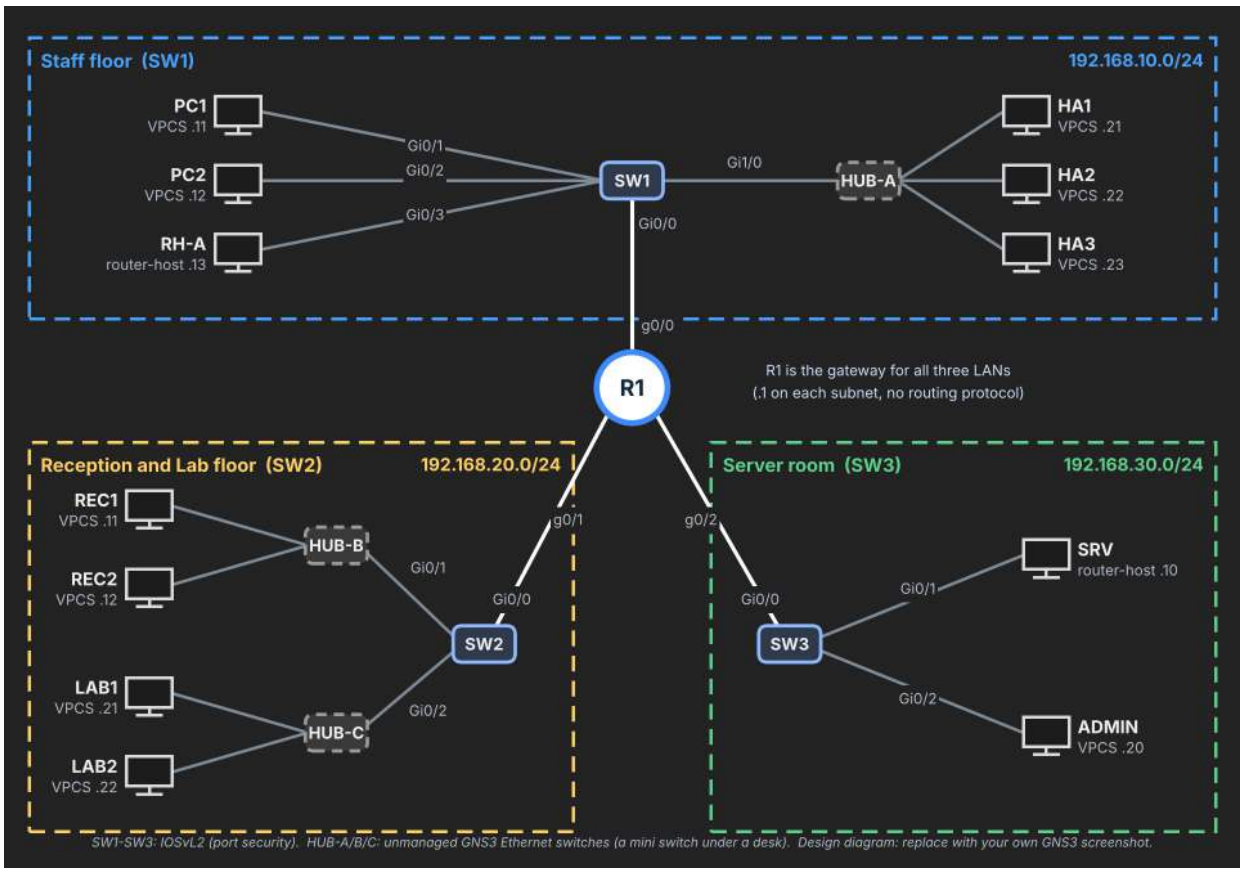
Port Security Mega Lab Book

Secure MAC addresses, violation modes, aging and sticky MACs: theory first, then hands-on GNS3 labs

Abdul Munim — CCNA 200-301 Prep



Lab topology: one gateway router R1 and three switch floors (Staff, Reception and Lab, Server room), with small switches HUB-A, HUB-B and HUB-C under desks

Contents

Preface

Part I — Theory

Chapter 1 — Why port security exists
Chapter 2 — Enabling port security
Chapter 3 — Reading show port-security interface
Chapter 4 — When the port shuts down, and bringing it back
Chapter 5 — ErrDisable recovery
Chapter 6 — Violation modes
Chapter 7 — Secure MAC address aging
Chapter 8 — Sticky secure MAC addresses
Chapter 9 — Secure MACs in the MAC address table
Chapter 10 — Choosing settings, and a checklist

Part II — Labs

Chapter 11 — The topology, the addresses and the company rulebook
Chapter 12 — Base configuration and baseline test
Chapter 13 — Lab 1: Enabling port security
Chapter 14 — Lab 2: A violation, and re-enabling by hand
Chapter 15 — Lab 3: ErrDisable recovery
Chapter 16 — Lab 4: Shutdown, restrict and protect side by side
Chapter 17 — Lab 5: Secure MAC address aging
Chapter 18 — Lab 6: Sticky secure MAC addresses
Chapter 19 — Lab 7 (extra): Static MACs, the MAC table and spoofing
Chapter 20 — Reset steps and GNS3 canvas labeling
Chapter 21 — Master configuration and test matrix
Chapter 22 — Break-it-then-fix-it drills
Chapter 23 — Quick reference

Part III — Beyond the videos

Chapter 24 — What comes after port security

Appendices

Appendix A — Glossary

Preface

Why port security feels harder than it is

Port security is one rule: **this switch port may only be used by these MAC addresses.** Five things make it feel hard.

1. **Many options, one feature.** Maximum count, static, dynamic, sticky, aging, three violation modes. They are small pieces. Chapter 2 puts them on one map.
2. **The word "secure" is everywhere.** Secure MAC address, secure port, secure-shutdown. Chapter 3 shows what each one means on screen.
3. **A violation can mean different things.** In one mode the port dies. In another it stays up and complains. In the third it stays up and says nothing. Chapter 6 compares them side by side.
4. **Dynamic MACs disappear.** If the port goes down, they are forgotten. That one fact explains most "why did the wrong device get in?" questions.
5. **It is not authentication.** The switch only reads the source MAC in a frame. A MAC address can be copied. Chapter 9 and Lab 7 show this.

The course lecture and where it lives in this book

I cannot watch videos. The order below comes from the published chapter list of the lecture, not from watching it. If the video covers a point in a different order, trust the video and use this book as a second explanation.

I could **not** confirm a separate lab video for this topic. So the labs in Part II are my own design. They practice every point of the lecture on a bigger network than a single switch.

Lecture chapter (in order)	Chapters here
Port security intro; why port security?	1
Enabling port security	2
`show port-security interface`	3
Manually re-enable a disabled interface	4
ErrDisable recovery	5
Violation modes: shutdown, restrict, protect	6
Secure MAC address aging	7
Sticky secure MAC addresses	8
MAC address table	9
Command review	10, 23
Not in the lecture (foundation I added)	MAC learning basics in 1; the checklist in 10
Not in the lecture (extra labs)	Lab 7 (static MAC and spoofing), the master test matrix, break-it drills
Not in the lecture (beyond)	Chapter 24

What you need before you start

- What a MAC address is and how it looks (`0050.7966.6801`).
- What an Ethernet frame is, and that it has a **source** MAC and a **destination** MAC.
- Switch basics: a switch has ports. A port in **access mode** connects one end device.
- How a host reaches another subnet: it sends to its default gateway (Chapter 1 reminds you).

How to read this book

- Read **Part I** first. Each chapter ends with practice questions (answers included) and a **Term check** table.
- Do **Part II** in order. Each lab starts from the state of the reset in Chapter 20.
- Boxes that start with **Gotcha** are the mistakes that waste the most time.
- Predict each result before you read the expected result.

An honest note

- Sample outputs are **illustrative**. Wording, spacing and counters change between IOS versions. I flag the places where behavior depends on the version.
- I have **not** run, built or verified this lab. The expected results come from tracing each frame by hand. If your GNS3 result differs, your result is the evidence. Send it to me and we will find out why.
- MAC addresses of VPCS hosts are made up in the outputs (`0050.7966.68xx`). Yours will differ. Read yours with `show ip` and use them.

Part I — Theory

Ten chapters. Each one uses only words explained before it.

Chapter 1 — Why port security exists

Two addresses on every frame

Data on an Ethernet network travels in **frames**. A frame has two MAC addresses in its header:

- the **source MAC**: the address of the device that sent the frame, and
- the **destination MAC**: the address of the device that should receive it.

A **MAC address** is a 48-bit hardware address, written like `0050.7966.6801`. Every network card has one.

What a switch does with them

A **switch** is a box with many **ports**. You plug devices into them. The switch keeps a list called the **MAC address table**. Each line says: “this MAC address lives behind this port”.

The switch builds the list by itself. This is called **learning**. When a frame comes in on a port, the switch reads its **source** MAC and writes down: “this MAC is on this port”.

```
Frame arrives on Gi0/1
   src MAC = 0050.7966.6801
   dst MAC = (the gateway)

MAC address table (learned)
   MAC               Port
   0050.7966.6801    Gi0/1      <- just learned from the source MAC
```

When the switch needs to forward a frame, it reads the **destination** MAC and looks it up. If the MAC is in the table, the frame goes out of that one port. If it is **not** in the table (an **unknown unicast**), the switch **floods** the frame: it sends a copy out of every port except the one it came in on.

> **Analogy.** The switch is a receptionist with a visitor book. Every time someone walks in through a door, the receptionist writes down which door that person used. Later, when a parcel arrives for that person, the receptionist knows which door to send it to. If the person is not in the book, the receptionist shouts the parcel down every corridor.

The problem

By default, a switch trusts everybody. Any device that plugs into any port gets in. Nothing asks “are you allowed?”. That opens three doors for trouble:

Problem	What happens
Rogue device	Someone plugs a laptop into a free wall socket and joins your network.
Extra devices on one port	Someone plugs a small switch under a desk. Now one port carries many devices.
MAC flooding	An attacker sends thousands of frames with made-up source MACs. The table is limited in size. When it is full, the switch can no longer learn, and it floods traffic out of many ports. The attacker may then see frames meant for others.

What port security does

Port security is a switch feature that limits **which source MAC addresses may send frames into a port**, and **how many**. If a frame arrives from a MAC that is not allowed, the switch takes an action you choose (Chapter 6).

This stops the three problems above:

- A rogue laptop has the wrong MAC, so it is blocked.
- A small switch under a desk brings several MACs, and the count goes over the limit.
- MAC flooding creates many MACs on one port, and the limit stops it after the first few.

What port security does not do

- It does **not** check passwords or identity. It only reads the source MAC in the frame.
- A MAC address is **easy to copy**. A person who knows an allowed MAC can set the same MAC on their own device. Lab 7 shows this.
- It does **not** encrypt anything.

So port security is a **cheap first fence**, not a lock. It is one layer in a bigger defence.

Practice questions

1. Which MAC address does a switch read to learn where a device is: source or destination?
2. A frame arrives with a destination MAC that is not in the table. What does the switch do?
3. Why does a small switch under a desk make a normal port see “too many” MAC addresses?

Answers

1. The **source** MAC.
2. It floods the frame out of every port except the one it arrived on.
3. Every device behind that small switch sends frames with its own source MAC. All of them arrive on the one port, so the main switch sees many MACs there.

Term check

Term	One-line meaning
Frame	A unit of data on Ethernet, with a source MAC and a destination MAC.
MAC address	A 48-bit hardware address, for example `0050.7966.6801`.
MAC address table	The switch’s list of “MAC lives behind this port”.
Learning	The switch reads the source MAC of arriving frames and records the port.
Flooding	Sending a frame out of all ports except the one it came in on.
MAC flooding attack	Filling the MAC table with fake MACs so the switch floods.
Port security	A switch feature that limits which source MACs may use a port.

