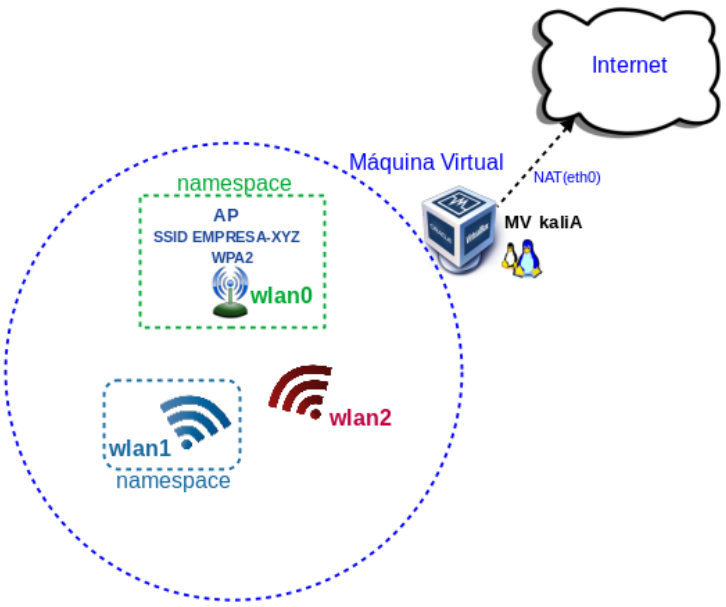
TALLER HE

PRÁCTICA Auditar contrasinal Wi-Fi WPA2 (EAP)

Apelidos	Nome

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

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 (freeradius, hostapd, ip netns)
  - wlan1 → Cliente
  - wlan2 → unknown (tools suite: aircrack-ng →auditar handshake)
- **Namespaces**
  - phy0 → wlan0 → AP aillado
  - phy1 → wlan1 → cliente autenticado aillado



Material necesario	Práctica: Auditar contrasinal Wi-Fi WPA2 (EAP)
■ Host alumnado ■ Máquina virtual GNU/Linux Kali ■ [0] 1-Taller-HE-Practica-WIFI-1 ■ [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) ■ Montar AP simulado e aillar ■ Conectar co cliente simulado e aillar ■ Investigar co cliente unknown(airodump-ng) →desconectar cliente simulado → capturar handshake →comprobar fortaleza contrasinal → auditar handshake con aircrack-ng e ataque por diccionario (rockyou).

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

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 [3]
v. Rede: 1 tarxeta en modo NAT *(as wlanX serán simuladas por mac80211_hwsim)*
vi. Nome: Practica-Kali-Auditar-EAP

(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 (EAP) (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 freeradius hostapd iw iproute2` #Actualizar o listado de paquetes dispoñibles nos repositorios configurados no sistema e, se esta operación remata correctamente, instala automaticamente os paquetes necesarios para o laboratorio Wi-Fi. En concreto:

- **freeradius**: servidor RADIUS empregado para realizar a autenticación centralizada de usuarios mediante EAP/802.1X (WPA2-Enterprise).
- **hostapd**: daemon que permite configurar o sistema como punto de acceso Wi-Fi, xestionando a autenticación e o cifrado das conexións sen fíos.
- **iw**: ferramenta de baixo nivel para xestionar dispositivos Wi-Fi (radios phy, interfaces, asignación a namespaces, etc.).
- **iproute2**: conxunto de utilidades avanzadas de rede (ip, ss, etc.) necesarias para crear e xestionar *network namespaces*, interfaces e rutas.

O parámetro **-y** fai que se acepten automaticamente todas as preguntas durante a instalación, permitindo executar o comando sen intervención manual.

`# 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.

`# 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_ap_wlan0 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 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.

`# 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_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.

`# ip link set lo up` #Activar a interface **loopback (lo)** dentro do namespace **wifi_ap_wlan0**. Isto é imprescindible para que os servizos que usan 127.0.0.1 (como FreeRADIUS e hostapd) poidan comunicarse correctamente dentro do mesmo namespace.

`# echo 'ana Cleartext-Password := "1234"' >> /etc/freeradius/3.0/users` #Engadir un usuario de proba chamado **ana** coa contrasinal **1234** ao ficheiro de usuarios de FreeRADIUS. Este usuario empregarase para probar a autenticación WPA2-Enterprise mediante EAP.

`# cd /etc/freeradius/3.0/certs` #Cambiar o directorio de traballo ao cartafol **certs de FreeRADIUS**, onde se atopan os ficheiros de configuración e as regras para xerar os certificados dixitais (CA e certificado do servidor) usados na autenticación EAP/PEAP.

`# make clean` #Eliminar todos os certificados e ficheiros criptográficos xerados previamente no directorio certs. Este paso garante que non queden restos de configuracións anteriores e que os certificados se rexeneren dende cero.

