

Protecting your Network from Internal Attacks

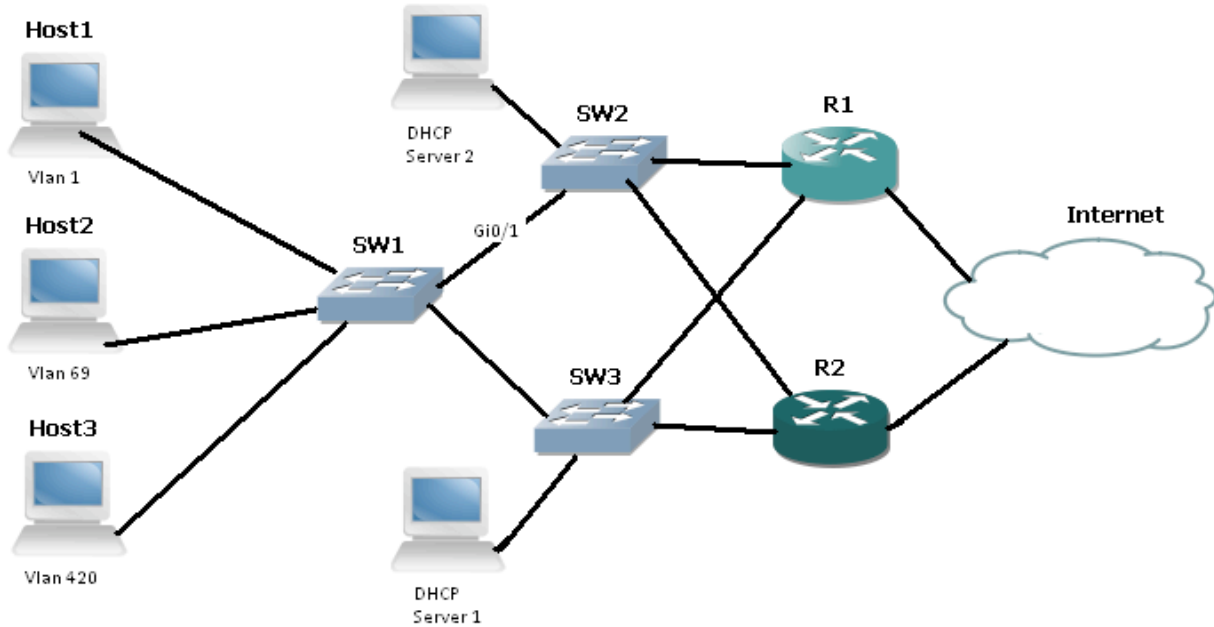
In this article I am going to give you some tips on how to stop someone from exploiting vulnerabilities in your network by turning on some simple security features in your switches. Most people don't know that the switches in their network can give a solid defense against several attack vectors. This article will focus on how just a few lines of configuration can harden your network to prevent the following attacks:

1. DHCP starvation
2. Rogue DHCP servers
3. Client side exploitation
4. ARP spoofing
5. ARP poison routing
6. VLAN hopping
7. MAC address flooding
8. Connection of Rouge devices

This article illustrates configuration on products from Cisco Systems, which are most likely already in use in your network. If you aren't using Cisco products consult your vendor's literature to see if the equivalent commands/protections are available.

Before trying this in a live environment make sure you test your configuration!

Figure 1 shows the topology that exists for the examples I will show.



Figure

Hiding end Hosts with Port Protection

If an attacker plugs into your network and runs a port scanner like NMAP they usually will get a response from a set of network devices. If the attacker doesn't get a response then they might think there isn't anything on that address to attack. Essentially, if you limit an attacker's view of the hosts on your network then you increase security.

From a network security standpoint we only want end hosts to be able to see certain things. Every host in an office doesn't need to see every other host. They only need to see the Servers where they save their data, the gateway for their subnet and possibly a network printer. We are going to limit what hosts can talk to other hosts by using a feature of Cisco switches called port protection. Port Protection stops the host from being able to view any other host that is in a port-protected state. If you turn on Port Protection for all of your end user stations then they can't communicate with each other directly. Which means one host can't attack the other.