Chapter 2 — Enabling port security

The three questions

Port security answers three questions for each port:

1. **Which** MAC addresses are allowed? (the *secure MAC addresses*)
2. **How many** may there be? (the *maximum*)
3. **What happens** when a frame from a MAC that is not allowed arrives? (the *violation mode*)

A **secure MAC address** is a MAC address that the switch has accepted for a port. A **violation** is the event where a frame from a not-allowed MAC arrives (or the count goes over the maximum).

The defaults

When you turn port security on and change nothing else:

Setting	Default
Maximum number of secure MACs	**1**
How the allowed MAC is found	**Dynamically**: the first source MAC seen on the port becomes the allowed one
Violation mode	**Shutdown**
Aging	Off (aging time 0)

So by default: one device, the first one to speak wins, and anyone else makes the port shut down.

Three ways a secure MAC address gets onto a port

Way	How it gets there	Survives a port going down?	Written into the running config?
Static	You type it with a command	Yes	Yes
Dynamic	The switch learns the first MAC(s)	**No**, it is forgotten	No
Sticky	The switch learns it, then writes it into the config	Yes	Yes

You can mix them. With a maximum of 3, you could configure one MAC by hand and let the switch learn two more.

The commands

```
SW1(config)# interface g0/1
SW1(config-if)# switchport mode access
SW1(config-if)# switchport port-security
```

Command	What it does
`switchport mode access`	Makes the port an access port (one end device). Port security needs this first.
`switchport port-security`	Turns port security on for this port, with the defaults above.
`switchport port-security maximum <number>`	How many secure MACs the port may have.
`switchport port-security mac-address <mac>`	Adds one **static** secure MAC.
`switchport port-security mac-address sticky`	Turns on **sticky** learning (Chapter 8).
`switchport port-security mac-address sticky <mac>`	Adds one sticky MAC by hand.
`switchport port-security violation <mode>`	Picks the violation mode: `shutdown`, `restrict` or `protect` (Chapter 6).
`switchport port-security aging time <minutes>`	How long a secure MAC lives (Chapter 7).
`switchport port-security aging type <type>`	How the aging timer counts: `absolute` or `inactivity` (Chapter 7).
`switchport port-security aging static`	Lets aging also remove static MACs (Chapter 7).

> **Gotcha:** A new switch port is usually in a **dynamic** mode (it may become a trunk). If you type `switchport port-security` on it, IOS refuses with a message like “Command rejected: … is a dynamic port”. The fix is to type `switchport mode access` first. The exact words vary by version.

What happens to each frame

Every frame that enters a secure port is checked by its **source** MAC. The switch looks at each frame on its own. It does not remember conversations.

```
Frame arrives, read source MAC
       |
  Is it a secure MAC for this port?
    yes -> forward the frame normally
    no  -> Is there room? (count < maximum)
             yes -> learn it as a new secure MAC, forward
             no  -> VIOLATION: do the violation action
```

Notice what is **not** checked: the destination MAC. Frames going **to** a device are not filtered by port security. Only frames coming **from** a device are checked.

A worked example

SW1 port Gi0/1 has port security on, with the defaults. PC1 (`0050.7966.6801`) sends its first frame.

1. The frame arrives. The switch reads the source MAC `0050.7966.6801`.
2. The port has no secure MAC yet, and the count (0) is below the maximum (1).
3. The switch learns `0050.7966.6801` as the one secure MAC. The count is now 1.
4. Later PC2 is plugged into the same port and sends a frame with a different source MAC. The count (1) already equals the maximum (1), and PC2's MAC is not the secure one. **Violation**: the port is shut down.

Practice questions

1. What are the default maximum and the default violation mode?
2. You type `switchport port-security` on a fresh port and IOS rejects it. What do you type first?
3. Does port security check the destination MAC of a frame?

Answers

1. Maximum 1, violation mode shutdown.
2. `switchport mode access`.
3. No. It only checks the source MAC of frames that enter the port.

Term check

Term	One-line meaning
Secure MAC address	A MAC the switch has accepted for this port.
Maximum	The most secure MACs a port may hold. Default 1.
Violation	A frame from a not-allowed MAC, or too many MACs.
Static / dynamic / sticky	Typed by you / learned and forgotten / learned and saved.
Access port	A port set to carry one end device, not a trunk.

Chapter 3 — Reading `show port-security interface`

Before you can trust a configuration, you must be able to read the evidence. These commands show you the state.

The show commands

Command	What it shows
`show port-security`	One line per secure port: maximum, current count, violations, action.
`show port-security interface <int>`	Full details for one port.
`show mac address-table secure`	The secure MAC addresses in the MAC table (Chapter 9).
`show errdisable recovery`	Whether automatic recovery is on, and which ports are waiting (Chapter 5).
`show interfaces status`	Port state. A shut-down secure port shows `err-disabled`.

The summary

Illustrative output:

```
SW1# show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
               (Count)       (Count)          (Count)
-----------------------------------------------------------------------------
      Gi0/1              1            1                  0         Shutdown
      Gi1/0              2            2                  0         Shutdown
-----------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 1
Max Addresses limit in System (excluding one mac per port) : 4096
```

One port in detail

Illustrative output:

```
SW1# show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0050.7966.6801:1
Security Violation Count   : 0
```

Line	What to learn from it
Port Security	`Enabled` or `Disabled`. If it says Disabled, nothing below matters.
Port Status	`Secure-up`: working. `Secure-down`: no device or cable. `Secure-shutdown`: err-disabled after a violation.
Violation Mode	Shutdown, Restrict or Protect (Chapter 6).
Aging Time / Aging Type	How long secure MACs live (Chapter 7). `0 mins` means never.
SecureStatic Address Aging	Whether static MACs also age.
Maximum MAC Addresses	The limit you set.
Total MAC Addresses	How many secure MACs the port holds **right now**.
Configured MAC Addresses	How many of them you typed (static).
Sticky MAC Addresses	How many are sticky.
Last Source Address:Vlan	The last source MAC seen on the port, and its VLAN. After a violation, this is the **offender**.
Security Violation Count	How many violations were counted (rules differ by mode).

> **Gotcha:** "Total MAC Addresses" counts what the port holds now. "Maximum" is the limit. If Total equals Maximum, the port is **full**. The next new MAC is a violation.

Practice questions

1. After a violation, which line tells you the MAC of the device that caused it?
2. Port Status shows `Secure-shutdown`. What happened?
3. Total MAC Addresses is 2 and Maximum is 2. A third MAC sends a frame. What happens?

Answers

1. **Last Source Address:Vlan**.
2. A violation happened in shutdown mode, and the port is err-disabled.
3. A violation. The port is full, so the third MAC is not allowed.

Term check

Term	One-line meaning
`show port-security`	Summary of all secure ports.
`show port-security interface`	Details of one secure port.
Secure-up / Secure-down / Secure-shutdown	Working / nothing connected / shut down by a violation.
Last Source Address	The last MAC seen on the port. After a violation, the offender.

Chapter 4 — When the port shuts down, and bringing it back

What err-disabled means

In **shutdown** mode, a violation puts the port into the **err-disabled** state. “Err” is short for “error”. The switch turns the port off, like pulling the cable. Nobody can use it, not even the legitimate device.

The switch prints log messages. Illustrative output (wording and order vary by version):

```
%PM-4-ERR_DISABLE: psecure-violation error detected on Gi1/0, putting Gi1/0 in err-disable state
%PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 0050.7966.6804 on port
GigabitEthernet1/0.
%LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed state to down
```

The word `psecure-violation` is the **reason**. You will meet it again in Chapter 5.

How to see it:

```
SW1# show interfaces status err-disabled
Port      Name        Status         Reason              Err-disabled Vlans
Gi1/0                 err-disabled   psecure-violation
```

```
SW1# show port-security interface g1/0
Port Status               : Secure-shutdown
Last Source Address:Vlan  : 0050.7966.6804:1
Security Violation Count  : 1
```

Bringing it back by hand

A port does not come back on its own (unless you turn on recovery, Chapter 5). To restore it, turn it off and on again:

```
SW1(config)# interface g1/0
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```

Do this first: remove the offender

> **Gotcha:** Do not re-enable the port while the unauthorized device is still connected. It will break the rules again and the port will shut down again.

The reason is subtle. Remember Chapter 2: **dynamic** secure MACs are forgotten when the port goes down. So after `shutdown` and `no shutdown`, the port starts with **no** secure MAC. The **first device to send a frame** becomes the new allowed device.

What is still connected	What happens after re-enable
Only the legitimate device	It talks first, becomes the secure MAC again. All is well.
Only the intruder	The intruder talks first and becomes the **new** secure MAC. The port now trusts the intruder.
Both, behind a small switch	Whoever talks first is accepted. The other one is a violation,

	and the port shuts down again.

This is why static and sticky MACs are better for important ports: they are not forgotten.

The violation counter

The lecture says that in shutdown mode the violation counter becomes 1 and is reset when the port is re-enabled. Some IOS versions behave a little differently, so check `show port-security interface` on yours.

Practice questions

1. What does `Secure-shutdown` mean?
2. After `shutdown` then `no shutdown`, what is the secure MAC on a port that used **dynamic** learning?
3. Why is it dangerous to re-enable a port while the intruder is still plugged in?

Answers

1. The port was shut down by a violation (err-disabled).
2. None. The first MAC to send a frame becomes the new secure MAC.
3. The intruder may become the trusted device, or the port shuts down again.

Term check

Term	One-line meaning
Err-disabled	The switch has turned the port off because of an error.
`psecure-violation`	The err-disable reason for a port security violation.
`shutdown` / `no shutdown`	Turn the port off / on again by hand.
Offender	The device whose MAC caused the violation.

Chapter 5 — ErrDisable recovery

The idea

Fixing err-disabled ports by hand does not scale. **ErrDisable recovery** tells the switch: “if a port is err-disabled for this reason, turn it on again by yourself after a while”.

The commands

These are **global** commands (typed in global configuration mode, not on a port):

```
SW1(config)# errdisable recovery cause psecure-violation
SW1(config)# errdisable recovery interval 30
```

Command	What it does
`errdisable recovery cause psecure-violation`	Turns recovery on for port security violations.
`errdisable recovery interval <seconds>`	How often the switch re-enables ports. Default is 300 seconds (5 minutes). The lowest value is usually 30.
`show errdisable recovery`	Shows which causes are on, the interval, and the ports that will be re-enabled.

The same pair of commands also covers other causes, such as `dhcp-rate-limit` and `arp-inspection`. Those belong to other topics.

Illustrative output:

```
SW1# show errdisable recovery
ErrDisable Reason            Timer Status
-----------------            --------------
bpduguard                    Disabled
psecure-violation            Enabled
...

Timer interval: 30 seconds

Interfaces that will be enabled at the next timeout:

Interface       Errdisable reason       Time left(sec)
---------       -----------------       --------------
  Gi1/0         psecure-violation          21
```

Do the warning from Chapter 4 again

> **Gotcha:** Recovery does not check that the intruder is gone. After the timer runs out the port comes back, with its dynamic MACs forgotten. If the intruder is still connected, then either the intruder becomes the new secure MAC, or the port shuts down again. Remove the offender first.

If the offender keeps coming back, the port cycles: down, up, violation, down again. You will see this in Lab 3.

Practice questions

1. Is `errdisable recovery` an interface command or a global command?
2. You set the interval to 60 seconds. A port is err-disabled at 10:00:00. Roughly when does it come back?
3. Does recovery remove an intruder?

Answers

1. A global command.
2. At the next 60-second timer tick after the violation (within about a minute). The timer runs on a schedule, so it can be sooner than 60 seconds.
3. No. It only turns the port back on.

Term check

Term	One-line meaning
ErrDisable recovery	Automatic re-enabling of err-disabled ports.
Cause	The reason for the err-disable that recovery should handle.
Interval	The time between recovery checks. Default 300 seconds.

Chapter 6 — Violation modes

The question

A frame arrives from a MAC that is not allowed. What should the switch do? You choose one of three answers.

Mode	In one line
Shutdown	Turn the whole port off. Loud and strict. This is the default.
Restrict	Drop the bad frames, keep the port up, and complain every time.
Protect	Drop the bad frames, keep the port up, and say nothing.