`# make` #Xerar novamente a infraestrutura de certificados de FreeRADIUS: a Autoridade Certificadora (CA), o certificado do servidor RADIUS e as claves privadas asociadas. Estes certificados son necesarios para establecer o túnel TLS durante a autenticación PEAP.

`# cd` #Voltar ao directorio persoal do usuario actual. Este comando non afecta á configuración, simplemente restaura un directorio de traballo neutro para continuar cos seguintes pasos.

`# chown -R freerad:freerad /etc/freeradius/3.0/certs` #Axustar os permisos dos certificados para que pertenzan ao usuario e grupo **freerad**, garantindo que FreeRADIUS poida acceder a eles sen erros de permisos.

`# NUM=$(cat -n /etc/freeradius/3.0/mods-enabled/eap | grep default_eap_type | head -1 | awk '{print $1}')` #Obter o **número de liña** da primeira aparición da directiva default_eap_type no ficheiro de configuración do módulo EAP de FreeRADIUS. Este valor gárdase na variable **NUM** para poder modificar esa liña exacta de forma automática e controlada.

`# sed -i "${NUM}s/default_eap_type = md5/default_eap_type = peap/" \`
`/etc/freeradius/3.0/mods-enabled/eap` #Modificar directamente (-i) a liña identificada anteriormente para substituír o método EAP por defecto **md5** por **peap**.
Con isto establécese **PEAP como método EAP principal**, requisito imprescindible para WPA2-Enterprise con autenticación baseada en usuario e contrasinal mediante túnel TLS.

```
# 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 #Actualizar as rutas dos ficheiros criptográficos usados por FreeRADIUS (clave privada, certificado do servidor e CA), mantendo a indentación orixinal do ficheiro de configuración. Isto garante unha configuración TLS coherente e válida para autenticación PEAP.

# freeradius -Cx #Validar a configuración completa de FreeRADIUS sen iniciar o servizo. Este comando comproba erros de sintaxe, rutas incorrectas ou problemas de certificados. Se a configuración é correcta, remata sen erros.

# freeradius -fxx & #Executar FreeRADIUS en primeiro plano e con nivel máximo de depuración (-fxx) dentro do namespace wifi_ap_wlan0. Isto permite observar en tempo real todo o proceso de autenticación EAP/PEAP, incluíndo solicitudes RADIUS, negociación TLS e validación de credenciais. O símbolo & envía o proceso a segundo plano mantendo o servizo activo no namespace. FreeRADIUS non debe iniciarse con systemctl neste escenario, xa que iso faría que se executase no namespace principal do sistema. Para que auth_server_addr=127.0.0.1 funcione correctamente, FreeRADIUS e hostapd deben executarse no mesmo network namespace.

# cat > wpa-eap.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
auth_server_addr=127.0.0.1
auth_server_port=1812
auth_server_shared_secret=testing123
EOF

#Este comando crea un ficheiro de configuración chamado wpa-eap.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 radioeléctrico no que emite o punto de acceso. O canal 6 pertence á banda de 2,4 GHz e é un dos canais non solapados máis empregados, o que axuda a reducir interferencias con outras redes Wi-Fi próximas.
hw_mode=g: Establece o modo de hardware como **g** (Wi-Fi 802.11g).
wpa=2: Activa **WPA2** (Wi-Fi Protected Access 2) como protocolo de seguridade.
wpa_key_mgmt=WPA-EAP: Define o método de xestión de chave como **WPA-EAP**, empregando autenticación 802.1X cun servidor **RADIUS**, en lugar dunha clave compartida (PSK).
wpa_pairwise=CCMP: Especifica o algoritmo de cifrado para tráfico unicast en **WPA**. **CCMP** baséase en **AES** e ofrece maior seguridade ca TKIP.
rsn_pairwise=CCMP: Define o algoritmo de cifrado usado no contexto **RSN (WPA2)**. Neste caso indícase **CCMP**, asegurando que as comunicacións cifradas usan **AES**.
beacon_int=100: Establece o intervalo de beacons (paquetes de publicidade de rede) en **100 ms**.
ieee8021x=1: Activa **802.1X** no punto de acceso para que a autenticación dos clientes se faga mediante EAP contra un servidor **RADIUS (WPA-Enterprise)**, en lugar dunha clave compartida.
auth_server_addr=127.0.0.1: Indica o enderezo IP do servidor de autenticación **RADIUS** ao que o punto de acceso enviará as solicitudes de autenticación. O valor **127.0.0.1** refírese á interface loopback (lo) do mesmo network namespace no que se executa hostapd. **Este valor só é válido se hostapd e FreeRADIUS se executan no mesmo namespace.** Se o servidor RADIUS se executa nun namespace distinto, debe empregarse unha dirección IP alcanzable entre namespaces (por exemplo, mediante interfaces veth).
auth_server_port=1812: Especifica o porto UDP no que o servidor RADIUS escoita as peticións de autenticación. O porto 1812 é o porto estándar para RADIUS Authentication.
auth_server_shared_secret=testing123: Define o segredo compartido entre o punto de acceso e o servidor RADIUS. Este valor utilízase para protexer a comunicación RADIUS e debe coincidir exactamente coa configuración do cliente correspondente en FreeRADIUS.

hostapd wpa-eap.conf #Iniciar **hostapd**, o daemon que permite que o sistema actúe como **punto de acceso Wi-Fi**. Ao executar este comando, hostapd carga o ficheiro de configuración **wpa-eap.conf** para configurar o punto de acceso empregando **WPA2-Enterprise**, con autenticación **802.1X/EAP** contra un servidor **RADIUS**. Os parámetros definidos no ficheiro especifican, entre outros, a interface de rede, o nome da rede (SSID), o método de xestión de chaves (WPA-EAP), os algoritmos de cifrado (CCMP/AES) e os datos de conexión co servidor FreeRADIUS. Como resultado, o dispositivo pasa a funcionar como punto de acceso Wi-Fi e acepta conexións de clientes autenticados mediante credenciais de usuario, en lugar dunha clave compartida.