To start the process we need to SSH into SW1 (You are using SSH right?). We log into the switch and type the command **Show Interface Status** which gives us a view like the following.

```
SW1(config)#do sh int status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/1		notconnect	1	auto	auto	10/100BaseTX
Fa0/2		connected	420	a-full	a-100	10/100BaseTX
Fa0/3	VOIP	connected	1	a-full	a-100	10/100BaseTX

Fa0/4	VOIP	connected	1	a-full	a-100	10/100BaseTX	
Fa0/5		notconnect	1	auto	auto	10/100BaseTX	
Fa0/6		connected	1	a-full	a-100	10/100BaseTX	
Fa0/7	VOIP	connected	1	a-full	a-100	10/100BaseTX	
Fa0/8		notconnect	1	auto	auto	10/100BaseTX	
Fa0/9		connected	420	a-full	a-100	10/100BaseTX	
Fa0/10	VOIP	connected	1	a-full	a-100	10/100BaseTX	
Fa0/11		connected	1	a-half	a-10	10/100BaseTX	
Fa0/12		connected	420	a-full	a-100	10/100BaseTX	
Fa0/13		connected	420	a-full	a-100	10/100BaseTX	
Fa0/14		connected	1	a-full	a-100	10/100BaseTX	Fa0/15
VOIP	connected	1	a-full	a-100	10/100BaseTX		
Fa0/16		connected	420	a-full	a-100	10/100BaseTX	
Fa0/17		notconnect	1	auto	auto	10/100BaseTX	
Fa0/18		connected	420	a-full	a-100	10/100BaseTX	
Fa0/19		connected	420	a-full	a-100	10/100BaseTX	
Fa0/20		connected	420	a-full	a-100	10/100BaseTX	
Fa0/21		connected	420	a-full	a-100	10/100BaseTX	
Fa0/22	WAP	connected	69	a-full	a-100	10/100BaseTX	
Fa0/23		connected	420	a-full	a-100	10/100BaseTX	
Fa0/24		notconnect	420	auto	auto	10/100BaseTX	
Gi0/1		connected	trunk	a-full	a-1000	1000BaseSX SFP	
Gi0/2		connected	trunk	a-full	a-1000	1000BaseSX SFP	

Now we look for the ports that are listed as trunk. These are the ports that we don't want to change as they are uplink ports to another switch. Every other port is fair game which means we can configure ports 1 through 24. Here are the commands to turn on Port Protection and secure the ports for interfaces 1 through 24. If you aren't in global configuration mode at this point type **configure terminal**.

```
SW1(config)#interface range f0/1-24      ← This command selects multiple ports
SW1(config-if-range)#switchport host    ← This command runs the host macro
SW1(config-if-range)#switchport protected ← This command turns on port protection
SW1(config-if-range)#switchport nonegotiate ← This command prevents it from trunk negotiation
```

With the above commands entered we have 24 ports that can't talk to each other, can't auto negotiate a trunk and are setup to be end user hosts. This prevents one end client from attacking another end client or being used as a jump off point to attack another client.

How to Stop a MAC Flooding Attack

As a Penetration tester if I feel that I am being limited by a switch I am going to attack that switch to try to get more information out of it than it would normally give me. One of the most common techniques to attack a switch is MAC Address flooding. You can use the open source tool macof to flood a switch and make it act like a hub. Once the switches CAM table is flooded it will try to continue to send traffic by flooding all traffic out of all ports. This allows an attacker to then sniff all traffic thereby gaining information about the network. The attacker can then use a packet sniffer and see unencrypted protocols.

We will use a security technique called Port Security to stop a Mac Flood. Port Security allows a certain number of MAC addresses to be registered to a port and then it will then take an action on the port. If the switch sees more than the number we set then it can stop the traffic, protect the port or shutdown the port. This can stop MAC flooding attacks dead in their tracks by literally blocking the traffic. We can set this up by using just a few commands. On the same switch we were looking at previously we will issue the following commands.

```
SW1(config)#interface range f0/1-24          ← The command to select multiple ports
SW1(config-if-range)#switchport port-security maximum 1    ← Sets the maximum # of mac to 1
SW1(config-if-range)#switchport port-security violation shutdown ← Shuts down the port
SW1(config-if-range)#switchport port-security mac-address sticky ← Saves the mac addresses
SW1(config-if-range)#switchport port-security          ← enables port security
```

Once these commands have been issued the switch will keep track of the MAC addresses that it sees on each port. If the number grows too large (more than one in this case) it will shut down the port. If a MAC flooding attack occurs then it will just shutdown the port. If the end user connects a switch like a WRT54G or a home router to the port they will get shut out as well. It will also send a log message when a violation occurs. If you look at your logs you will see any violations that happen (You are looking at your logs right?).

Defending against DHCP attacks

To defend our network against DHCP attacks we will use a technique called DHCP snooping. DHCP snooping allows the switch to snoop into the DHCP transaction allowing it to take actions to prevent attacks. DHCP snooping only allows the DHCP transaction to come from a port that it trusts. This means if an end user attempts a DHCP starvation attack, Rogue DHCP server attack, or attempts to push DHCP options from a port the switch will stop it from occurring.

To turn on DHCP snooping we use the following commands.