```
SW1(config-if)# switchport port-security violation shutdown
SW1(config-if)# switchport port-security violation restrict
SW1(config-if)# switchport port-security violation protect
```

Side by side

	Shutdown	Restrict	Protect
Frames from the unauthorized MAC	Dropped	Dropped	Dropped
Port state	**err-disabled (off)**	Stays up	Stays up
The legitimate device	**Loses its link too**	Keeps working	Keeps working
Log (Syslog) or SNMP message	Yes	Yes, for violations	**No**
Violation counter	Set to 1	**+1 per unauthorized frame**	**Does not increase**
How the port recovers	Manually, or with ErrDisable recovery	Nothing needed	Nothing needed
Default	**Yes**	No	No

Why the counter matters

In **restrict** mode the counter climbs with every dropped frame. It tells you how loud the intruder is. In **protect** mode the counter stays at 0, so a quiet attack leaves no trace on the switch.

> **Gotcha:** Protect hides attacks. If you see `Security Violation Count : 0` and no logs on a protect port, that does **not** prove nobody tried. The port may be dropping frames in silence.

What “dropped” means for the unauthorized device

In restrict and protect, the intruder is not cut off from the cable. It just never gets through. Its frames die at the switch port. Frames **to** it are not filtered (Chapter 2), but the intruder cannot complete any conversation because every reply it sends is dropped.

A trace: HA1 is secure, HA2 joins

The port has a maximum of 1. HA1 spoke first, so HA1 is the secure MAC. HA2 (behind the same small switch) now pings the gateway.

Step	What happens
1	HA2 sends an ARP broadcast (“who has the gateway address?”). Its source MAC is HA2’s.
2	The frame reaches the secure port. HA2’s MAC is not secure and the port is full. **Violation.**
3 (shutdown)	The port goes err-disabled. HA1 and HA2 both lose connectivity.
3 (restrict)	The frame is dropped, a log line is printed, the counter goes up. HA1 still works.
3 (protect)	The frame is dropped. Nothing is printed. The counter stays 0. HA1 still works.
4	HA2 never gets an ARP reply, so its ping fails in every mode.

Which one to choose

Situation	Good choice	Why
Server port, one known device	Shutdown	Any other device is an alarm worth an outage.
Reception desk, outages are costly, you still want alerts	Restrict	The port stays up, and the log tells you.
A lab port where extra devices are	Protect	Quiet, but you lose visibility.

common and noise is a nuisance		

Practice questions

1. Which mode leaves the legitimate device working **and** logs every violation?
2. Which mode never increases the violation counter?
3. Port is in restrict mode. The counter reads 37. What does that tell you?

Answers

1. Restrict.
2. Protect.
3. About 37 frames from unauthorized MACs were dropped. It does not mean 37 different devices.

Term check

Term	One-line meaning
Shutdown mode	A violation err-disables the port. Default.
Restrict mode	Drop bad frames, stay up, log and count them.
Protect mode	Drop bad frames, stay up, no log, no count.
Violation counter	How many violations the port has counted.
Syslog / SNMP	Ways a device sends alert messages to a log server or manager.

Chapter 7 — Secure MAC address aging

The idea

By default a dynamic secure MAC lives **forever** (until the port goes down). If a laptop is removed and a different laptop is plugged in, the new one is blocked, because the old MAC still fills the one slot.

Aging gives each dynamic secure MAC a lifetime. When the lifetime ends, the MAC is removed, and the slot is free for another device.

> **Analogy.** Parking permits. With no aging, a permit is valid for life. With aging, each permit expires after some minutes.

The settings

Setting	Default	Meaning
Aging time	**0 minutes**	0 means “never age”. Any other value is the lifetime in minutes.
Aging type	**absolute**	How the timer counts (below).
Static aging	Off	If on, **static** secure MACs also age. Normally only dynamic ones do.

Two ways to count

Type	The timer starts at…	Is it reset by traffic?	Effect
Absolute	The moment the MAC was learned	**No**	The MAC is removed when the time is up, even if the device is busy.
Inactivity	The last frame from that MAC	**Yes**, every frame resets it	The MAC is removed only after the device has been **silent** for the whole time.

```
Absolute (2 minutes)        Inactivity (2 minutes)
learned |--------| removed   learned |--> frame |--> frame |-----| removed
        0       2 min                  resets    resets     2 min of silence
```

Think of absolute as “this permit is valid for 2 minutes from today, no matter what”. Think of inactivity as “this permit lapses if you do not use it for 2 minutes”.

What happens to a busy device on absolute aging

When its MAC is removed, the device’s **very next frame** is learned again. So a busy, legitimate device barely notices. A silent device is the one that loses its slot. And a *different* device can take the slot if it speaks first.

The commands

```
SW3(config-if)# switchport port-security aging time 2
SW3(config-if)# switchport port-security aging type inactivity
SW1(config-if)# switchport port-security aging static
```

Illustrative evidence with `show port-security address` (this command is not in the lecture, but it shows the remaining time):

```
SW3# show port-security address
              Secure Mac Address Table
-----------------------------------------------------------------
Vlan    Mac Address       Type                Ports   Remaining Age
                                                         (mins)
----    -----------       ----                -----   -------------
   1    0050.7966.680a    SecureDynamic       Gi0/2         2 (I)
```

The `(I)` marks inactivity aging on many versions. Exact layout varies.

> **Gotcha:** With **inactivity** aging and a device that sends traffic all the time, the remaining age keeps jumping back to the full value, and the MAC **never** leaves. With **absolute** aging the remaining age counts all the way down and the MAC is then learned again.

Does sticky age?

The lecture teaches that sticky MACs do not age out. That is the normal behaviour. Static MACs only age if you turn on `aging static`.

Practice questions

1. What does an aging time of 0 mean?
2. A device sends a frame every 10 seconds. Aging time is 2 minutes. Will its MAC ever be removed with inactivity aging? With absolute aging?
3. Why would you want a MAC to age?

Answers

1. Aging is off. Secure MACs never expire.
2. Inactivity: never, because every frame resets the timer. Absolute: yes, after 2 minutes, but it is learned again right away by the next frame.
3. To free the slot so a different device can use the port after the first one is gone.

Term check

Term	One-line meaning
Aging	Giving a secure MAC a limited lifetime.
Aging time	The lifetime in minutes. 0 = forever.
Absolute aging	The timer runs from learning and is not reset.

Inactivity aging	The timer resets whenever a frame from that MAC arrives.
Static aging	Lets static secure MACs age too.

Chapter 8 — Sticky secure MAC addresses

The problem with dynamic MACs

A dynamic secure MAC is forgotten when the port goes down. Static MACs survive but you must type every one. Typing MACs for 200 ports is slow and error-prone.

The idea

Sticky learning gives you both: the switch **learns** the MAC itself, then **writes it into the running configuration** as if you had typed it.

> **Analogy.** Dynamic is a list written in pencil that is erased when the room is locked. Sticky is the same list copied into the permanent notebook. But the notebook (running config) only survives a reboot if you also **save** it.

The commands

```
SW1(config)# interface g0/2
SW1(config-if)# switchport mode access
SW1(config-if)# switchport port-security
SW1(config-if)# switchport port-security mac-address sticky
```

Add one sticky MAC by hand:

```
SW1(config-if)# switchport port-security mac-address sticky 0050.7966.6802
```

What changes in the configuration

After PC2 sends a frame, the running config of the port contains (illustrative):

```
interface GigabitEthernet0/2
 switchport mode access
 switchport port-security
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0050.7966.6802
```

The last line was **added by the switch**. And `show port-security interface g0/2` shows `Sticky MAC Addresses : 1`.

Rules to remember

Rule	Detail
Turning sticky **on**	Any dynamic secure MACs already on the port become sticky.
Turning sticky **off** (`no switchport port-security mac-address sticky`)	The sticky MACs become ordinary dynamic MACs again.
Not saved	The sticky lines are only in the **running** config. After a reload without saving, they are gone.
Saved (`copy running-config startup-config`)	They come back after a reload.
Maximum	Sticky MACs count toward the maximum.
Aging	Sticky MACs do not age out in the normal setup.

> **Gotcha:** Sticky is only as permanent as your **save**. Many people configure sticky, forget `copy running-config startup-config`, and lose every learned MAC at the next reboot. The port then re-learns whoever speaks first.

> **Gotcha:** Sticky locks in the **first** device. If that first device was already the wrong one (say an attacker on a Friday night), the wrong MAC is now saved as trusted.

Practice questions

1. What is the difference between a dynamic and a sticky MAC after the port goes down?
2. You enabled sticky, tested for a day, and the switch reloaded after a power cut. The MACs are gone. What did you forget?
3. A port has three dynamic MACs. You turn sticky on. What happens to them?

Answers

1. Dynamic is forgotten. Sticky stays, because it is in the config.
2. `copy running-config startup-config`.
3. All three become sticky.

Term check

Term	One-line meaning
Sticky secure MAC	A learned MAC that is saved into the running config.
Running config	The configuration in memory now. Lost at reload unless saved.
Startup config	The saved configuration the switch loads at boot.

Chapter 9 — Secure MACs in the MAC address table

Where they appear

Secure MAC addresses are listed in the switch's normal MAC address table. To see only the secure ones:

