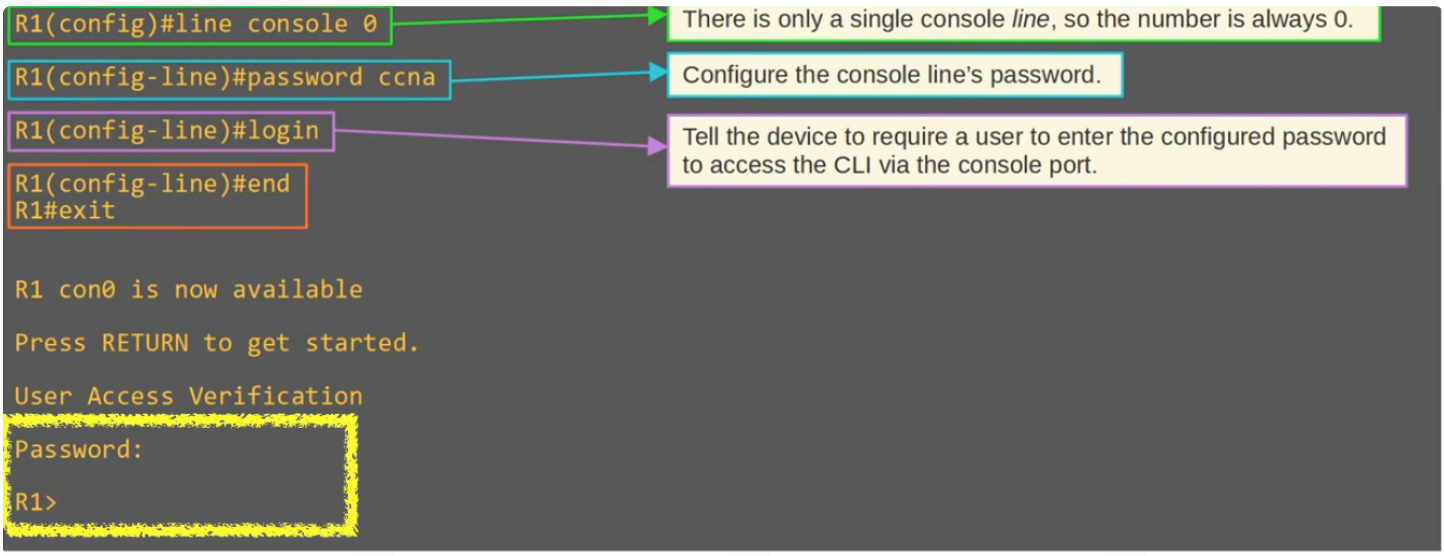
Lecture Introduction

- Jeremy's IT Lab is a free, complete course for the CCNA
- This lecture covers **SSH (Secure Shell)**
- SSH is a protocol used to connect to devices and configure them via the CLI
- One option to connect to a device and configure it is via the console port, which was introduced in an early lecture of this course
- SSH allows you to connect to a device remotely, without being directly connected to its console port
- SSH is exam topic **4.8**, which says you must be able to configure network devices for remote access using SSH

Topics Covered in This Lecture

- **Console port security**
  - In the last lecture about Syslog, the console line and the VTY lines were mentioned
  - This lecture explains more about the console line and how to make it more secure
- **Management IP address on Layer 2 switches**
  - Layer 2 switches don't route packets and don't build a routing table
  - However, you can configure a management IP address on them so that you can remotely connect to them via SSH
- **Telnet**
  - A protocol similar to SSH, but simpler and less-secure
- **SSH** — the main topic of the lecture

Console Port Security



- BBy default, no password is needed to access the CLI of a Cisco IOS device via the console port

Example:

- You can take a new device out of the box
- Use a console cable to connect your laptop to it
- Then start configuring it in the CLI

- However, you can configure a password on the console line
    - The console line is where you configure all settings related to console port connections
- When you do this, a user will have to enter a password to access the CLI via the console port

To configure the console line, use the command `line console 0` from global config mode:

```
R1(config)# line console 0
```

- Then the command is PASSWORD, followed by the password

```
R1(config-line)# password ccna
```

- But configuring a password isn't enough — you have to use the LOGIN command
  - The LOGIN command tells the device to require a user to enter the password to access the CLI via the console port
- Note: Configuring a password alone isn't enough — the LOGIN command is also required

