**Switchport Security**

Table of Contents

[Setup 3](#_Toc87293925)

[Learning and Limiting MAC Addresses 3](#_Toc87293926)

[Port Security Aging 5](#_Toc87293927)

[Port Security Violations 5](#_Toc87293928)

[Verification 6](#_Toc87293929)

Note: The ports we are discussing here refer to the physical connection ports on layer 2 switches, not the application layer ports.

If an attacker manages to gain **physical access** to the devices in an organization, it becomes almost trivial to do serious damage. One of the common ways in which physical access is gained is via the **ports** on network switches. This is because, by default, all ports on a switch are **open**.

To prevent such attacks, we can do two things. First, we can **shutdown** any interfaces that are not being used via the shutdown command. Second, we can enable **security features** on the interfaces that are being used to make sure only specific devices that are authorized can communicate via those interfaces. Since switches work on layer 2, the security features will identify devices using **MAC addresses**. This second solution is what we are referring to when we say **switchport security**.

## Setup

By default, an interface will be set to **dynamic mode**, which means it can work either in access mode or trunk mode as required. Recall from the lecture on VLAN that **access mode** allows the interface to access a **single VLAN** while **trunk mode** allows the interface to access **multiple VLANs**. To enable switchport security on a particular interface, we must first set the port to either **access or trunk mode**.

S1(config-if)# switchport mode access  
S1(config-if)# switchport mode trunk

CLI

After this, we can **enable** switchport security using the following command:

S1(config-if)# switchport port-security

CLI

Once switchport security is enabled, we have access to a few more commands that give us control over how the security will work. We will now be going over each of these in turn.

### Learning and Limiting MAC Addresses

We can **limit** the number of MAC addresses that can be connected to the port.

S1(config-if)# switchport port-security maximum 1

CLI

We can also configure how the MAC addresses that are valid are to be **learnt** by the switch. This can be done in one of three ways:

1. Manual Configuration
2. Dynamic Learning
3. Dynamic Learning – Sticky

For **manual configuration**, we must manually enter each of the valid MAC addresses.

S1(config-if)# switchport port-security mac-address 00:00:5E:00:53:AF

CLI

For **dynamic learning**, the valid MAC addresses are learnt **automatically** by the switch when we connect a device to the port. This is the default. For example, if we set the maximum number of MAC addresses to 1 and then connected a device, that device will be the only one that can now connect to that interface. To be able to connect a different device, we must **restart** the switch, at which point it will **forget** the previously learnt MAC addresses.

If we want the switch to **not forget** the MAC addresses it dynamically learnt upon restart, we can use **dynamic learning – sticky**.

S1(config-if)# switchport port-security mac-address sticky

CLI

This causes the MAC addresses to be added to the **running configuration**. If we **save** the running configuration, the MAC addresses will be added to the **NVRAM**.

### Port Security Aging

We can set the MAC addresses that are allowed to **expire** after a certain amount of time in minutes. Setting the value to 0 will cause the device to **never expire**.

S1(config-if)# switchport port-security aging 60

CLI

We can also configure this time so that it is either **absolute**, meaning the addresses will expire exactly at the specified time, or **inactivity**, meaning the addresses will expire if there is no activity from the device for the specified time.

S1(config-if)# switchport port-security aging type absolute

S1(config-if)# switchport port-security aging type inactivity

CLI

However, note that the aging type cannot be set using Cisco Packet Tracer.

### Port Security Violations

Finally, we can configure what **action** should be taken if the there is a **violation**, for example if more than the allowed MAC addresses are connected or a MAC address that is not in the list is connected. The action can be of one of three types:

1. **Protect** – Drop all traffic from the offending device.
2. **Restrict** – Drop all traffic from the offending device, log the violation and increment the violation counter.
3. **Shutdown** – Send the switchport into error disabled mode, where no further traffic is forwarded. This is the default mode. It can be reset by disabling and re-enabling the switchport. The violation is also logged and the violation counter is incremented.

## Verification

We can check the **device-wide port security** status.

S1# show port-security

CLI

We can check the port security status of a **particular interface**.

S1# show port-security interface fastethernet 0/1

CLI

We can check which **addresses** are allowed at all interfaces of the device.

S1# show port-security address

CLI