```
SW1# show mac address-table secure
          Mac Address Table
-------------------------------------------
Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0050.7966.6801    DYNAMIC     Gi0/1
   1    0050.7966.6802    STATIC      Gi0/2
   1    000a.0000.0013    STATIC      Gi0/3
Total Mac Addresses for this criterion: 3
```

(Illustrative. Some versions label the types differently.)

Reading the Type column

Type shown	Which secure MACs
STATIC	Static secure MACs **and** sticky MACs
DYNAMIC	Dynamically learned secure MACs

> **Gotcha:** Do not read `STATIC` as "I typed it by hand". Sticky MACs also show as STATIC. To tell them apart, use `show port-security interface` (Configured vs Sticky counts) or look at the running config.

Why this matters for evidence

- If a MAC is **not** in `show mac address-table secure`, the port has not accepted it.
- Ports **without** port security still learn MACs normally. Those appear in `show mac address-table dynamic`, not in the secure list.

A MAC moving between secure ports

On many versions, if a secure MAC is seen on a **different** secure port in the same VLAN, the second port counts it as a violation. So moving a sticky device from one secured port to another can trigger a violation. Behavior varies by version.

MAC addresses can be copied

Port security trusts the source MAC in the frame. A device can **change** the MAC it uses (many operating systems and routers allow it). If an attacker learns an allowed MAC (by watching traffic, or from a label on the device), the attacker can copy it. Then the switch cannot tell them apart. Lab 7 does this on purpose.

So: port security limits the casual attacker and accidents. It does not stop a determined one. For that you need stronger tools (Chapter 24 names a few).

Practice questions

1. A sticky MAC appears with Type `STATIC`. Why?
2. Which command lists only secure MACs?
3. Can an attacker bypass port security by copying an allowed MAC?

Answers

1. Sticky MACs are stored like static ones, in the config, so the table marks them STATIC.
2. `show mac address-table secure`.
3. Yes, if they know the allowed MAC and can set it on their device.

Term check

Term	One-line meaning
`show mac address-table secure`	Lists secure MACs and their ports.
STATIC / DYNAMIC (in this table)	Static or sticky secure / dynamic secure.
MAC spoofing	Setting a device's MAC to someone else's.

Chapter 10 — Choosing settings, and a checklist

Choosing

I need	Use
One fixed, known device on a port (a server)	Static MAC, maximum 1, shutdown
One device, but I do not want to type the MAC	Sticky, maximum 1
A small switch under a desk with up to N devices	Maximum N
Alerts but no outage	Restrict
Silent dropping	Protect (accept the loss of visibility)
Devices that change over time (hot desks)	Dynamic with inactivity aging
Automatic return of err-disabled ports	`errdisable recovery cause psecure-violation`

The checklist when a port misbehaves

1. **Is the port an access port?** (`show running-config interface ...`)
2. **Is port security on?** (`show port-security interface ...`: Port Security Enabled)
3. **What is the status?** `Secure-up`, `Secure-down` or `Secure-shutdown`.
4. **Is it full?** Compare Total with Maximum.
5. **Which MAC was the last to send?** (`Last Source Address:Vlan`)
6. **Which mode?** Shutdown, restrict or protect. In restrict or protect, the port is up but frames may be dropped.
7. **Was the port down recently?** Then dynamic MACs were forgotten and the first new sender won.

8. **Is recovery on?** (`show errdisable recovery`) Is the offender still connected?
9. **Did you save the sticky MACs?** (`copy running-config startup-config`)

Most real problems are step 1, 4, 7 or 9.

Practice scenarios

A. The switch port for a payment server must only ever accept that one server. B. A meeting room has a small switch for up to 4 laptops. C. At the reception desk the PC must never lose its link, but IT wants an alert if a stranger plugs in.

Answers

A. Static MAC, maximum 1, shutdown. B. Maximum 4, restrict, maybe with inactivity aging so old laptops free their slots. C. Restrict.

Term check

Term	One-line meaning
Checklist	A fixed list of checks you run in order.
Full port	Total MAC addresses equals the maximum.

Part II — Labs

Build once, then run seven labs on the same network. Only one router (R1) is needed, and it only connects the three LANs. There is no routing protocol and no static route to configure, so every result you see comes from port security itself.

Chapter 11 — The topology, the addresses and the company rulebook

The story

A small company has three floors, each with one switch.

- The **Staff floor** (SW1) has three desks and a meeting-room mini switch (HUB-A) that shares one wall socket.
- The **Reception and Lab floor** (SW2) has a reception mini switch (HUB-B) and a lab mini switch (HUB-C).
- The **Server room** (SW3) has one server and one admin PC.

A single router, **R1**, is the gateway for all three floors. The three floors are three separate subnets.

A “mini switch” is the cheap unmanaged switch people hide under a desk. In GNS3 you use the built-in **Ethernet switch** for it. It has no port security. It is there so that **one wall port can see several MAC addresses**, which is the thing port security is built to catch.

Devices

Device	Role	Use in GNS3
R1	Gateway for all floors	IOS router (IOSv) with 3 interfaces
SW1, SW2, SW3	Floor switches **with port security**	A Layer 2 IOS switch image such as IOSvL2
HUB-A, HUB-B, HUB-C	Mini switches under desks	GNS3 built-in Ethernet switch (no config)
PC1, PC2, HA1, HA2, HA3, REC1, REC2, LAB1, LAB2, ADMIN	Simple hosts	VPCS (can only ping)
RH-A, SRV	**Router-hosts**: IOS routers acting as hosts, so their MAC can be changed	IOS router (IOSv) with 1 interface

> **Gotcha:** The GNS3 built-in Ethernet switch **cannot do port security**. SW1, SW2 and SW3 must be a real IOS switch image. Interface names (Gi0/0 and so on) follow IOSvL2. Use the names your image shows.

> **Gotcha:** VPCS can only ping, and you cannot easily change its MAC. That is why RH-A and SRV are IOS routers: on an IOS interface you can type `mac-address ...` to **change the MAC on purpose**. That is how we play “a different device” and “a spoofer”.

> **Resources.** The full lab runs three switch images and three router images. If your PC is short of memory, build the Staff floor first (Labs 1 to 3, 5, 6 and 7) and add the other floors later.

Addressing table

Device	Interface	Address	Role
R1	g0/0	192.168.10.1/24	Staff floor gateway
R1	g0/1	192.168.20.1/24	Reception and Lab gateway
R1	g0/2	192.168.30.1/24	Server room gateway
PC1	e0	192.168.10.11/24, gw .1	Staff desk
PC2	e0	192.168.10.12/24, gw .1	Staff desk

RH-A	g0/0	192.168.10.13/24, gw .1	Staff desk, router-host
HA1	e0	192.168.10.21/24, gw .1	Behind HUB-A
HA2	e0	192.168.10.22/24, gw .1	Behind HUB-A
HA3	e0	192.168.10.23/24, gw .1	Behind HUB-A
REC1	e0	192.168.20.11/24, gw .1	Behind HUB-B
REC2	e0	192.168.20.12/24, gw .1	Behind HUB-B
LAB1	e0	192.168.20.21/24, gw .1	Behind HUB-C
LAB2	e0	192.168.20.22/24, gw .1	Behind HUB-C
SRV	g0/0	192.168.30.10/24, gw .1	Server, router-host
ADMIN	e0	192.168.30.20/24, gw .1	Admin PC

The switches have no IP address. Nothing in this topic needs one.

Switch port plan

Switch	Port	Connects to	Port security role
SW1	Gi0/0	R1 g0/0	Uplink, **no** port security
SW1	Gi0/1	PC1	Dynamic, aging labs
SW1	Gi0/2	PC2	Sticky
SW1	Gi0/3	RH-A	Static MAC, spoofing
SW1	Gi1/0	HUB-A (HA1, HA2, HA3)	Violation, recovery, maximum
SW2	Gi0/0	R1 g0/1	Uplink, **no** port security
SW2	Gi0/1	HUB-B (REC1, REC2)	Restrict
SW2	Gi0/2	HUB-C (LAB1, LAB2)	Protect
SW3	Gi0/0	R1 g0/2	Uplink, **no** port security
SW3	Gi0/1	SRV	Static MAC, shutdown
SW3	Gi0/2	ADMIN	Inactivity aging

MAC addresses used in this book

You can set the MACs of RH-A and SRV yourself. The VPCS MACs are chosen by GNS3, so the ones below are **made up for the outputs**. Read yours with `show ip` and use them.

Host	MAC in the sample outputs	How you get yours
PC1	0050.7966.6801	`show ip` on the VPCS
PC2	0050.7966.6802	`show ip`
HA1	0050.7966.6803	`show ip`
HA2	0050.7966.6804	`show ip`
HA3	0050.7966.6805	`show ip`
REC1	0050.7966.6806	`show ip`
REC2	0050.7966.6807	`show ip`
LAB1	0050.7966.6808	`show ip`
LAB2	0050.7966.6809	`show ip`
ADMIN	0050.7966.680a	`show ip`
RH-A	000a.0000.0013	**You set it** with `mac-address`
SRV	000c.0000.0010	**You set it** with `mac-address`
R1 g0/0	5e00.0001.0000	`show interfaces g0/0` (made up here)

Link table

#	From	To
1	R1 g0/0	SW1 Gi0/0
2	SW1 Gi0/1	PC1 e0
3	SW1 Gi0/2	PC2 e0
4	SW1 Gi0/3	RH-A g0/0
5	SW1 Gi1/0	HUB-A port 0
6	HUB-A port 1	HA1 e0
7	HUB-A port 2	HA2 e0
8	HUB-A port 3	HA3 e0
9	R1 g0/1	SW2 Gi0/0
10	SW2 Gi0/1	HUB-B port 0
11	HUB-B port 1	REC1 e0
12	HUB-B port 2	REC2 e0
13	SW2 Gi0/2	HUB-C port 0
14	HUB-C port 1	LAB1 e0
15	HUB-C port 2	LAB2 e0
16	R1 g0/2	SW3 Gi0/0
17	SW3 Gi0/1	SRV g0/0
18	SW3 Gi0/2	ADMIN e0

The company rulebook

These rules decide what each lab must achieve.

1. **Staff desks** (SW1 Gi0/1, Gi0/2, Gi0/3): one device per desk.
2. **Meeting-room mini switch** (HUB-A on SW1 Gi1/0): up to **two** devices. A third one shuts the port, and the port comes back by itself.
3. **Reception** (HUB-B on SW2 Gi0/1): one device. An intruder must **not** take the reception link down, but IT wants **alerts**.
4. **Lab** (HUB-C on SW2 Gi0/2): one device. Extra devices are ignored **quietly**.
5. **Server** (SRV on SW3 Gi0/1): only the server's own MAC, typed by hand. Anything else shuts the port.
6. **Admin PC** (ADMIN on SW3 Gi0/2): one device. If it is silent for 2 minutes, the slot is freed.
7. **Uplinks to R1** have **no** port security.

Why this design

Concept	Where it shows up
Default dynamic learning, maximum 1	SW1 Gi0/1 (Lab 1)
Violation, err-disable, manual re-enable	SW1 Gi1/0 (Lab 2)
ErrDisable recovery	SW1 Gi1/0 (Lab 3)
Shutdown vs restrict vs protect	SW1 Gi1/0, SW2 Gi0/1, SW2 Gi0/2 (Lab 4)
Absolute vs inactivity aging	SW1 Gi0/1, SW3 Gi0/2, SW1 Gi1/0 (Lab 5)
Sticky	SW1 Gi0/2 (Lab 6)
Static MAC, MAC spoofing, static plus dynamic	SW1 Gi0/3, SW1 Gi1/0 (Lab 7)

Chapter 12 — Base configuration and baseline test

Configure R1

```
hostname R1
no ip domain-lookup
interface g0/0
 ip address 192.168.10.1 255.255.255.0
 no shutdown
interface g0/1
 ip address 192.168.20.1 255.255.255.0
 no shutdown
interface g0/2
 ip address 192.168.30.1 255.255.255.0
 no shutdown
```

R1 knows all three subnets as **connected** routes. No other routing is needed.

Configure the switches

SW1 (repeat for SW2 and SW3 with their own port lists):

```
hostname SW1
no ip domain-lookup
interface range g0/0 - 3 , g1/0
 switchport mode access
 spanning-tree portfast
 no shutdown
```

SW	Ports to put in access mode
SW1	Gi0/0, Gi0/1, Gi0/2, Gi0/3, Gi1/0
SW2	Gi0/0, Gi0/1, Gi0/2
SW3	Gi0/0, Gi0/1, Gi0/2

Why `switchport mode access`: port security needs it (Chapter 2). Why `spanning-tree portfast`: a new access port normally waits about 30 seconds before it forwards frames. That wait would make every test look broken right after `no shutdown`. IOS may print a portfast warning. That is normal.

> **Gotcha:** If a test fails right after you bring a port up, wait a few seconds and try again before you suspect port security.

Save each switch and router: `copy running-config startup-config`.

Configure the router-hosts

RH-A:

```
hostname RH-A
no ip domain-lookup
no ip routing
interface g0/0
 mac-address 000a.0000.0013
 ip address 192.168.10.13 255.255.255.0
 no shutdown
ip default-gateway 192.168.10.1
```

SRV (same pattern):

```
hostname SRV
no ip domain-lookup
no ip routing
interface g0/0
 mac-address 000c.0000.0010
 ip address 192.168.30.10 255.255.255.0
 no shutdown
ip default-gateway 192.168.30.1
```

`no ip routing` turns the router into a plain host. `ip default-gateway` is the gateway a host uses. If your IOS image refuses the `mac-address` command, read the real MAC with `show interfaces g0/0` and use that wherever the book says `000a.0000.0013` or `000c.0000.0010`. You will lose the "change the MAC on purpose" tests, so try the command first.

Configure the VPCS hosts