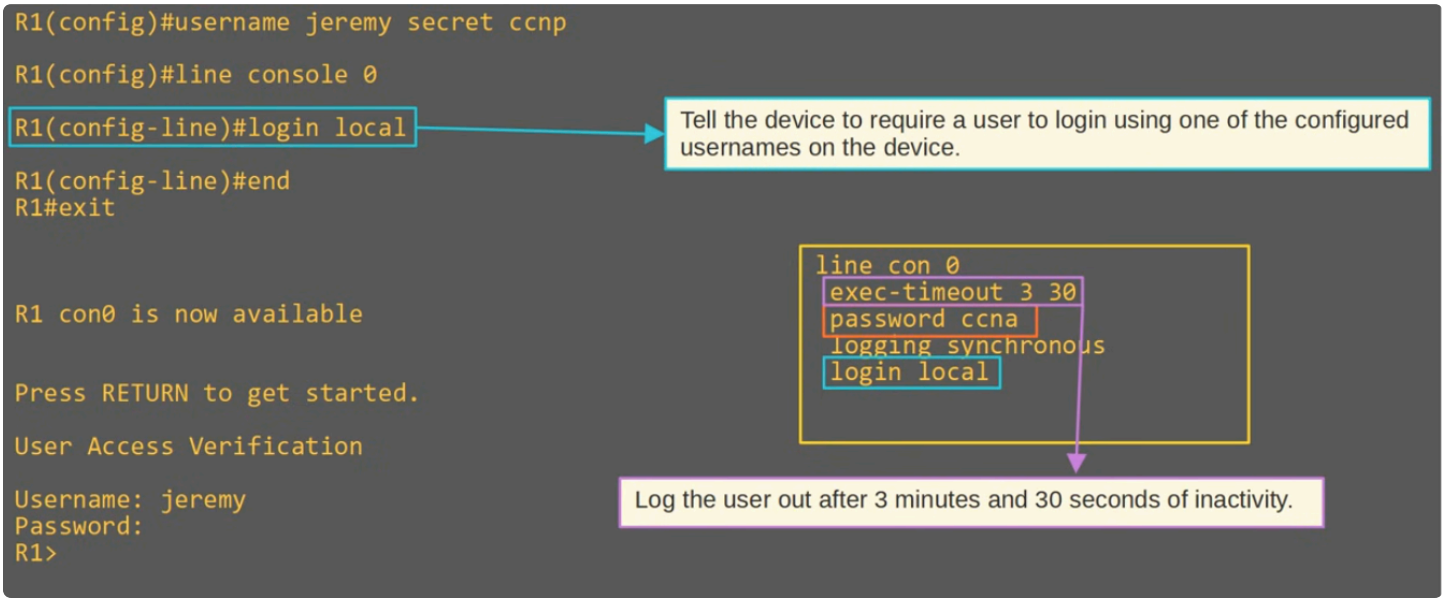
```
R1(config-line)# login
```

- That's it — now a password will be required to access the console port
- **Example:**
  - To demonstrate, the professor used `end` and `exit` to terminate the console connection
  - Then pressed enter to connect again, and was asked for a password
  - He entered the password of `ccna`, and was able to connect

  Note:
  - The password isn't displayed as you type it, so the password can't be seen by anyone nearby

---

Console Port Security (continued)



- Alternatively, you can configure the console line to require users to login using one of the configured usernames on the device
  - This is different than the previous example, in which a specific password was configured just for the console line
- First, the professor created a username, `jeremy`, and gave it a secret password of `ccnp`

```
R1(config)# username jeremy secret ccnp
```

- The professor once again used `line console 0` to configure the console line

```
R1(config)# line console 0
```

- Then used the command `login local`
  - This tells the device to require a user to login using one of the configured usernames on the device
  - So, instead of logging in using the password configured on the console line, the user will have to use a username and password

```
R1(config-line)# login local
```

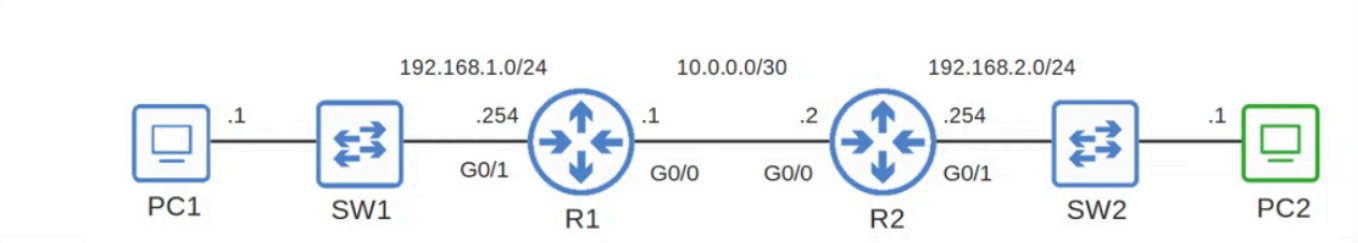
- By the way, here's the current configuration of R1's console line
  - Notice that the password of `ccna` that was configured previously is still there
  - However, the login mode was changed from `login` to `login local`
  - So the console line's password of `ccna` can no longer be used
  - The user must login using an account on the device
- **Example:**
  - The professor logged out of the connection
  - He pressed enter to connect again
  - He was asked for a username and password, not just a password
- One more command: `exec-timeout`
  - This will cause the device to log the user out after a certain period of inactivity, 3 minutes and 30 seconds in this case
  - This is a good security practice, in case you leave your desk but forget to log out of the console connection

```
R1(config-line)# exec-timeout 3 30
```

- That's all for console line security

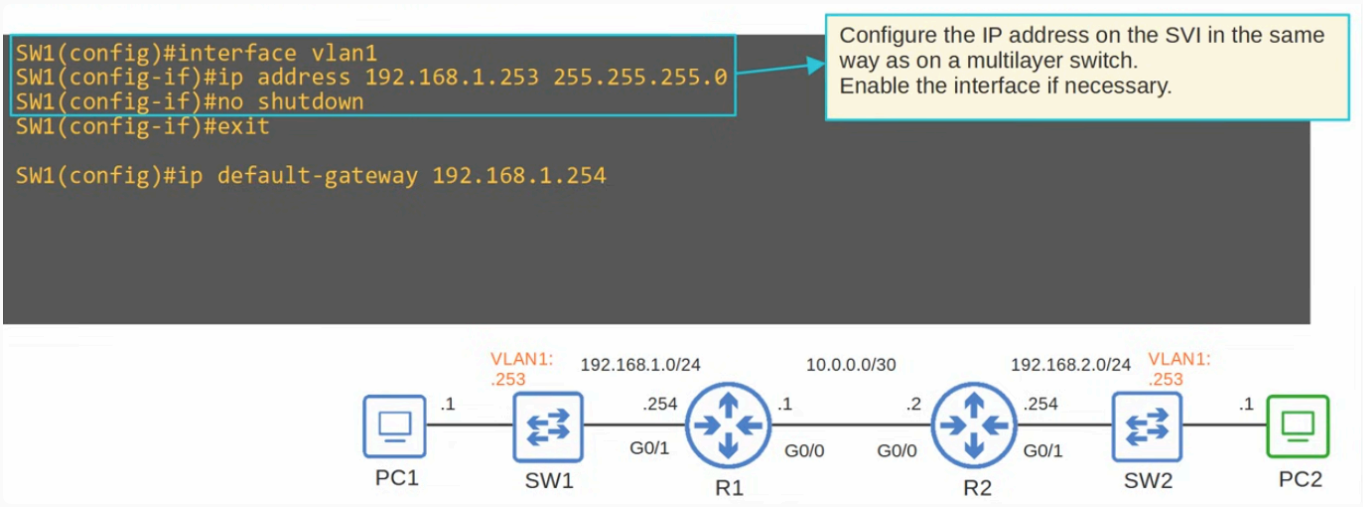
Management IP Address on Layer 2 Switches

