LABORATORIO 5

Configuración de contraseñas seguras y habilitar SSH

Introducción

En este laboratorio se repasan conceptos importantes de laboratorios anteriores y se avanza en nuevos temas. Se sigue la dinámica en donde se explican los conceptos teóricos necesarios antes de mostrar los comandos, y luego se desglosa cada línea de comando con una explicación detallada, pensando en quién nunca antes configuró un dispositivo Cisco.

Objetivos del laboratorio

- Diferenciar entre `enable password` y `enable secret` y por qué usar `enable secret`.
- Crear usuarios locales con contraseñas seguras usando `username … secret …`.
- Configurar `service password-encryption` y entender sus limitaciones.
- Generar claves RSA y habilitar SSH version 2 en un dispositivo Cisco.
- Restringir el acceso remoto a SSH y deshabilitar Telnet en las líneas VTY.
- Verificar la configuración y solucionar errores básicos de conexión.

Conceptos importantes

¿Qué es una contraseña "segura"?

Una contraseña segura tiene:

- Longitud suficiente (ideal $\geq$ 12 caracteres).
- Mezcla de mayúsculas, minúsculas, números y símbolos.
- No está basada en palabras del diccionario ni en datos personales.
- Se cambia regularmente y no se reusa entre dispositivos críticos.

Tipos de contraseñas en IOS y su almacenamiento

Texto plano

Si en la configuración la contraseña aparece en texto claro, cualquiera con acceso a la configuración la podrá ver. Evitarlo.

- `service password-encryption`: cifra las contraseñas con un hash tipo 7 (una ofuscación reversible, por lo tanto, débil). Evita mostrar en

texto claro cuando se usa el comando `show running-config`, pero no es criptografía fuerte. Es útil como mínima ofuscación, no como protección criptográfica real.

- `enable secret` y `username … secret …`: almacenan la contraseña cifrada con algoritmos de hash (en muchas versiones aparece como tipo 5 o tipo 8/9 en IOS moderno). Esto es significativamente más seguro que tipo 7. Siempre preferir `secret` sobre `password`.

SSH vs Telnet

Telnet

Transmite los datos y las contraseñas en texto claro. Es vulnerable a sniffing (captura de paquetes).

SSH

Cifra la sesión (confidencialidad e integridad), evita que un atacante lea credenciales o comandos. Debe usarse en entornos productivos para administración remota. SSH usa un par de claves (pública/privada) basadas en criptografía asimétrica (RSA, por ejemplo). El dispositivo debe tener generadas sus claves RSA para aceptar conexiones SSH.

Claves RSA

SSH utiliza claves RSA para iniciar una sesión segura. Se generan en el propio equipo con el comando `crypto key generate rsa`. Se selecciona el tamaño de la clave (módulo), por ejemplo 1024, 2048 o 4096 bits. 2048 es una buena práctica mínima hoy en día.

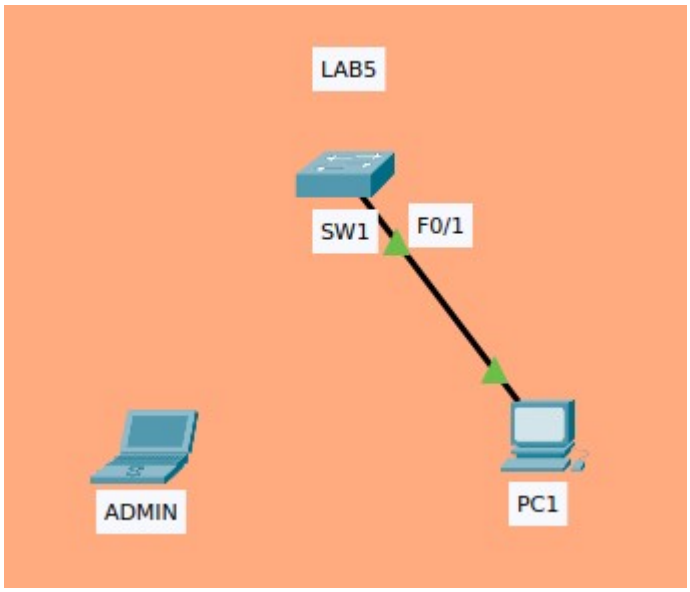
Líneas VTY

Las líneas VTY (`line vty 0 4`) son las que controlan las conexiones remotas, ya sea mediante Telnet o SSH. En ellas se configura qué protocolos se acepta (`transport input ssh/telnet`) y cómo realizar la autenticación (`login local`, `login` o AAA). El número de conexiones VTY permitidas puede variar en cada equipo.

Laboratorio en Packet Tracer

Topología

Se configura un switch (SW1) para su administración desde la red 192.168.1.0/24. La PC de testeo (PC1) tendrá la IP 192.168.1.10, la cual ya fue configurada.



Actividades

1) Desde la PC ADMIN, conectarse al switch mediante cable consola y acceder al modo exec privilegiado

```
Switch> enable
Switch# configure terminal
Switch(config)#
```

Explicación:

- `enable`: eleva del modo usuario (`Switch>`) al modo privilegiado (`Switch#`) donde se pueden ejecutar configuraciones.