```
PC1> ip 192.168.10.11/24 192.168.10.1
PC2> ip 192.168.10.12/24 192.168.10.1
HA1> ip 192.168.10.21/24 192.168.10.1
HA2> ip 192.168.10.22/24 192.168.10.1
HA3> ip 192.168.10.23/24 192.168.10.1
REC1> ip 192.168.20.11/24 192.168.20.1
REC2> ip 192.168.20.12/24 192.168.20.1
LAB1> ip 192.168.20.21/24 192.168.20.1
LAB2> ip 192.168.20.22/24 192.168.20.1
ADMIN> ip 192.168.30.20/24 192.168.30.1
```

Save each with `save`. Run `show ip` on every VPCS and write down its MAC (Chapter 11).

Baseline test (no port security yet)

This proves the network works and shows what happens when nothing limits a port. Predict each result first.

#	From	Command	Expected	Why
1	PC1	`ping 192.168.10.1`	Success	Same subnet as R1. The ARP request and reply are plain Layer 2.
2	HA1	`ping 192.168.10.1`	Success	HA1 is behind HUB-A.
3	HA2	`ping 192.168.10.1`	Success	A **second** device behind the same wall port. Nothing stops it.
4	HA3	`ping 192.168.10.1`	Success	A **third** device. Still nothing stops it.
5	PC1	`ping 192.168.20.11`	Success	Routed through R1. The request goes up to R1 and down to REC1. The reply returns the same way.
6	SRV	`ping 192.168.10.11`	Success	Routed through R1, in the other direction.

> **Gotcha:** The first ping of a new path may lose its first reply. That is ARP learning, not a fault. Ping again.

> **The lesson of the baseline:** tests 2 to 4 are the problem. Three different devices share one switch port, and the switch is fine with it. Port security is how you make it **not** fine.

Evidence

On SW1, after those pings (illustrative):

```
SW1# show mac address-table dynamic
          Mac Address Table
-------------------------------------------
Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0050.7966.6801    DYNAMIC     Gi0/1
   1    0050.7966.6803    DYNAMIC     Gi1/0
   1    0050.7966.6804    DYNAMIC     Gi1/0
   1    0050.7966.6805    DYNAMIC     Gi1/0
   1    5e00.0001.0000    DYNAMIC     Gi0/0
Total Mac Addresses for this criterion: 5
```

Three MACs behind one port (`Gi1/0`). R1's MAC is on the uplink (`Gi0/0`). The `show port-security` command says that port security is off:

```
SW1# show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
               (Count)       (Count)          (Count)
---------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 4096
```

No secure ports are listed, because none exist yet.

Chapter 13 — Lab 1: Enabling port security

Starting state: base configuration only. No port security anywhere.

Task 1.1 — Turn it on with the defaults

Goal. Protect the Staff desk of PC1 (SW1 Gi0/1) with the plain defaults: maximum 1, dynamic learning, shutdown.

Configuration

```
SW1(config)# interface g0/1
SW1(config-if)# switchport port-security
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```

The `shutdown` and `no shutdown` make the port forget any MAC it learned in the baseline test, so we start clean.

Test table

#	From	Command	Expected	Request path and reply path
1	SW1	`show port-security interface g0/1`	`Port Security : Enabled`, Maximum 1, **Total 0**	No frame has arrived yet, so nothing is learned.
2	PC1	`ping 192.168.10.1`	Success	Request: PC1 sends an ARP broadcast with source `0050.7966.6801`. Gi0/1 is secure, empty, and the count (0) is below the maximum (1), so the MAC is learned and the frame is forwarded. Reply: R1 answers PC1 by unicast. It enters SW1 on Gi0/0 (no port security) and leaves on Gi0/1. Port security never checks the destination, so it passes.
3	SW1	`show port-security interface g0/1`	Total 1, Last Source `0050.7966.6801:1`, Violations 0	PC1's MAC is now the secure MAC.
4	SW1	`show mac address-table secure`	One entry, `DYNAMIC`, port `Gi0/1`	Dynamic learning.
5	SW1	`show port-security`	One line for `Gi0/1`: Max 1, Current 1, Violations 0, `Shutdown`	The summary.

> **Gotcha:** On some versions, test 1 may still show a MAC from the baseline. If so, the `shutdown` and `no shutdown` did not clear it, or you skipped them. Do them and look again.

Verification (illustrative)

```
SW1# show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0050.7966.6801:1
Security Violation Count   : 0
```

```
SW1# show mac address-table secure
          Mac Address Table
-------------------------------------------
Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0050.7966.6801    DYNAMIC     Gi0/1
Total Mac Addresses for this criterion: 1
```

Lesson. One command turns it on. The first device to speak becomes the allowed device. Only **source** MACs are checked.

Task 1.2 — A dynamic MAC is forgotten when the port goes down

Goal. See why dynamic MACs are fragile.

Configuration

```
SW1(config)# interface g0/1
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```

Test table

#	From	Command	Expected	Why
1	SW1	`show port-security interface g0/1`	**Total 0**	The link went down, and the dynamic MAC was removed.
2	SW1	`show mac address-table secure`	Empty	Same reason.
3	PC1	`ping 192.168.10.1`	Success	PC1 speaks first, so it becomes the secure MAC again.
4	SW1	`show port-security interface g0/1`	Total 1 again	Learned again.

Lesson. Dynamic MACs live only as long as the link. After any link flap, the first device to speak wins. Remember this for Chapter 4's warning.

Task 1.3 — The "dynamic port" rejection

Goal. See the access-mode rule in action. We use Gi0/2 and then put it back.

Configuration

```
SW1(config)# interface g0/2
SW1(config-if)# switchport mode dynamic auto
SW1(config-if)# switchport port-security
```

Test

#	From	Command	Expected	Why
1	SW1	`switchport port-security` (as above)	**Rejected**, with a message such as `Command rejected: GigabitEthernet0/2 is a dynamic port.`	The port could still become a trunk, so IOS refuses. The wording varies by version.

Fix, then put it back

```
SW1(config-if)# switchport mode access
SW1(config-if)# switchport port-security
SW1(config-if)# no switchport port-security
```

The second command is now accepted. The third one removes port security again, because Gi0/2 is used for the sticky lab (Lab 6).

Lesson. `switchport mode access` first, port security second.

Chapter 14 — Lab 2: A violation, and re-enabling by hand

Starting state: base configuration. Gi0/1 from Lab 1 may stay as it is. Gi0/2 must be back to the base state.

Task 2.1 — Let HA1 in first

Goal. Protect the meeting-room wall socket (SW1 Gi1/0 with HUB-A) with the defaults. HA1 will be the first, so HA1 will be the secure MAC.

Configuration

```
SW1(config)# interface g1/0
SW1(config-if)# switchport port-security
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```

Test table

#	From	Command	Expected	Request path and reply path
1	HA1	`ping 192.168.10.1`	Success	Request: HA1's frames pass through HUB-A to Gi1/0. HA1's MAC is learned as the one secure MAC. Reply: R1 to HA1 through Gi0/0 and then Gi1/0.
2	SW1	`show port-security interface g1/0`	`Secure-up`, Total 1, Last Source HA1's MAC (`...6803`)	HA1 is the secure MAC.

Lesson. Behind one wall socket, the first one to speak wins the only slot.

Task 2.2 — HA2 joins: a violation

Goal. Cause a violation and read the evidence.

Test table

#	From	Command	Expected	Request path and reply path
1	HA2	`ping 192.168.10.1`	**Fail**	Request: HA2's frame (ARP or ICMP) crosses HUB-A and arrives at Gi1/0 with source `...6804`. The port is full (1 of 1) and `...6804` is not the secure MAC. **Violation.** In shutdown mode the port goes err-disabled. Reply: none, nothing was sent.
2	HA1	`ping 192.168.10.1`	**Fail**	The whole port is down. HA1 is punished too.
3	SW1	(console)	Log lines `%PM-4-ERR`	The switch tells you the reason and the offender.

			`_DISABLE: psecure-violation ...` and `%PORT_SECURITY-2-PSECURE_VIOLATION ... caused by MAC address 0050.7966.6804`	
4	SW1	`show interfaces status err-disabled`	`Gi1/0 err-disabled psecure-violation`	The port is off.
5	SW1	`show port-security interface g1/0`	`Secure-shutdown`, Last Source `...6804`, Count 1	The offender is HA2.
6	PC1	`ping 192.168.10.1`	Success	Gi0/1 is a different port. The violation did not spread.

Verification (illustrative)

```
%PM-4-ERR_DISABLE: psecure-violation error detected on Gi1/0, putting Gi1/0 in err-disable state
%PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 0050.7966.6804 on port
GigabitEthernet1/0.
%LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed state to down
```

```
SW1# show port-security interface g1/0
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
...
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Last Source Address:Vlan   : 0050.7966.6804:1
Security Violation Count   : 1
```

Lesson. In shutdown mode, one bad device takes down the good one too. `Last Source Address` names the offender.

Task 2.3 — Re-enable with the intruder still connected

Goal. Prove the warning from Chapter 4.

Configuration

```
SW1(config)# interface g1/0
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```

Test table

#	From	Command	Expected	Why
1	SW1	`show port-security interface g1/0`	`Secure-up`, **Total 0**	The dynamic MAC (HA1) was forgotten when the port went down.
2	HA2	`ping 192.168.10.1`	**Success**	HA2 speaks first, so **the intruder is now the trusted device**.
3	HA1	`ping 192.168.10.1`	**Fail**	HA1 is now the unauthorized MAC. The port goes err-disabled again.
4	SW1	`show port-security interface g1/0`	`Secure-shutdown`, Last Source HA1's MAC (`...6803`)	The roles swapped.

Lesson. A dynamic port trusts whoever speaks first after the link comes up. Re-enabling is not the same as fixing.

Task 2.4 — Re-enable properly

Goal. Remove the offender first, then re-enable.

Configuration

1. In GNS3, right-click **HA2** and choose **Stop**. This is the same as unplugging it.
2. Then:

```
SW1(config)# interface g1/0
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```

Test table

#	From	Command	Expected	Why
1	HA1	`ping 192.168.10.1`	Success	HA1 is the only talker, so it becomes the secure MAC again.
2	SW1	`show port-security interface g1/0`	`Secure-up`, Total 1, Last Source HA1's MAC, Count 0	Healthy. The counter may or may not be 0 on your version.

HA3 is also connected, but silent. It does nothing until it speaks.

Lesson. Remove the cause, then re-enable. The command order is: fix, `shutdown`, `no shutdown`.

Chapter 15 — Lab 3: ErrDisable recovery

Starting state: run Reset A (Chapter 20). HA2 is running again, and Gi1/0 has the defaults.

Task 3.1 — Turn recovery on

Goal. Let the switch re-enable err-disabled ports by itself, every 30 seconds.

Configuration

```
SW1(config)# errdisable recovery cause psecure-violation
SW1(config)# errdisable recovery interval 30
```

Test

#	From	Command	Expected	Why
1	SW1	`show errdisable recovery`	`psecure-violation  Enabled`, `Timer interval: 30 seconds`, no ports waiting	Recovery is on.

Task 3.2 — Cause a violation and watch the timer

Test table

#	From	Command	Expected	Why
1	HA1	`ping 192.168.10.1`	Success	HA1 is the secure MAC.
2	HA2	`ping 192.168.10.1`	**Fail**	Violation. Gi1/0 is err-disabled.
3	SW1	`show errdisable recovery`	`Gi1/0 psecure-violation` with **Time left** of 30 seconds or less	The port is waiting.
4	SW1	(console, within 30 seconds)	`%PM-4-ERR_RECOVER: Attempting to recover from psecure-violation err-disable state on Gi1/0`	The switch re-enabled it.

5	SW1	`show port-security interface g1/0`	`Secure-up`, **Total 0**	Back up, with no secure MAC.

> **Gotcha:** The timer ticks on a fixed schedule. The port may recover **sooner** than 30 seconds after the violation. That is normal.

Verification (illustrative)

```
SW1# show errdisable recovery
ErrDisable Reason            Timer Status
-----------------            --------------
psecure-violation            Enabled
...
Timer interval: 30 seconds

Interfaces that will be enabled at the next timeout:

Interface       Errdisable reason       Time left(sec)
---------       -----------------       --------------
  Gi1/0         psecure-violation          21
```

Task 3.3 — Recovery does not remove the intruder

Goal. See the cycle.