- Routers and Layer 3 switches have IP addresses you can use to connect remotely to manage the devices
- However, what about Layer 2 switches?
  - Layer 2 switches don't perform packet routing and don't build a routing table
  - They aren't IP routing aware
  - Their purpose is simply to forward frames in the LAN
- You can actually assign an IP address to an **SVI**, a switch virtual interface, to allow remote connections to the CLI of the switch, for example using Telnet or SSH



- **Example:**
  - For the rest of this lecture, the professor uses this network topology
  - The network admin is using PC2 and needs to be able to connect to all of the devices in the network to configure them, without having to travel to different offices
  - To allow this, the Layer 2 switches need an IP address

- Here's how you can configure it:



  - First, configure the IP address on the SVI in the same way as on a multilayer switch
  - Use **`interface vlan`**, followed by the VLAN ID

```
SW1(config)# interface vlan <vlan-id>
```

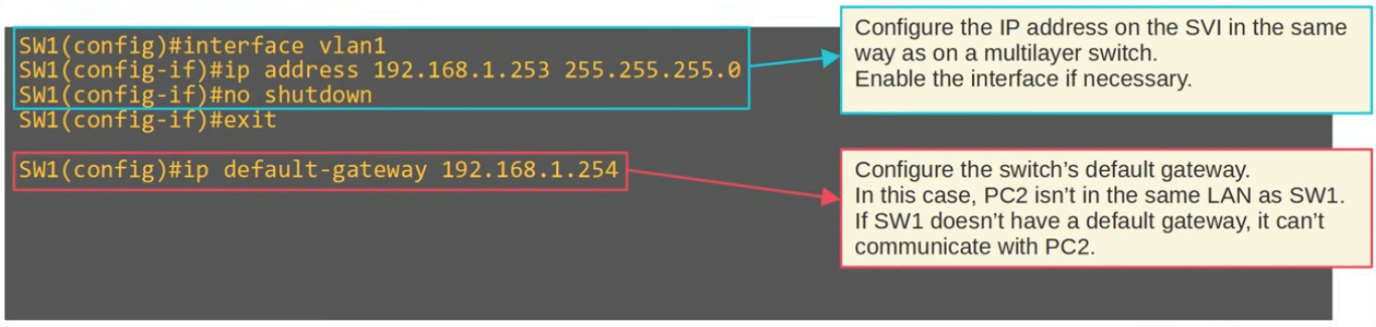
- Then configure the IP address

```
SW1(config-if)# ip address <ip-address> <subnet-mask>
```

- Finally, enable the interface if it's shutdown by default

```
SW1(config-if)# no shutdown
```

- So, the SVI is configured
- However, there's one more step you need to configure to allow a Layer 2 switch to communicate with devices outside of its local LAN



- Use the `ip default-gateway` command to configure the default gateway of the switch

```
SW1(config)# ip default-gateway <ip-address>
```

Why the Default Gateway Is Needed

- **Example:** PC2 isn't in the same LAN, so SW1 can't send network traffic directly to PC2. It has to send the traffic to a router, which will then route the packets to their destination.
- It's like configuring a default route; however, Layer 2 switches don't have a routing table, so you have to use this command to specify the default gateway instead.

Wrapping Up

- That's all the configuration needed for a Layer 2 switch's management IP
- To demonstrate Telnet and SSH, the professor continues configuring SW1

Telnet

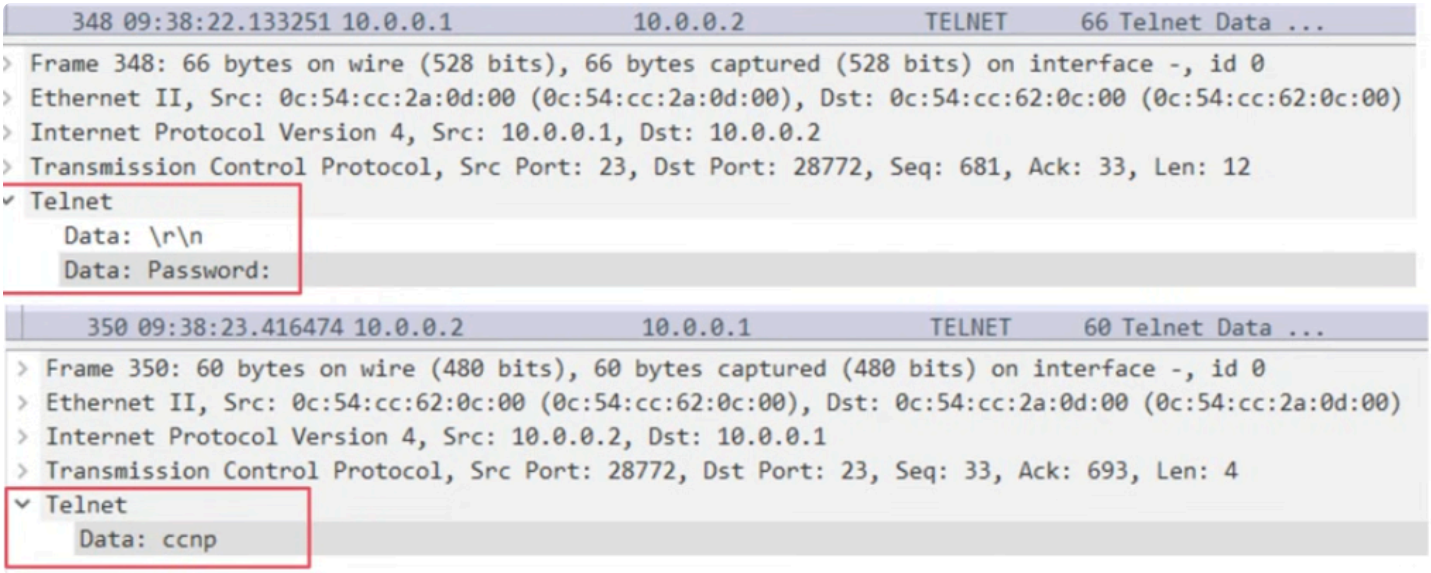
- Telnet is not commonly used today due to a lack of security, but it's good to know before looking at SSH

