605.B598 Nov 18 2016 05:12 PM Automatic
```

STP Attack:

**S3 show spanning tree Pre-Attack:**

```
S3(config)#do sh spanning-tree
VLAN0001
Spanning tree enabled protocol ieee
Root ID Priority 1
Address 0030.F2B6.4804
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 1 (priority 0 sys-id-ext 1)
Address 0030.F2B6.4804
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 20

Interface Role Sts Cost Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/2 Desg FWD 19 128.2 P2p
Fa0/1 Desg FWD 19 128.1 P2p
```

```
Fa0/3 Desg FWD 19 128.3 P2p
```

**S2 show spanning tree Pre-Attack:**

```
S2(config)#do sh spanning-tree
VLAN0001
Spanning tree enabled protocol ieee
Root ID Priority 1
Address 0030.F2B6.4804
Cost 19
Port 2(FastEthernet0/2)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 4097 (priority 4096 sys-id-ext 1)
Address 0002.1680.B5DC
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 20

Interface Role Sts Cost Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/1 Desg FWD 19 128.1 P2p
Fa0/2 Root FWD 19 128.2 P2p
```

**S1 show spanning tree Pre-Attack:**

```
S1(config)#do sh spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    1
             Address     0030.F2B6.4804
             Cost        19
             Port        2(FastEthernet0/2)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    8193  (priority 8192 sys-id-ext 1)
             Address     0030.F235.2880
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/2            Root FWD 19        128.2    P2p
Fa0/3            Desg FWD 19        128.3    P2p
Fa0/1            Altn BLK 19        128.1    P2p
```

# Problems

In implementing the attacks and mitigation, I did not encounter many problems.

Learning to use the Yersinia command line Linux took time. I was not used to the command-line style of operation, and would often confuse the keyboard shortcuts when using the software. But, I discovered a graphical version of Yersinia that was a lot more user-friendly, so I switched to use that version instead of the command line based version.

In addition, when I was creating my videos, I had filmed almost all of my DHCP Starvation attack video on my second hard drive. When I came into class one day, the hard drive was broken and had issues loading an operating system. As a result, I had to spend time re-filming and re-narrating the video.

# Conclusion

Through this lab, I've learned that there are many ways to attack a network on the layer 2 level, but Cisco has also developed robust solutions to combat these attacks. By taking the mindset of both the hacker and the network administrator helped me to better understand the nature of networking.