Test table

#	From	Command	Expected	Why
1	HA2	`ping 192.168.10.1` (right after the port recovers)	Success	HA2 speaks first, so it is now the trusted MAC.
2	HA1	`ping 192.168.10.1`	**Fail**	HA1 is now the intruder. The port is err-disabled again.
3	SW1	(console, within about 30 seconds)	Another `ERR_RECOVER` line	The port comes back up, then breaks again the next time somebody speaks.

Stop the cycle by stopping HA2 (right-click, **Stop**).

Lesson. Recovery gives the port a fresh start. It does not know who belongs. If the wrong device keeps talking, the loop continues.

Chapter 16 — Lab 4: Shutdown, restrict and protect side by side

Starting state: run Reset B (Chapter 20). Recovery stays on for SW1.

The same event (a second device speaks on a full port) happens three times, with three different modes.

Port	Mode	Devices
SW1 Gi1/0	Shutdown (default)	HA1, HA2
SW2 Gi0/1	Restrict	REC1, REC2
SW2 Gi0/2	Protect	LAB1, LAB2

Task 4.1 — Shutdown (confirm)

Test table

#	From	Command	Expected	Why
1	SW1	`show port-security interface g1/0`	`Violation Mode : Shutdown`	The default mode, already seen in Labs 2 and 3.

Task 4.2 — Restrict on the reception port

Goal. Rulebook rule 3: the reception link must stay up, and IT wants alerts.

Configuration

```
SW2(config)# interface g0/1
SW2(config-if)# switchport port-security
SW2(config-if)# switchport port-security violation restrict
SW2(config-if)# shutdown
SW2(config-if)# no shutdown
```

Test table

#	From	Command	Expected	Request path and reply path
1	REC1	`ping 192.168.20.1`	Success	REC1 speaks first and is the secure MAC.
2	REC2	`ping 192.168.20.1`	**Fail**	Request: the frame reaches Gi0/1 with source `...6807`. The port is full. **Violation.** In restrict mode the frame is **dropped**, a log line appears, and the counter goes up. The port stays up. Reply: none.
3	REC1	`ping 192.168.20.1`	**Success**	The port is still up, and REC1 is still the secure MAC.
4	SW2	`show port-security interface g0/1`	`Secure-up`, `Violation Mode : Restrict`, Last Source `...6807`, **Count above 0**	The count goes up by one per dropped frame, so it will be more than 1.
5	R1	`ping 192.168.20.12`	**Fail**	Request: R1's ARP broadcast reaches REC2 (the destination is not filtered). REC2's ARP **reply** is a frame **from** REC2 and is dropped at Gi0/1. R1 never learns REC2's MAC.

Verification (illustrative)

```
%PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 0050.7966.6807 on port
GigabitEthernet0/1.
```

```
SW2# show port-security interface g0/1
Port Status                : Secure-up
Violation Mode             : Restrict
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Last Source Address:Vlan   : 0050.7966.6807:1
Security Violation Count   : 5
```

The log may be rate-limited on some versions, so you might see fewer lines than violations.

Lesson. Restrict keeps the good device working and keeps a running count.

Task 4.3 — Protect on the lab port

Goal. Rulebook rule 4: extra devices are ignored quietly.

Configuration

```
SW2(config)# interface g0/2
SW2(config-if)# switchport port-security
SW2(config-if)# switchport port-security violation protect
SW2(config-if)# shutdown
SW2(config-if)# no shutdown
```

Test table

#	From	Command	Expected	Request path and reply path
1	LAB1	`ping 192.168.20.1`	Success	LAB1 is the secure MAC.
2	LAB2	`ping 192.168.20.1`	**Fail**	Request: the frame reaches Gi0/2 and is dropped silently. Reply: none.
3	SW2	(console)	**No log line**	Protect does not log.

4	LAB1	`ping 192.168.20.1`	**Success**	The port stayed up.
5	SW2	`show port-security interface g0/2`	`Secure-up`, `Violation Mode : Protect`, **Count 0**	Nothing was counted.

> **Gotcha:** On many versions, `Last Source Address` still shows the offender's MAC (`...6809`), even in protect mode. That is the only clue the switch gives you. It may differ on your version.

Lesson. Protect works, and it leaves almost no trace. Use it only where you accept that.

Task 4.4 — Compare

Fill in this table from your own observations. The expected values are in the last three columns.

Observation	Shutdown (SW1 Gi1/0)	Restrict (SW2 Gi0/1)	Protect (SW2 Gi0/2)
The second device's ping	Fails	Fails	Fails
The first device's ping afterwards	**Fails**	Works	Works
Port Status	Secure-shutdown	Secure-up	Secure-up
Log message	Yes	Yes	**No**
Violation count	1	Greater than 1	0
Recovery needed	Yes	No	No

Chapter 17 — Lab 5: Secure MAC address aging

Starting state: SW1 Gi0/1 still has port security with the defaults from Lab 1 (PC1). Run Reset C (Chapter 20) so SW1 Gi1/0 is at base. SW3 Gi0/2 is at base.

Task 5.1 — Absolute aging

Goal. See a timer that counts down no matter what.

Configuration

```
SW1(config)# interface g0/1
SW1(config-if)# switchport port-security aging time 2
SW1(config-if)# switchport port-security aging type absolute
```

Then on PC1, keep the traffic flowing: `ping 192.168.10.1 -t`. Press Ctrl+C later to stop it.

Test table

#	From	Command	Expected	Why
1	SW1	`show port-security interface g0/1`	`Aging Time : 2 mins`, `Aging Type : Absolute`	The settings are in place.
2	SW1	`show port-security address` (right after the first ping)	PC1's MAC, Remaining Age about **2**	The timer starts when the MAC is learned.
3	SW1	the same command about 1 minute later	Remaining Age about **1**	The ping is running, but absolute aging does not care.
4	SW1	the same command after about 2 minutes	Remaining Age back to about **2**	The MAC expired, PC1's next frame learned it again. PC1 notices nothing.

Verification (illustrative)

```
SW1# show port-security address
              Secure Mac Address Table
-----------------------------------------------------------------
Vlan    Mac Address       Type                Ports   Remaining Age
                                                         (mins)
----    -----------       ----                -----   -------------
   1    0050.7966.6801    SecureDynamic       Gi0/1        1
```

Lesson. Absolute aging removes the MAC on schedule. A busy device is simply learned again.

Task 5.2 — Inactivity aging

Goal. Rulebook rule 6: free the admin slot after 2 minutes of silence.

Configuration

```
SW3(config)# interface g0/2
SW3(config-if)# switchport port-security
SW3(config-if)# switchport port-security aging time 2
SW3(config-if)# switchport port-security aging type inactivity
```

On ADMIN: `ping 192.168.30.1 -t`.

Test table

#	From	Command	Expected	Why
1	SW3	`show port-security address` (several times over 3 minutes)	Remaining Age stays near **2** and keeps jumping back up	Every frame resets the inactivity timer.
2	ADMIN	Press Ctrl+C to stop the ping	The ping stops	ADMIN is now silent.
3	SW3	`show port-security interface g0/2` (after about 2 minutes and 30 seconds)	**Total 0**	The MAC expired after 2 minutes of silence.

Lesson. Inactivity frees a slot only when the device is quiet. A chatty device keeps it forever.

Task 5.3 — Aging frees the slot for a different device

Goal. Show the purpose of aging with HUB-A.

Configuration

```
SW1(config)# interface g1/0
SW1(config-if)# switchport port-security
SW1(config-if)# switchport port-security violation restrict
SW1(config-if)# switchport port-security aging time 1
SW1(config-if)# switchport port-security aging type inactivity
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```

We use restrict here so that a violation does not shut the port.

Test table

#	From	Command	Expected	Why
1	HA1	`ping 192.168.10.1`	Success	HA1 takes the only slot.
2	HA2	`ping 192.168.10.1` (at once)	**Fail**	The port is full. Restrict drops the frames.
3	all	**Do nothing for 90 seconds**		HA1 is silent. Its inactivity timer runs out.
4	HA2	`ping 192.168.10.1`	**Success**	The slot is free. HA2 speaks first and takes it.

5	HA1	`ping 192.168.10.1`	**Fail**	HA1 is now the unauthorized MAC.

Lesson. Aging turns a permanent claim into a temporary one. The device that speaks first after the slot frees wins.

Chapter 18 — Lab 6: Sticky secure MAC addresses

Starting state: run Reset D (Chapter 20). SW1 Gi0/2 is at base.

Task 6.1 — Turn sticky on and let PC2 in

Goal. Protect the desk of PC2 (SW1 Gi0/2) and have the switch **write the MAC into the config**.

Configuration

```
SW1(config)# interface g0/2
SW1(config-if)# switchport port-security
SW1(config-if)# switchport port-security mac-address sticky
SW1(config-if)# end
SW1# copy running-config startup-config
```

The save comes **before** PC2 speaks, so the startup config holds the sticky command but no MAC yet.

Test table

#	From	Command	Expected	Request path and reply path
1	PC2	`ping 192.168.10.1`	Success	PC2's frame is learned as a sticky secure MAC. The reply goes by Gi0/0.
2	SW1	`show running-config interface g0/2`	A new line `switchport port-security mac-address sticky 0050.7966.6802`	The **switch** wrote it.
3	SW1	`show port-security interface g0/2`	`Sticky MAC Addresses : 1`, Total 1	Learned and sticky.
4	SW1	`show mac address-table secure`	One entry, type **STATIC**, port `Gi0/2`	Sticky MACs show as STATIC.

Verification (illustrative)

```
SW1# show running-config interface g0/2
interface GigabitEthernet0/2
 switchport mode access
 switchport port-security
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0050.7966.6802
 spanning-tree portfast
```

Lesson. Sticky means "learn it, then write it down".

Task 6.2 — Sticky survives the port going down

Goal. Compare with Task 1.2.

Configuration

```
SW1(config)# interface g0/2
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```

Test table

#	From	Command	Expected	Why
1	SW1	`show port-security inter`	**Total 1**, Sticky 1,	The MAC is in the configuration, so the link flap did not

		`face g0/2`	before any traffic	remove it. In Task 1.2 the dynamic MAC was gone.
2	PC2	`ping 192.168.10.1`	Success	PC2 is still the allowed device.

Task 6.3 — Save or lose (optional, needs a reload)

Goal. See that sticky is only permanent if you save.

A. Reload without saving

```
SW1# reload
System configuration has been modified. Save? [yes/no]: no
Proceed with reload? [confirm]
```

#	After the reload	Expected	Why
1	`show running-config interface g0/2`	`switchport port-security mac-address sticky` is there, but the line **with PC2's MAC is gone**	The command was saved in Task 6.1, but the learned MAC was added later and not saved.
2	`show port-security interface g0/2`	Total 0	Nothing is remembered.

B. Learn again, save, reload

1. PC2 pings `192.168.10.1`. The sticky MAC is learned again.
2. `copy running-config startup-config`
3. `reload`

#	After the reload	Expected	Why
1	`show running-config interface g0/2`	The sticky MAC line is **present**	It was saved.

Reload time depends on your image. Expect one to two minutes.

Task 6.4 — Turn sticky off

Configuration

```
SW1(config)# interface g0/2
SW1(config-if)# no switchport port-security mac-address sticky
```

Test table

#	From	Command	Expected	Why
1	SW1	`show port-security interface g0/2`	`Sticky MAC Addresses : 0`, **Total 1**	The sticky MAC became an ordinary dynamic MAC.
2	SW1	`show mac address-table secure`	The entry is now type **DYNAMIC**	It is no longer stored in the config.
3	SW1	`show running-config interface g0/2`	The sticky MAC line is gone	Removed with the sticky command.

Lesson. Sticky on converts dynamic to sticky. Sticky off converts it back.

Chapter 19 — Lab 7 (extra): Static MACs, the MAC table and spoofing

> **Extra lab.** The lecture covers the static command only briefly. These tasks go deeper, and they prove Chapter 9's warning that MAC addresses can be copied.

Starting state: run Reset E (Chapter 20). SW1 Gi0/3 and Gi1/0 are at base. RH-A has MAC `000a.0000.0013`. SW1 recovery from Lab 3 may still be on.