- **Telnet** (teletype network) is a protocol used to remotely access the CLI of a remote host
    - So instead of plugging your PC directly into the device with a console cable, you can connect to the device on a remote network

History and Security

- Telnet was developed in **1969**, so it's a very old protocol
- It has been largely, almost entirely, replaced by SSH, which is more secure
- However, SSH was developed in **1995**, so Telnet had many years of use
- Telnet sends data in **plain text**, no encryption
    - So if someone uses a tool like Wireshark to capture the traffic, they can read exactly what was sent

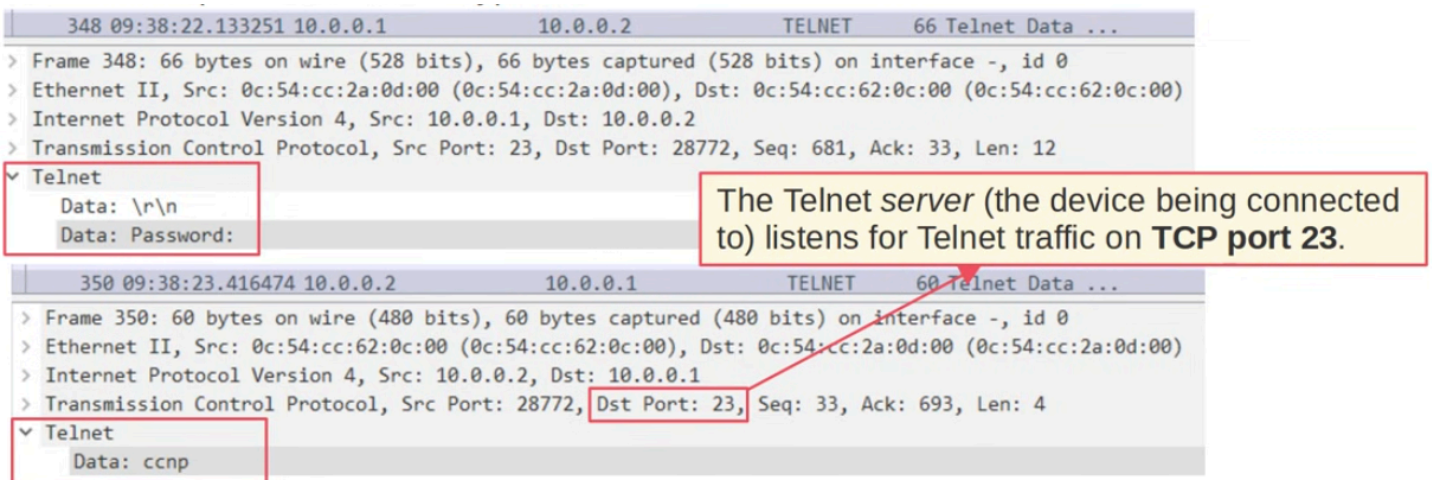
Example — Wireshark capture:



- Up top is a Telnet packet sent from R1, inside it says 'password'
    - This is the password prompt the CLI displays when trying to login
- The professor entered the password, and it was sent in a packet to R1
- But the password is displayed in plain text, `ccnp`, no encryption
- So, anyone in the middle who is able to capture the traffic like the professor did here will be able to read **all** of the traffic between the device and R1 — the username, the password, and all of the traffic after that