- `configure terminal`: entra al modo de configuración global, desde aquí se hacen los cambios que afectan al dispositivo.

2) Establecer hostname y dominio (necesarios para SSH)

```
Switch(config)# hostname SW1
SW1(config)# ip domain-name redes.local
```

Explicación:

- `hostname SW1`: cambia el nombre del equipo, útil para identificar el dispositivo en la CLI, en logs y en la red.

- `ip domain-name redes.local`: define el dominio del equipo. SSH y la generación de claves RSA usan el nombre de dominio para crear el nombre de la clave pública. **Este paso es obligatorio** antes de generar las claves RSA.

3) Crear un usuario local con contraseña cifrada

```
SW1(config)#username admin privilege 15 secret C0ntr@s3n4
```

Explicación:

- `username admin`: crea un usuario llamado **admin**.

- `privilege 15`: asigna privilegios completos (**nivel 15**) al usuario `admin`, con esto, al autenticarse tendrá acceso privilegiado. (Es opcional, ya que si se omite, el usuario obtiene el nivel por defecto).

- `secret C0ntr@s3n4`: guarda la contraseña **cifrada** (no en texto claro).

Nota: usar `secret` siempre que sea posible. Evitar `username … password …` que puede guardar en texto plano si no se combina con cifrados.

4) Configurar contraseña para ingresar al modo privilegiado

```
SW1(config)# enable secret 3ntr4Priv1l3g10
```

Explicación:

- `enable secret 3ntr4Priv1l3g10`: establece la contraseña que se solicitará cuando alguien ejecute `enable` desde modo usuario. Se guarda de forma segura (hasheada). Es el modo recomendado para proteger acceso al modo privilegiado.

5) Ofuscar contraseñas

```
SW1(config)# service password-encryption
```

Explicación:

- `service password-encryption`: cifra las contraseñas de tipo texto plano en la configuración (type 7).

Nota: esto no es fuerte, protege contra miradas casuales, pero no reemplaza `secret` ni políticas reales de seguridad.

6) Generar claves RSA para SSH

```
SW1(config)# crypto key generate rsa modulus 2048
```

Explicación:

– `crypto key generate rsa`: genera un par de claves RSA para SSH.

– `modulus 2048`: es el tamaño de la clave en bits, 2048 es un valor razonable y seguro hoy en día. Si el comando pide confirmación, contestar `yes` o `y`.

Nota: si aparece un aviso de que la clave ya existe, puede ser necesario borrar la anterior o confirmar la sobreescritura. Si el dispositivo no acepta `modulus` en la misma línea, puede pedir el tamaño después del comando, aparecerá el siguiente texto: **"**`How many bits in the modulus [512]: 2048`**"**.

7) Forzar uso de SSH versión 2

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

Explicación:

– `ip ssh version 2`: fuerza que el servidor SSH use la versión 2 del protocolo SSH, que es estándar y más segura que la versión 1. Es una buena práctica siempre usar la v2.

8) Configurar parámetros de SSH

```
SW1(config)# ip ssh time-out 60
SW1(config)# ip ssh authentication-retries 3
```

Explicación:

– `ip ssh time-out 60`: reduce el tiempo (en segundos) que SSH espera para realizar una autenticación, si no se realiza la autenticación en ese período de tiempo, cierra la conexión.

– `ip ssh authentication-retries 2`: limita la cantidad de intentos de autenticación, 3 es un valor razonable. Evita ataques de fuerza bruta.

Nota: los nombres exactos y disponibilidad de los comandos pueden variar según versión de IOS.

9) Configurar la IP de gestión del switch

```
SW1(config)# interface vlan 1
SW1(config-if)# ip address 192.168.1.1 255.255.255.0
SW1(config-if)# no shutdown
SW1(config-if)# exit
```

Explicación:

– `interface vlan 1`: entra a la interfaz lógica de gestión (**SVI**), por defecto en switches Cisco es la interfaz VLAN 1.

– `ip address 192.168.1.1 255.255.255.0`: asigna dirección IP a la interfaz de gestión del switch para que sea accesible en la red.

– `no shutdown`: habilita la interfaz.

– `exit`: vuelve al modo de configuración global.

Nota: en un entorno real es preferible crear una VLAN de gestión dedicada (**no usar la VLAN 1**) y aplicar controles de acceso. Para laboratorios se usa la VLAN1 por simplicidad.

10) Configurar líneas VTY para aceptar solo SSH y autenticación local

```
SW1(config)# line vty 0 15
SW1(config-line)# transport input ssh
SW1(config-line)# login local
SW1(config-line)# exec-timeout 5 0
SW1(config-line)# logging synchronous
SW1(config-line)# exit
```

Explicación:

– `line vty 0 15`: entra a la configuración de las primeras 16 líneas VTY (conexiones remotas simultáneas 0 a 15). La cantidad de líneas puede variar de dispositivo a dispositivo.

– `transport input ssh`: solo permite SSH como método de acceso, esto bloquea Telnet.