Task 7.1 — A static secure MAC

Goal. Rulebook rule 1 for RH-A: only the MAC that **you type** may use this port.

Configuration

```
SW1(config)# interface g0/3
SW1(config-if)# switchport port-security
SW1(config-if)# switchport port-security mac-address 000a.0000.0013
```

Test table

#	From	Command	Expected	Request path and reply path
1	SW1	`show port-security interface g0/3`	**Total 1, Configured 1**, before any traffic	A static MAC is there from the moment you type it.
2	RH-A	`ping 192.168.10.1`	Success	RH-A's source MAC matches the static one. The reply returns by Gi0/0 and out of Gi0/3.
3	SW1	`show mac address-table secure`	`000a.0000.0013`, type **STATIC**, `Gi0/3`	Static MACs show as STATIC.

Lesson. A static MAC is allowed before any frame arrives, and it survives link flaps.

Task 7.2 — A different device

Goal. Play an intruder by changing RH-A's MAC.

Configuration (on RH-A)

```
RH-A(config)# interface g0/0
RH-A(config-if)# mac-address 000a.0000.00ff
```

Test table

#	From	Command	Expected	Request path and reply path
1	RH-A	`ping 192.168.10.1`	**Fail**	Request: the frame arrives with source `000a.0000.00ff`. It is not the static MAC and the port is full. **Violation.** Shutdown mode: Gi0/3 goes err-disabled. Reply: none.
2	SW1	`show port-security interface g0/3`	`Secure-shutdown`, Last Source `000a.0000.00ff`	The offender is named.

If recovery from Lab 3 is on, the port comes back by itself within about 30 seconds, and breaks again as soon as RH-A speaks.

Fix. Put the original MAC back, then re-enable the port.

```
RH-A(config-if)# mac-address 000a.0000.0013
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```


#	From	Command	Expected	Why
3	RH-A	`ping 192.168.10.1`	Success	The static MAC was **not** forgotten when the port went down.

Lesson. Static MACs do not depend on who speaks first.

Task 7.3 — Spoofing: copy the allowed MAC

Goal. Show that port security can be fooled by a copied MAC.

Pretend that `000a.0000.00aa` is the MAC of a legitimate laptop that RH-A has seen on a label.

Configuration

```
SW1(config)# interface g0/3
SW1(config-if)# no switchport port-security mac-address 000a.0000.0013
SW1(config-if)# switchport port-security mac-address 000a.0000.00aa
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```

Test table

#	From	Command	Expected	Why
1	RH-A (still `...0013`)	`ping 192.168.10.1`	**Fail**	The only allowed MAC is `...00aa`. Violation, port err-disabled.
2	RH-A	Set `mac-address 000a.0000.00aa` on g0/0		RH-A now copies the allowed MAC.
3	SW1	`shutdown`, then `no shutdown` on Gi0/3		Re-enable the port.
4	RH-A	`ping 192.168.10.1`	**Success**	The switch only reads the source MAC. It cannot tell RH-A from the real laptop.

Lesson. Port security checks a label, not an identity. Anyone who knows an allowed MAC can use it.

Task 7.4 — Static plus dynamic on the meeting-room port

Goal. Mix both kinds on HUB-A: maximum 2, one MAC typed, one learned.

First, start HA2 again if it is stopped. Then read HA1's MAC with `show ip` on HA1 (illustrative: `0050.7966.6803`).

Configuration

```
SW1(config)# interface g1/0
SW1(config-if)# switchport port-security
SW1(config-if)# switchport port-security maximum 2
SW1(config-if)# switchport port-security mac-address 0050.7966.6803
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```

Test table

#	From	Command	Expected	Why
1	SW1	`show port-security interface g1/0`	Maximum 2, **Total 1, Configured 1**	One slot is taken by the typed MAC (HA1).
2	HA3	`ping 192.168.10.1`	Success	HA3 speaks first and is **learned** into the second slot.
3	HA1	`ping 192.168.10.1`	Success	HA1 is the static MAC.
4	HA2	`ping 192.168.10.1`	**Fail**	Both slots are taken. Violation, port err-disabled (shutdown mode).
5	SW1	`show port-security interface g1/0`	`Secure-shutdown`, Last Source HA2's MAC	HA2 is the third MAC.

Verification (illustrative, before the violation)

```
SW1# show port-security interface g1/0
Port Status                : Secure-up
Violation Mode             : Shutdown
Maximum MAC Addresses      : 2
Total MAC Addresses        : 2
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
```

Lesson. A port can hold typed, learned and sticky MACs together. They all count toward the maximum.

Chapter 20 — Reset steps and GNS3 canvas labeling

The reset macro for one port

Use this to put any switch port back to the base state. It removes port security, aging, sticky lines and violation mode.

```
SW1(config)# default interface g1/0
SW1(config)# interface g1/0
SW1(config-if)# switchport mode access
SW1(config-if)# spanning-tree portfast
SW1(config-if)# no shutdown
```

Replace the port and the switch as needed. Some images print a portfast warning. That is fine.

Resets between labs

Reset	When	Steps
A	Before Lab 3	Start HA2 in GNS3. On SW1: run the macro on Gi0/2 (Lab 1 task 1.3 touched it). On SW1 Gi1/0: `switchport port-security`, then `shutdown` and `no shutdown`.
B	Before Lab 4	Start HA2 if it is stopped. On SW1 Gi1/0: `shutdown`, `no shutdown` (defaults, shutdown mode). Recovery stays on.
C	Before Lab 5	On SW1 Gi1/0: run the macro. SW1 Gi0/1 keeps its Lab 1 settings.
D	Before Lab 6	On SW1 Gi0/2: run the macro. On SW1 Gi0/1: run the macro (aging is finished).
E	Before Lab 7	On SW1 Gi0/3 and Gi1/0: run the macro. RH-A: `mac-address 000a.0000.0013`. Start HA2.
Master	Before Chapter 21	On every secured port of SW1, SW2 and SW3: run the macro. On RH-A and SRV: set the original MACs. Then apply the master configuration.

Global recovery commands to remove:

```
SW1(config)# no errdisable recovery cause psecure-violation
SW1(config)# no errdisable recovery interval
```

Clearing learned MACs without shutting a port

```
SW1# clear port-security dynamic interface g1/0
```

This removes dynamic secure MACs only. It is not in the lecture, but it is handy between tests. It leaves static and sticky MACs alone on most versions.

GNS3 canvas labeling reference

A well-labeled canvas makes a good screenshot.

Device	Label to type under it
R1	`R1 gateway` and its three interface IPs
SW1, SW2, SW3	Name and floor
HUB-A, HUB-B, HUB-C	`mini switch`
Hosts	Name and last octet (`.11` , `.21`)
RH-A, SRV	Name, last octet and `MAC set`

Zone	Suggested rectangle color	Label
Staff floor	Blue	192.168.10.0/24
Reception and Lab floor	Yellow	192.168.20.0/24
Server room	Green	192.168.30.0/24

Use the drawing tools in the GNS3 toolbar. Turn on interface labels so the Gi0/x names show on the links. Add the secure-port roles next to each port (for example `Gi1/0 max 2`).

> **Remember:** The diagram on the front page is a design drawing, not a GNS3 screenshot. Replace it with your own screenshot once you have built the lab.

Chapter 21 — Master configuration and test matrix

The master state

This is the final state, built from the base configuration. Port by port, the rulebook is satisfied.

```
! ---- SW1 ----
errdisable recovery cause psecure-violation
errdisable recovery interval 30
interface g0/1
 switchport port-security
interface g0/2
 switchport port-security
 switchport port-security mac-address sticky
interface g0/3
 switchport port-security
 switchport port-security mac-address 000a.0000.0013
interface g1/0
 switchport port-security
 switchport port-security maximum 2
```

```
! ---- SW2 ----
interface g0/1
 switchport port-security
 switchport port-security violation restrict
interface g0/2
 switchport port-security
 switchport port-security violation protect
```

```
! ---- SW3 ----
errdisable recovery cause psecure-violation
errdisable recovery interval 30
interface g0/1
 switchport port-security
 switchport port-security mac-address 000c.0000.0010
interface g0/2
 switchport port-security
 switchport port-security aging time 2
 switchport port-security aging type inactivity
```

Save every switch: `copy running-config startup-config`. Make sure every VPCS is running and the router-hosts have their MACs.

Master test matrix

Run the rows in order. A row's result depends on who spoke first.

#	From and command	Expected	Why
1	PC1 `ping 192.168.10.1`	Success	Dynamic, first device, max 1
2	PC2 `ping 192.168.10.1`	Success	Sticky learns PC2. `show running-config interface g0/2` gets a sticky line
3	RH-A `ping 192.168.10.1`	Success	Its MAC equals the static one
4	HA1 `ping 192.168.10.1`	Success	First of two allowed on HUB-A
5	HA2 `ping 192.168.10.1`	Success	Second of two
6	HA3 `ping 192.168.10.1`	**Fail**	A third MAC. Gi1/0 goes err-disabled. **HA1 and HA2 lose their link too**
7	Stop HA3. Wait up to 30 seconds. HA1 `ping 192.168.10.1`, then HA2	Success, success	Recovery re-enabled the port. HA1 and HA2 become the secure MACs again
8	REC1 `ping 192.168.20.1`	Success	First on a restrict port
9	REC2 `ping 192.168.20.1`	**Fail**	Dropped, logged. Port stays up
10	REC1 `ping 192.168.20.1`	Success	REC1 is unaffected
11	LAB1 `ping 192.168.20.1`	Success	First on a protect port
12	LAB2 `ping 192.168.20.1`	**Fail**	Dropped silently, no log, count stays 0
13	LAB1 `ping 192.168.20.1`	Success	Protect keeps the port up
14	SRV `ping 192.168.30.1`	Success	Its MAC equals the static one
15	On SRV set `mac-address 000c.0000.00ee`, then `ping 192.168.30.1`	**Fail**	Violation on SW3 Gi0/1. Port err-disabled
16	On SRV set `mac-address 000c.0000.0010`. Wait up to 30 seconds. `ping 192.168.30.1`	Success	Recovery brought the port back. The static MAC was not forgotten
17	ADMIN `ping 192.168.30.1`, then wait 3 minutes in silence. `show port-security interface g0/2`	Total 0	Inactivity aging freed the slot
18	PC1 `ping 192.168.20.11`	Success	Routed through R1. Port security does not stop **allowed** hosts from using the router
19	R1 `ping 192.168.20.12` (REC2)	**Fail**	R1's ARP broadcast reaches REC2, but REC2's reply is a frame from an unauthorized MAC and is dropped at SW2 Gi0/1
20	SW1 `show port-security`	Lists Gi0/1, Gi0/2, Gi0/3, Gi1/0	Summary of every secure port on SW1

> **Gotcha:** Row 6 is the lesson of shutdown mode. Even innocent devices behind the same wall socket go dark.

> **Gotcha:** Row 19 shows that port security is not a one-way filter in practice. Frames **to** a blocked device are delivered, but nothing the blocked device sends comes back. So the conversation still fails.

Evidence commands for any row: `show port-security`, `show port-security interface <int>`, `show mac address-table secure`, `show interfaces status err-disabled`, `show errdisable recovery`.

Chapter 22 — Break-it-then-fix-it drills

Start every drill from the master state (Chapter 21). Break one thing, find the cause using only the evidence, fix it, then confirm with the matching matrix row. Read the symptom first. Cover the evidence and fix, and try to guess.

Drill 1 — The typo in the static MAC

Setup (SW3)

```
SW3(config)# interface g0/1
SW3(config-if)# no switchport port-security mac-address 000c.0000.0010
SW3(config-if)# switchport port-security mac-address 000c.0000.0101
SW3(config-if)# shutdown
SW3(config-if)# no shutdown
```

Symptom. SRV cannot reach its gateway. The port goes down and comes back every 30 seconds, and then fails again whenever SRV speaks.

Evidence (illustrative)

```
SW3# show port-security interface g0/1
Port Status                : Secure-shutdown
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Last Source Address:Vlan   : 000c.0000.0010:1
Security Violation Count   : 1
```

```
SW3# show mac address-table secure
   1    000c.0000.0101    STATIC      Gi0/1
```

The **Last Source** is SRV's real MAC (`...0010`). The **secure** MAC is `...0101`. One digit is wrong.

Fix

```
SW3(config-if)# no switchport port-security mac-address 000c.0000.0101
SW3(config-if)# switchport port-security mac-address 000c.0000.0010
SW3(config-if)# shutdown
SW3(config-if)# no shutdown
```

Lesson. After a violation, compare **Last Source** with the **secure** MAC. The difference is the bug. Confirm with matrix row 14.

Drill 2 — The old device still holds the slot

Setup (SW1). Put Gi1/0 back to base, then:

```
SW1(config)# interface g1/0
SW1(config-if)# switchport port-security
SW1(config-if)# switchport port-security violation restrict
```

HA1 pings `192.168.10.1` (HA1 is now the secure MAC). Then right-click HA1 and **Stop** it. Now HA2 pings `192.168.10.1`.

Symptom. "HA1 has left and HA2 sits at the same desk, but HA2 has no network."

Evidence (illustrative)

```
SW1# show port-security interface g1/0
Port Status                : Secure-up
Violation Mode             : Restrict
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Last Source Address:Vlan   : 0050.7966.6804:1
Security Violation Count   : 4
```

The port is up and the count is rising. `Total` is 1, so the port is **full**. The secure MAC is still HA1's. The mini switch keeps the link to SW1 **up**, so the port never flapped and the dynamic MAC was never forgotten.

Fix. Free the slot.