- That is definitely not secure, and that's why SSH is used instead of Telnet

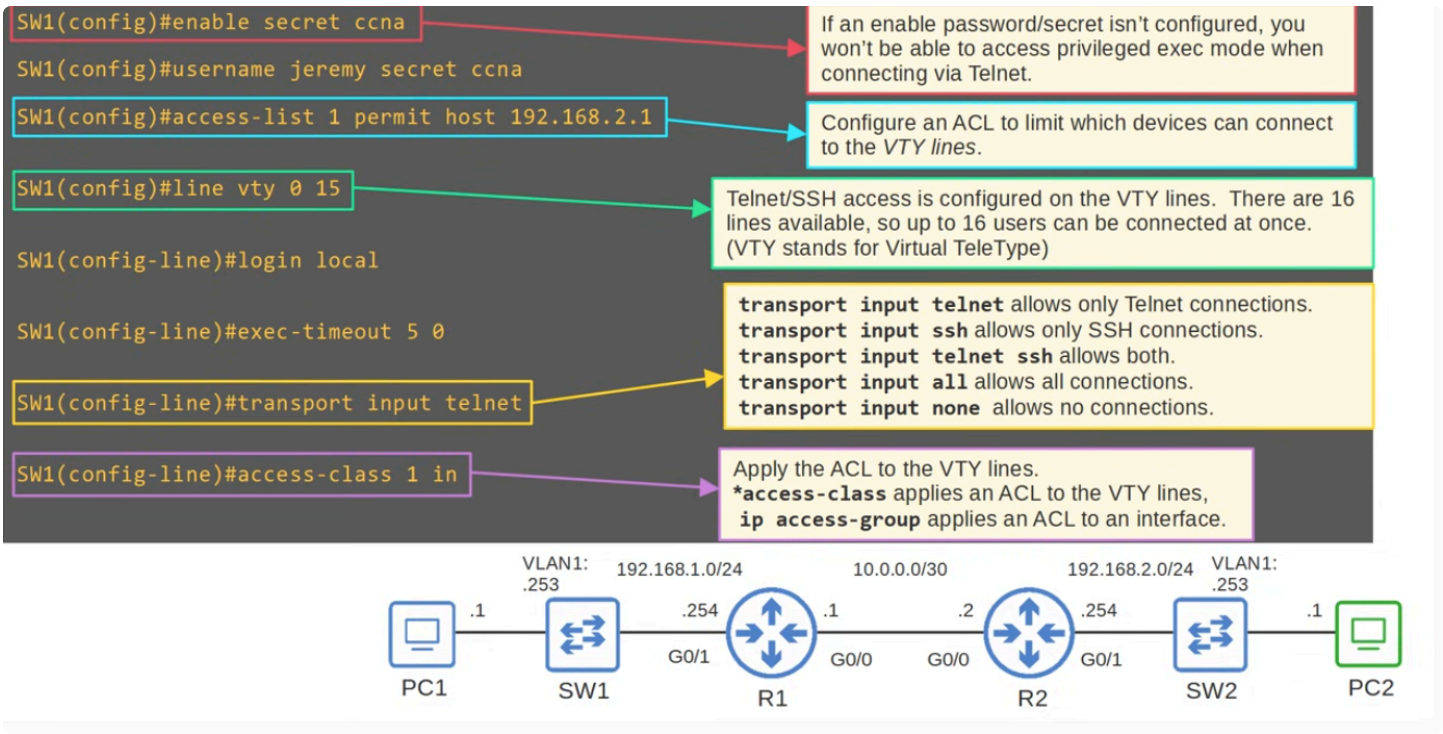
TCP Port and Client/Server Roles



- The **Telnet server** (the device being connected to, R1 in this case) listens for Telnet traffic on **TCP port 23**
  - So, when the professor's device sent the password to R1, the destination port was TCP port 23
  - **Note:** Make sure you remember that port number
- The **Telnet client** is the device that is connecting
- The **Telnet server** is the device that is being connected to

---

Telnet Configuration



- Here's how to configure a device so that you can use Telnet to connect to it
- First, you should always configure an `enable secret` — if you haven't, make sure you do
  - You won't be able to access privileged exec mode when connecting via Telnet if an `enable secret` isn't configured
- The professor also configured a username and password, since he will later configure `login local` mode
- This isn't necessary, but you can configure an ACL to limit which devices can connect to the VTY lines of the device

To configure the VTY lines, use the command `line vty 0 15`:

```
SW1(config)# line vty 0 15
```

- Telnet and SSH access is configured on the VTY lines

- There are 16 lines available, so up to 16 users can be connected at once
- `line vty 0 15` means you are configuring all lines, from 0 through 15
- This is recommended, so all of the VTY lines have the same configuration
- By the way, VTY stands for **Virtual TeleType**

On the VTY lines, the professor first configured `login local` as well as `exec-timeout`:

```
SW1(config-line)# login local
SW1(config-line)# exec-timeout 5 0
```

- The default exec timeout depends on the device
- In this case it was 10 minutes, but the professor shortened it to 5 minutes

Next, the professor used the command `transport input telnet`:

```
SW1(config-line)# transport input telnet
```

- This is how you configure what kind of connections to the VTY lines are allowed
- `transport input telnet` — allows only Telnet connections
- `transport input ssh` — allows only SSH connections
- `transport input telnet ssh` — allows both Telnet and SSH
- `transport input all` — allows all connections; there are some other protocols aside from Telnet and SSH
- `transport input none` — allows no connections to the VTY lines
- Note: The device used for this demo defaults to `transport input none`, but many devices default to `transport input all`

Finally, the professor applied the ACL to the VTY lines:

- This means that only PC2 will be able to connect to SW1 using Telnet
- Note: This only applies to VTY line connections; other devices will still be able to communicate with SW1, send it pings, etc.

As a reminder:

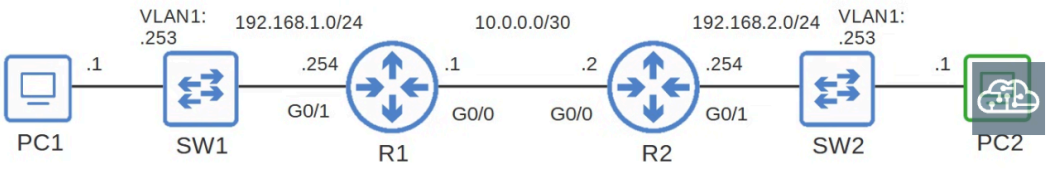
- The command to apply the ACL to the VTY lines is `access-class`
- The command to apply an ACL to an interface is `ip access-group`
- The command to actually create the ACL is `access-list` or `ip access-list`

Telnet Configuration (continued)

```
R2#ping 192.168.1.253
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.253, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/11/16 ms

R2#telnet 192.168.1.253
Trying 192.168.1.253 ...
% Connection refused by remote host
```

```
line vty 0 4
 access-class 1 in
 exec-timeout 5 0
 login local
 transport input telnet
line vty 5 15
 access-class 1 in
 exec-timeout 5 0
 login local
 transport input telnet
```

```
C:\Users\user>telnet 192.168.1.253
Connecting To 192.168.0.1…

User Access Verification

Username: jeremy
Password:
SW1>
```



- Try not to confuse those commands: `access-class`, `ip access-group`, and `access-list` or `ip access-list`

Verifying the Configuration

- To verify the configuration, the professor first tried to ping SW1 from R2
  - The ping was successful

- However, when he tried to telnet to SW1, he got a message saying "connection refused by remote host"
    - That's because of the ACL applied to SW1's VTY lines
    - Only PC2 should be able to Telnet to SW1
- So, he did Telnet from PC2, and it worked

Example:

- Notice how the VTY line configurations are displayed in the running config of the device
- The first 5 lines are displayed above, and the remaining 11 are below
- The professor believes this is just a result of the fact that old versions of IOS only had 5 VTY lines
- So, even if you configure all 16 lines at once, they display separately in the config
- That's just a bit of trivia, it doesn't have any effect on the operation of the lines