– `login local`: indica que la autenticación debe realizarse con los usuarios locales configurados (`username … secret …`).

– `exec-timeout 5 0`: define tiempo de inactividad (minutos segundos) antes de cerrar la sesión, en este caso 5 minutos y 0 segundos para proteger sesiones abandonadas.

– `logging synchronous`: evita que mensajes del sistema (o logs) interrumpan la línea en donde se está tipeando (mejora la experiencia en CLI).

– `exit`: sale de la subconfiguración de línea.

11) Bloquear intentos fallidos de login (opcional)

```
SW1(config)# login block-for 60 attempts 3 within 60
```

Explicación:

- `login block-for 60 attempts 3 within 60`: si hay 3 intentos fallidos en 60 segundos, el sistema bloqueará nuevos intentos durante 60 segundos. Es una medida de mitigación contra ataques de fuerza bruta.

Nota: este comando puede variar según la versión de IOS.

12) Banner de aviso legal (Opcional)

```
SW1(config)# banner motd #Acceso autorizado solo para
personal de TI. Toda actividad queda registrada.#
```

Explicación:

- `banner motd`: muestra un mensaje (Message Of The Day) a quien se conecta. Util para advertir que sólo el personal autorizado debe acceder y que las sesiones se registran.

13) Prueba de conectividad desde la PC1

```
PC1:\>ping 192.168.1.1

Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.1.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms


PC1> ssh -l admin 192.168.1.1
Password: (C0ntr@s3n4)

Acceso autorizado solo para personal de TI. Toda actividad
queda registrada.

SW1>enable
Password: (3ntr4Priv1l3g10)
SW1#
```

Explicación:

– `ssh`: comando para iniciar la conexión SSH desde un cliente.

– `-l admin`: se usa para indicar el nombre de usuario, usuario `admin` en este caso.

Nota: El cliente preguntará si acepta la huella de la clave del servidor, responder `yes` la primera vez. Luego pedirá la contraseña (la `secret` configurada).

Comandos de Verificación

- `show running-config`: Muestra la configuración actual. Información como: username, enable secret, ip domain-name, line vty y transport input ssh.
- `show ip interface brief`: Verifica que la interfaz VLAN de gestión esté up y tenga IP.
- `show ip ssh`: Muestra si SSH está habilitado, la versión utilizada y parámetros de tiempo. Salida de ejemplo:

  ```
  SSH Enabled - version 2.0
  Authentication timeout: 60 secs; Authentication retries: 3
  ```

  Quiere decir que SSH v2 está activo, espera 60s para la autenticación y permite 3 reintentos.
- `show users`: Lista las sesiones activas y sus origenes. Útil para ver conexiones SSH actuales.
- `show crypto key mypubkey rsa`: Muestra la(s) clave(s) RSA pública(s) generadas en el equipo.
- `show ip ssh` y `show running-config | section line vty`: Comandos útiles para confirmar que `transport input ssh` y `login local` estén correctamente aplicados.

Errores comunes y soluciones rápidas (troubleshooting)

- **No se acepta crypto key generate rsa**: Verificar que `ip domain-name` y `hostname` esten configurados, la generación de la llave usa el nombre del host y el dominio.
- **Conexion SSH falla: "Connection refused"**
  - Revisar `show ip ssh` para ver si SSH está habilitado.

    - Revisar `show running-config | section line vty` para confirmar `transport input ssh` y `login local`.
    - Verificar que la dirección IP del equipo sea la correcta.
- **Contraseña no funciona**
    - Asegurarse de usar el usuario correcto (admin) y la contraseña que se ingresó con `username … secret …`.
    - Revisar `show running-config | include username` para ver qué usuarios existen (no se verá la contraseña en texto claro si se usó `secret`).
- **Se solicita Telnet en lugar de SSH**: Comprobar `transport input`, si contiene `telnet` o `all`, cambiarlo a `ssh`.

Revisión de conceptos importantes

- Siempre usar `enable secret` y `username … secret …`, y nunca exponer contraseñas en texto claro.
- Usar SSH versión 2, es la opción más segura: `ip ssh version 2`.
- Generar claves RSA con al menos 2048 bits para buena protección.
- Bloquear Telnet en `line vty` con `transport input ssh`.
- Usar `login local` para autenticar con usuarios.
- No confiar en `service password-encryption` como única protección, usarla sólo para ofuscación mínima.
- Habilitar tiempos de expiración (`exec-timeout`) y limitar reintentos para reducir vectores de ataque.

Conclusión

En este laboratorio, se aprendió no sólo qué comandos ejecutar para proteger la administración de los dispositivos Cisco sino por qué cada comando es importante. Se explicó en detalle la diferencia entre tipos de contraseñas, la necesidad de SSH sobre Telnet, la generación de claves RSA y las configuraciones de las líneas VTY.

Estas acciones (`enable secret`, `username … secret …`, `crypto key generate rsa`, `ip ssh version 2`, y `transport input ssh`), constituyen la base para administrar dispositivos de forma segura. En entornos reales, estas medidas se deben complementar con AAA centralizado, ACLs de gestión, logging y políticas operativas.