```
SW1(config)#ip dhcp snooping vlan 1,69,420          ← This is what VLANs to snoop
SW1(config)#no ip dhcp snooping information option    ← This allows some DHCP options
SW1(config)#no ip dhcp snooping verify mac-address    ← This prevents MAC verification
SW1(config)#no ip dhcp snooping verify no-relay-agent-address ← This prevent it from relaying
SW1(config)#ip dhcp snooping                        ← This turns on DHCP snooping
```

Now it's important to know where your DHCP server is so you can trust the port that will be sending the traffic to the Server. We will need to trust our trunk ports or no one will be able to get addresses. To do this we enter the following commands.

```
SW1(config)#interface range g0/1-2          ←- Command to configure multiple interfaces
SW1(config-if)#ip dhcp snooping trust      ←- Command to trust a port for DHCP
```

To prevent DHCP starvation attacks we want to limit the rate of DHCP requests on untrusted ports. The following commands limit the rate which DHCP packets can be sent across a port.

```
SW1(config)#interface range f0/1-24      <- Command to configure multiple interfaces
SW1(config)#ip dhcp snooping limit rate 100 <- Command to limit the rate of DHCP packets
```

Now if an attacker attempts to starve our DHCP server they will only be able to get 100 DHCP packets per second across the interface. Some tuning is necessary for using this command. Make sure that the number of packets per second is correct for your network or some hosts won't receive addresses.

Once this is setup you can verify it works by using the following command.

```
SW1(config)#do show ip dhcp snooping binding
```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
00:1E:CA:F9:C6:8C	10.2.101.152	356245	dhcp-snooping	69	FastEthernet0/15
00:26:54:13:67:BA	10.112.83.33	597451	dhcp-snooping	420	FastEthernet0/23
00:1E:CA:F9:BE:F6	10.2.101.156	356693	dhcp-snooping	1	FastEthernet0/4
00:1E:4F:51:57:63	10.112.83.136	536885	dhcp-snooping	69	FastEthernet0/13
00:1E:CA:F9:C6:77	10.2.101.154	356377	dhcp-snooping	420	FastEthernet0/7
00:0B:DB:7D:06:6C	10.112.83.138	674790	dhcp-snooping	420	FastEthernet0/21
00:21:70:1F:73:86	10.112.83.144	511592	dhcp-snooping	69	FastEthernet0/19
00:1E:CA:F9:C6:95	10.2.101.155	356446	dhcp-snooping	69	FastEthernet0/3
00:08:74:D4:A9:C6	10.112.83.125	513299	dhcp-snooping	420	FastEthernet0/16
00:17:5A:7B:7E:18	10.69.83.11	347349	dhcp-snooping	69	FastEthernet0/22
00:1E:CA:F9:C6:8B	10.2.101.157	431052	dhcp-snooping	420	FastEthernet0/10
00:24:E8:02:34:D5	10.112.83.143	510274	dhcp-snooping	420	FastEthernet0/2

Total number of bindings: 12

Protecting from ARP attacks

ARP spoofing is sending spoofed ARP messages to a Switch. Generally, the aim is to associate the attacker's MAC with the IP address of another host like the gateway for the subnet. This type of attack is usually an attempt to execute a Man in the Middle attack making the attacking host become the gateway for the subnet. The attacker then tries to intercept any traffic and then route it on to the actual gateway if necessary. This technique is called ARP poisoning and tools like Cain & Able and Ettercap can specifically perform this type of attack.

To protect from ARP spoofing attacks we need to have completed the previous DHCP snooping configuration. (This can be done without DHCP snooping but involves hard coding the addresses) The switch will then use the DHCP snooping table it created to prevent ARP spoofing attacks. The switch will know what MAC is assigned to what address on each port allowing it to make a decision whether an ARP is valid or not.

To protect from ARP spoofing we enter the following commands

```
SW1(config)#ip arp inspection vlan 1,69,420      <- This tells it what VLANs to check
SW1(config)#interface range g0/1-2             <- Command to configure multiple interfaces
SW1(config-if-range)#ip arp inspection trust    <- This command trusts the ARPs
```

The last two commands trust any ARP from the trusted port. That is very important for ports connected to other switches as it can break the topology.

In conclusion the commands I have shown can seriously decrease the Attack surface for an internal attack. This setup also increases the administrators knowledge of what is going on with the network by showing what IP is connected to what port and the MAC address to which it belongs.

If everyone used all the security features of the equipment they already have it would make a Penetration testers life much harder. In our examples we literally entered less than 20 commands and protected ourselves from at least 8 or more common attack vectors.

One feature I didn't cover is Private VLANs this configuration is extremely complex but works well in an ISP type environment. If you would like more information on this here is a link from the Cisco website showing how to set up Private VLANs.

http://www.cisco.com/en/US/tech/tk389/tk814/technologies_configuration_example09186a008017acad.shtml

by (infosec Samurai) Brian Almond