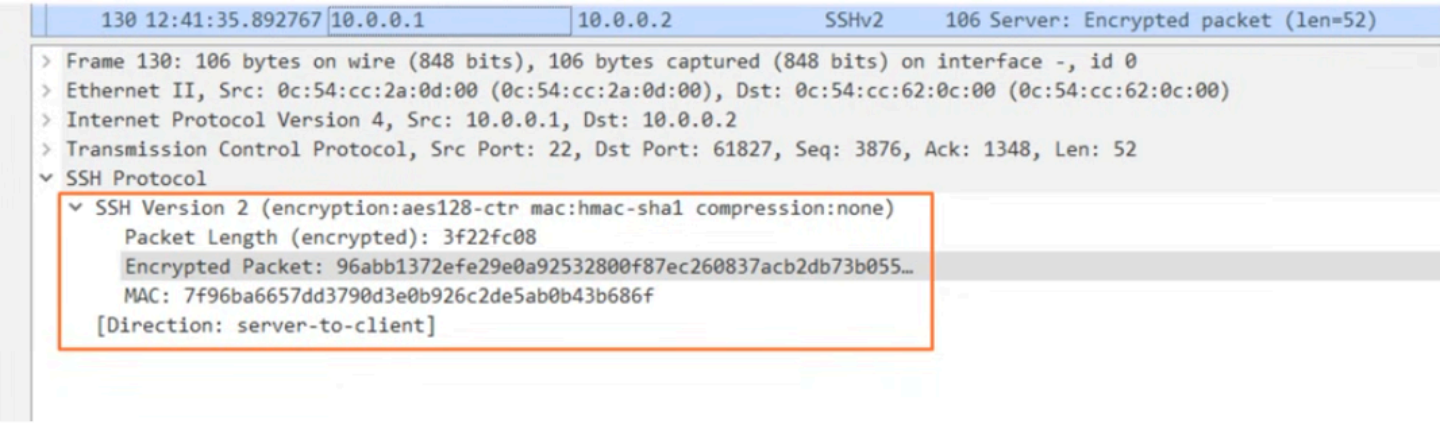
- So, that was a quick look at Telnet
- Finally, let's go to the main topic of the Lecture — SSH

SSH

- SSH, which stands for **Secure Shell**, was developed in 1995 to replace less secure protocols like Telnet
- By the way, if you're wondering what a 'shell' is, here's Wikipedia's definition: a shell is a computer program which exposes the operating system's services to a human user or other program
    - So, any time you're accessing the CLI of a device, that is a shell
- SSHv2, a major revision of SSHv1, was released in 2006
    - Version 2 is more secure, so it should be used whenever possible

- If a device supports both version 1 and version 2, it is said to run **version 1.99**
    - Note that 1.99 isn't actually a version of SSH, it just means that the device supports both version 1 and version 2
- SSH provides security features such as **data encryption** and **authentication**
    - You'll learn more about those terms in the security section of the course

Example — Wireshark capture:



- Here's an SSH packet that the professor captured in Wireshark
- Look at the encrypted packet section — it's just a seemingly random string of characters
- Only the SSH server and client have the keys to decrypt this packet
- So, even though he captured the packet on the way to its destination, he doesn't know what the contents are

- By the way, SSH uses **TCP port 22**
    - Remember that Telnet uses TCP 23 and SSH uses TCP 22

Checking IOS Support for SSH

```
SW1#show version
Cisco IOS Software, vios_l2 Software (vios_l2-ADVENTERPRISEK9-M), Version 15.2(4.0.55)E, TEST
ENGINEERING ESTG_WEEKLY BUILD, synced to  END_OF_FLO_ISP
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Tue 28-Jul-15 18:52 by sasyamal

SW1#show ip ssh
SSH Disabled - version 1.99
%Please create RSA keys to enable SSH (and of atleast 768 bits for SSH v2).
Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa
Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa
Encryption Algorithms:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
MAC Algorithms:hmac-sha1,hmac-sha1-96
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded): NONE
```

- Before configuring SSH, you should make sure that the version of IOS on your device supports it
- The professor used the `show version` command, and here's the IOS image name
  - Note the **K9** at the end that the professor highlighted
  - IOS images that support SSH will have **K9** in their name
- Cisco exports **NPE** (**No Payload Encryption**) IOS images to countries that have restrictions on encryption technologies
  - Those NPE IOS images do not support cryptographic features like SSH
  - Weak forms of encryption might be supported, but the professor would have to do more research on that to find out, and unfortunately he doesn't have access to any NPE IOS images

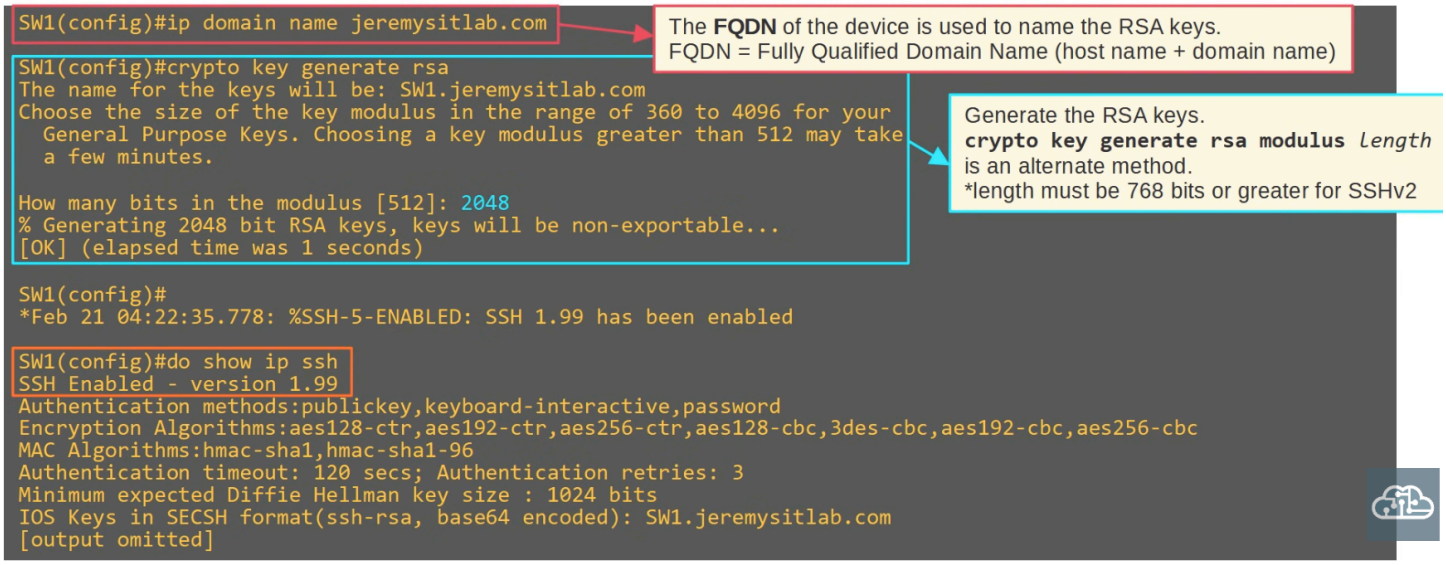
Verification:

```
R1# show version
R1# show ip ssh
```