```
SW1# clear port-security dynamic interface g1/0
```

Or use `shutdown` and `no shutdown` on the port, or turn on aging (Lab 5).

Lesson. With a mini switch under the desk, the port stays up when one device leaves. A stale dynamic MAC can block the replacement for ever unless aging is on.

Drill 3 — The quiet block

Setup. Use the master state. LAB1 pings `192.168.20.1` first. Then LAB2 pings it.

Symptom. "LAB2 has no network. And IT sees no alert on the switch."

Evidence (illustrative)

```
SW2# show port-security interface g0/2
Port Status               : Secure-up
Violation Mode            : Protect
Maximum MAC Addresses     : 1
Total MAC Addresses       : 1
Last Source Address:Vlan  : 0050.7966.6809:1
Security Violation Count  : 0
```

No log lines, count 0, the port looks healthy. The clues are: **Total equals Maximum** (the port is full), and **Last Source** (on many versions) is not the secure MAC.

Fix. Decide what the rulebook wants.

- If LAB2 is allowed: `switchport port-security maximum 2`.
- If you want alerts: `switchport port-security violation restrict`.

Lesson. Protect mode hides the problem. A count of 0 does not mean "no violations".

Drill 4 — The recovery loop

Setup. On SW1 Gi1/0 set `switchport port-security maximum 1` (instead of 2). On HA1 and HA2 run `ping 192.168.10.1 -t` at the same time.

Symptom. "The meeting-room network goes up and down about every half minute."

Evidence (illustrative)

```
%PM-4-ERR_DISABLE: psecure-violation error detected on Gi1/0, putting Gi1/0 in err-disable state
%PM-4-ERR_RECOVER: Attempting to recover from psecure-violation err-disable state on Gi1/0
%PM-4-ERR_DISABLE: psecure-violation error detected on Gi1/0, putting Gi1/0 in err-disable state
```

`show errdisable recovery` shows the port with a countdown, again and again. `Last Source Address` changes between HA1 and HA2.

Fix. The cause is two devices on a one-device port. Stop one of them, or (if two are allowed) set `maximum 2`. Do **not** turn off recovery as the fix.

Lesson. Recovery repeats the break as long as the cause is still connected.

Drill 5 — Sticky MACs that were never saved

Setup. From the master state, let PC2 ping `192.168.10.1` (the sticky MAC is learned). Then run `reload` and answer **no** to saving.

Symptom. "After the power cut, the desk of PC2 accepts the first device that speaks, not PC2."

Evidence

```
SW1# show port-security interface g0/2
Sticky MAC Addresses       : 0
Total MAC Addresses        : 0
```

```
SW1# show running-config interface g0/2
 switchport port-security mac-address sticky
```

The sticky **command** is there. The sticky **MAC** line is not.

Fix. Let PC2 speak again, then save.

```
SW1# copy running-config startup-config
```

Lesson. Sticky MACs live in the running config. Save after learning. Confirm with Lab 6, Task 6.3.

Drill 6 — More static MACs than the maximum

Setup (SW3)

```
SW3(config)# interface g0/1
SW3(config-if)# switchport port-security mac-address 000c.0000.0011
```

Symptom. IOS rejects the second static MAC (or, on some versions, accepts it and the port still allows only one).

Evidence (illustrative; the wording varies by version)

```
% Maximum number of secure addresses has been reached
```

```
SW3# show port-security interface g0/1
Maximum MAC Addresses      : 1
Configured MAC Addresses   : 1
```

Fix. Raise the maximum **first**, then add the MAC.

```
SW3(config-if)# switchport port-security maximum 2
SW3(config-if)# switchport port-security mac-address 000c.0000.0011
```

Lesson. Every secure MAC counts toward the maximum, including the ones you type. If you want two typed MACs, the maximum must be at least 2.

Chapter 23 — Quick reference

Commands

Job	Command
Make the port an access port	`switchport mode access`
Turn port security on	`switchport port-security`
Set the maximum	`switchport port-security maximum <n>`
Add a static MAC	`switchport port-security mac-address <mac>`
Turn sticky learning on	`switchport port-security mac-address sticky`
Add a sticky MAC by hand	`switchport port-security mac-address sticky <mac>`
Set the violation mode	`switchport port-security violation <shutdown, restrict or protect>`
Set the aging time (minutes)	`switchport port-security aging time <n>`
Set the aging type	`switchport port-security aging type <absolute or inactivity>`

Let static MACs age	`switchport port-security aging static`
Turn recovery on	`errdisable recovery cause psecure-violation`
Set the recovery interval	`errdisable recovery interval <seconds>`
Re-enable a port by hand	`shutdown` then `no shutdown`
Clear dynamic secure MACs (extra)	`clear port-security dynamic interface <int>`

The show commands

Command	Use it to
`show port-security`	See all secure ports in one table
`show port-security interface <int>`	See status, mode, counts, last source, violations
`show port-security address` (extra)	See secure MACs with their remaining age
`show mac address-table secure`	See secure MACs and their type
`show interfaces status err-disabled`	List ports shut down by an error
`show errdisable recovery`	See recovery status and the waiting ports

Defaults

Setting	Default
Maximum MACs	1
Learning	Dynamic (first MAC seen)
Violation mode	Shutdown
Aging time	0 (never)
Aging type	Absolute
Static aging	Off
Recovery interval	300 seconds
Recovery	Off

The three kinds of secure MAC

	Static	Dynamic	Sticky
Typed by you	Yes	No	Optional
Learned	No	Yes	Yes
In the running config	Yes	No	Yes
Survives link down	Yes	**No**	Yes
Ages by default	No	Only with aging	No

The three violation modes

	Shutdown	Restrict	Protect
Frames dropped	Yes	Yes	Yes
Port up	**No**	Yes	Yes
Log	Yes	Yes	**No**
Counter	Set to 1	Rises	Stays 0
Recovery	Needed	Not needed	Not needed

Aging

	Absolute	Inactivity
Timer starts	When learned	At the last frame
Reset by traffic	No	Yes
Busy device	Re-learned at once	Never removed

Port status values

Status	Meaning
Secure-up	Working
Secure-down	Nothing connected
Secure-shutdown	Err-disabled by a violation

When it fails: nine checks

1. Is the port in access mode?
2. Is port security on (`Port Security : Enabled`)?
3. What is the status?
4. Is the port full (Total equals Maximum)?
5. What is the Last Source Address?
6. Which violation mode?
7. Did the port flap, so dynamic MACs were forgotten?
8. Is recovery on, and is the offender still connected?
9. Did you save the sticky MACs?

Part III — Beyond the videos

> **Beyond the videos.** Nothing in this part is in the lecture. These are short previews so that you know the next doors. They are optional.

Chapter 24 — What comes after port security

Voice VLAN and port security

An IP phone often sits between the wall and a PC, so one port sees **two** MACs (phone and PC). A port that holds a phone needs a maximum of at least 2, or 3 on some setups. You will meet this on real desks.

Limits per VLAN

On some versions, you can set the maximum and the secure MACs **per VLAN** on a port, not just per port. That matters for trunks and voice ports.

SNMP traps and syslog

Restrict and shutdown send messages. You can send them to a log server (Syslog) and a management system (SNMP). Port security also has its own SNMP trap setting on many versions.

Other things that err-disable a port

BPDU Guard, DHCP Snooping rate limits and Dynamic ARP Inspection also err-disable ports. The `errdisable recovery cause ...` command works for them with a different cause name.

Stronger tools

Tool	What it adds
802.1X	Real authentication: the device or user proves who it is before the port opens.
DHCP Snooping	Blocks rogue DHCP servers and builds a table of which IP belongs to which port.
Dynamic ARP Inspection	Uses that table to stop fake ARP messages.
BPDU Guard	Shuts a port if a switch is plugged in where only hosts belong.
Storm control	Limits floods of broadcast or unknown traffic.

Testing MAC flooding

Attackers use tools that send thousands of frames with random source MACs. With port security and a small maximum, the port stops after the first few. Doing this test needs a tool outside this book, and it should only be done on your own lab.

The honest summary

Port security is cheap and easy. It stops accidents and casual attacks. It does not stop someone who copies an allowed MAC. For high-value networks, add 802.1X.

Appendices

Appendix A — Glossary

Term	Meaning
Absolute aging	The aging timer starts when the MAC is learned and is not reset.
Access port	A switch port set for one end device (not a trunk).
Aging	Giving a secure MAC a limited lifetime.
Aging time	The lifetime in minutes. 0 means never.
CAM table	Another name for the MAC address table.
Dynamic secure MAC	A secure MAC learned automatically and forgotten when the port goes down.
Err-disabled	The switch has turned the port off because of an error.
ErrDisable recovery	Automatic re-enabling of err-disabled ports after an interval.
Flooding	Sending a frame out of every port except the one it arrived on.
Frame	A unit of data on Ethernet, with a source and destination MAC.
Inactivity aging	The aging timer resets every time a frame from the MAC arrives.
Learning	The switch reads the source MAC of a frame and records the port.
MAC address	A 48-bit hardware address such as `0050.7966.6801`.
MAC address table	The switch's list of which MAC lives behind which port.
MAC flooding	Filling the MAC table with fake MACs.
MAC spoofing	Setting a device's MAC to someone else's.
Maximum	The most secure MACs a port may hold. Default 1.
Mini switch	A small unmanaged switch under a desk. Called HUB-A, B, C in this book.
Offender	The device whose MAC caused a violation.
Port security	A switch feature that limits which source MACs may use a port.
Protect	Violation mode: drop bad frames, no log, no count.
Psecure-violation	The err-disable reason for a port security violation.
Restrict	Violation mode: drop bad frames, keep the port up, log and count.
Router-host	An IOS router set up to act like a PC (`no ip routing` and `ip default-gateway`).
Secure MAC address	A MAC the switch has accepted for a port.
Shutdown	Violation mode: err-disable the port. The default.
Static secure MAC	A secure MAC you type by hand.
Sticky secure MAC	A learned secure MAC that is saved in the running config.
Syslog	A way to send log messages to a log server.
Unknown unicast	A frame whose destination MAC is not in the table.
VPCS	A tiny GNS3 host that can only do basic tasks like ping.
Violation	A frame from a not-allowed MAC, or too many MACs.