- Another command to use is `show ip ssh`
  - If your device doesn't support SSH, it will tell you here
  - In this case, SSH is supported, but disabled
  - Notice the version is 1.99, meaning it supports both versions 1 and 2

- Now, here's a hint about the first step in configuring SSH: please create RSA keys to enable SSH
- RSA keys are cryptographic keys that are essential for the various security features of SSH

Generating RSA Keys for SSH



- After ensuring that the IOS image you're using supports SSH, you must generate those RSA keys
- The keys are used for data encryption and decryption, authentication, etc.

Here's how to do that:

- First, the professor configured the domain name of SW1 with `ip domain name jeremysitlab.com`
  - The reason for this is that the FQDN of the device is used to name the RSA keys
  - By the way, FQDN means **Fully Qualified Domain Name**, which is the host name plus the DNS domain name

```
SW1(config)# ip domain name jeremysitlab.com
```

- Then the professor generated the RSA keys
  - The command is `crypto key generate rsa`
  - Here it shows the name of the keys, SW1.jeremysitlab.com, which is the FQDN of SW1
  - Then you have to choose the size of the modulus, the size of the keys
  - The professor configured 2048 bits

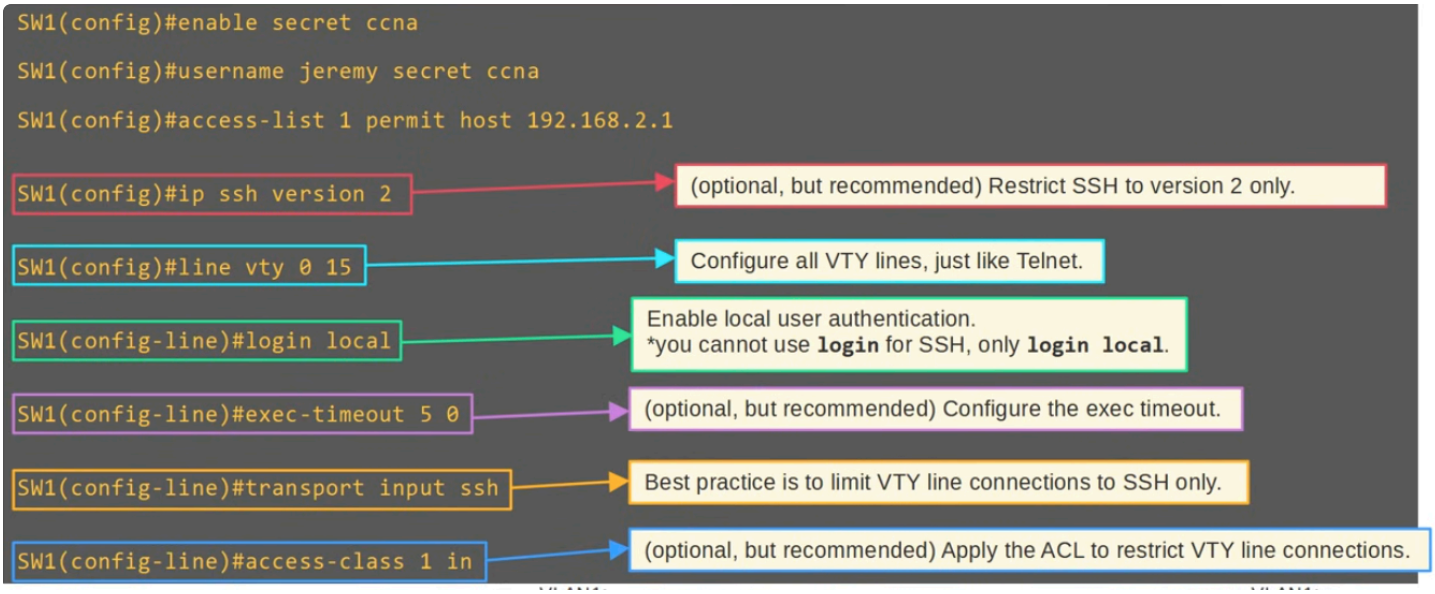
```
SW1(config)# crypto key generate rsa
```

- Note: You can just use the command `crypto key generate rsa modulus`, and then specify the length, without having to specify it separately like this — the effect is the same
- Note: The length must be **768 bits or greater** to use SSHv2, so make sure the keys are at least that length
- Greater key lengths are more secure, but take longer to generate and use
- After the keys are generated, a Syslog message is displayed, indicating that SSH has been enabled
- The professor checked `show ip ssh` again, and indeed SSH has been enabled

Verification:

```
SW1# show ip ssh
```

SSH Configuration



- Now that SSH is enabled, configure it
- The Telnet configurations on SW1 have been removed, so this is a clean configuration of SSH

Steps:

- First, just like before, configure the enable secret, a username, and an ACL to restrict connections to only allow PC2
- Then use the command `ip ssh version 2`
  - This is optional, but recommended because SSH version 2 is more secure

```
SW1(config)# ip ssh version 2
```

- Then once again, use the command **`line vty 0 15`** to configure all 16 VTY lines, just like when configuring Telnet

```
SW1(config)# line vty 0 15
```

- Then enable local authentication
  - Note: You can't use **`login`** for SSH, only **`login local`** works, a username is needed
  - You can also authenticate to an authentication server, but that's a topic for another video

```
SW1(config-line)# login local
```

- Then, since the Telnet configurations were removed, configure the exec timeout again
  - This is optional because there will most likely be a default exec timeout, but you can use this command to specify the timeout you want

```
SW1(config-line)# exec-timeout 5 0
```

- Then use **`transport input ssh`**
  - Best practice is to limit VTY line connections to SSH only, disabling Telnet because it's less secure

```
SW1(config-line)# transport input ssh
```

- Then finally, apply the ACL
  - Just like for Telnet, this is optional, but it is recommended because it adds further security

---

SSH Configuration Process Summary

1) Configure host name
2) Configure DNS domain name
3) Generate RSA key pair
4) Configure enable PW, username/PW
5) Enable SSHv2 (only)
6) Configure VTY lines

```
Router(config)#crypto key generate rsa

% Please define a hostname other than Router.

Router(config)#hostname R2

R2(config)#crypto key generate rsa

% Please define a domain-name first.

R2(config)#ip domain name jeremysitlab.com

R2(config)#crypto key generate rsa
The name for the keys will be: R2.jeremysitlab.com
[output omitted]
```

Connect: **ssh -l** *username ip-address* OR **ssh** *username@ip-address*

- Here's a summary of the SSH configuration process

Step 1: Configure the Device Host Name

- First, configure the device host name
- The professor didn't mention this previously because he had already configured the host name
- The device cannot generate its RSA keys without a non-default hostname

- Example: On a router with the default host name, the professor tried the `crypto key generate rsa` command
  - He got a message saying, "please define a hostname other than router"
  - So he did that
  - Then he tried to generate the RSA key pair again
  - However, he hadn't defined a domain name yet, so he got a message telling him to do so

Step 2: Configure a DNS Domain Name

- That's the next step in SSH configuration — configure a DNS domain name
- So that's what the professor did:

```
R1(config)# ip domain name jeremysitlab.com
```

- Then he tried to generate the RSA key pair, and the command works
- Remember: To generate the RSA key pair you need to configure the host name and the domain name first
- Actually, you can manually specify a name for the key pair, but for the CCNA you just have to know this method

Step 3: Generate the Key Pair

- So, that's the next step — generate the key pair

Step 4: Configure an Enable Password and Local Username/Password

- Then the next step is to configure an enable password and local username and password combination
- The order of this step doesn't matter, and it's not an SSH-specific configuration step, but make sure they are configured

Step 5: Enable SSH Version 2 Only

- Then enable SSH version 2 only

- This isn't necessary, but it is best practice, so the professor decided to include it in this summary

Step 6: Configure the VTY Lines

- Then configure the VTY lines
- The most important one is to make sure `transport input ssh` is enabled
- Then you can do any other VTY line configurations you want, like exec timeout, applying an ACL, etc.
- And that's it, SSH should be working
- From a PC you can connect using the command `ssh -l`, followed by the username and IP address, or `ssh username@IP address`
- You can try that out in the practice lab
- Make sure you do the practice lab — you can make your own lab or use the professor's lab file
- You need to know how to configure SSH, so make sure you practice it

---

New Commands Summary

- Here's a summary of the new commands in this lecture
- Like the professor said, you'll definitely want to do some labbing in Packet Tracer to get used to SSH configuration
- Unlike Syslog and SNMP, SSH configuration is a CCNA exam topic
- If you don't remember the purpose of any of these commands, go back in the lecture to review

Review Before the Quiz

- Before the quiz, here's a review of what we covered

SSH & Remote Access Commands — Quick Reference

Command	What it does
`show version`	Shows IOS version, uptime, hardware, and device information.
`show ip ssh`	Shows whether SSH is enabled, SSH version, and settings.
`ip default-gateway <IP>`	Sets the default gateway on a Layer 2 switch for reaching other networks.
`line con 0`	Enters configuration for the physical **console line**.
`line vty 0 15`	Configures the **VTY lines** used for remote SSH/Telnet access.
`crypto key generate rsa`	Generates RSA keys required for SSH.
`ip ssh version 2`	Forces the device to use the more secure **SSH version 2**.
`login`	Requires the configured line password for login.
`login local`	Uses the device's **local username/password database** for login.
`transport input ssh`	Allows only **SSH** remote connections.
`transport input telnet`	Allows only **Telnet** connections.
`transport input ssh telnet`	Allows both SSH and Telnet.
`transport input none`	Blocks remote access on those VTY lines.

`exec-timeout <min> <sec>`	Automatically logs out an inactive session after the specified time.
`access-class <ACL> in`	Uses an ACL to control **which source IP addresses can remotely access** the device.
`telnet <IP>`	Starts a Telnet connection to another device.
`ssh -l <username> <IP>`	Starts an SSH connection using the specified username.
`ssh <username>@<IP>`	Another syntax for connecting through SSH.

Typical secure SSH configuration

```
username admin secret Cisco123

ip domain-name example.com
crypto key generate rsa
ip ssh version 2

line vty 0 15
 login local
 transport input ssh
 exec-timeout 10 0
```

The main CCNA idea is:

```
Console → Local physical access
VTY     → Remote access
Telnet  → Remote + unencrypted
SSH     → Remote + encrypted
RSA     → Required to enable SSH
```

For real networks, **SSH is preferred over Telnet** because SSH encrypts the session.

Review Before the Quiz

- First, console port security
  - By default, anyone who has physical access to the console port can access the CLI of the device
  - So, it's a good idea to configure some security on it
- Then the professor introduced the concept of a Layer 2 switch management IP
  - Layer 2 switches can't route packets, but they can still send traffic from and receive traffic on an SVI
  - This allows them to respond to pings, received connections via SSH, etc.
- Then the professor introduced Telnet, a protocol that allows you to connect to the CLI of a device
  - However, Telnet is old and not secure
  - So, in modern days we typically use SSH, Secure Shell, instead when we remotely configure devices