```
wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
wlan0: RADIUS Authentication server 127.0.0.1:1812
wlan0: interface state COUNTRY_UPDATE->ENABLED
wlan0: AP-ENABLED
```

(e) Configura wlan1 como cliente autentificado 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 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=WPA-EAP
eap=PEAP
identity="ana"
password="1234"
phase2="auth=MSCHAPV2"
ca_cert="/etc/freeradius/3.0/certs/ca.pem"
pairwise=CCMP
group=CCMP
}' >> wpa_supplicant.conf
```

#Este comando engade unha definición de rede ao ficheiro **wpa_supplicant.conf**, que será empregada polo cliente Wi-Fi para conectarse á rede WPA2-Enterprise. Así:

ssid="EMPRESA-XYZ": Define o nome da rede Wi-Fi á que se conectará o cliente.

key_mgmt=WPA-EAP: Indica que a xestión de chaves se realiza mediante WPA-EAP (802.1X), empregando un servidor RADIUS, e non mediante unha clave compartida.

eap=PEAP: Especifica que o método EAP empregado será PEAP, que crea un túnel TLS para protexer as credenciais do usuario.

identity="ana": Nome de usuario que se enviará ao servidor RADIUS para a autenticación.

password="1234": Contrasinal asociado ao usuario definido en FreeRADIUS.

phase2="auth=MSCHAPV2": Define o método de autenticación interno usado dentro do túnel PEAP (MSCHAPv2).

ca_cert="/etc/freeradius/3.0/certs/ca.pem": Indica o certificado da Autoridade Certificadora (CA) que se empregará para verificar a identidade do servidor RADIUS durante o establecemento do túnel TLS.

pairwise=CCMP: Define o algoritmo de cifrado para tráfico unicast, empregando AES-CCMP.

group=CCMP: Define o algoritmo de cifrado para tráfico multicast/broadcast, tamén baseado en AES-CCMP.

`# wpa_supplicant -B -i wlan1 -c wpa_supplicant.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.conf: Indica o ficheiro de configuración que contén os parámetros da rede Wi-Fi (como SSID, clave, etc.), neste caso wpa_supplicant.conf.

```
wlan0: STA 86:07:00:8a:99:6f IEEE 802.11: authenticated

wlan0: STA 86:07:00:8a:99:6f IEEE 802.11: associated (aid 1)

wlan0: CTRL-EVENT-EAP-STARTED 86:07:00:8a:99:6f

wlan0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1

wlan0: CTRL-EVENT-EAP-SUCCESS2 86:07:00:8a:99:6f

wlan0: STA 86:07:00:8a:99:6f WPA: pairwise key handshake completed (RSN)

wlan0: EAPOL-4WAY-HS-COMPLETED 86:07:00:8a:99:6f

wlan0: AP-STA-CONNECTED 86:07:00:8a:99:6f

wlan0: STA 86:07:00:8a:99:6f RADIUS: starting accounting session 561556BBF325882C

wlan0: STA 86:07:00:8a:99:6f IEEE 802.1X: authenticated - EAP type: 25 (PEAP)
```

(f) 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.

`# 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.

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

`# pkill -f wpa_supplicant || true` #Eliminar os procesos `wpa_supplicant` existentes.
`# wpa_supplicant -B -i wlan1 -c wpa_supplicant.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.conf** indica que o ficheiro de configuración a usar é **wpa_supplicant.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.

(h) Consola3. Unha vez capturado o handshake:

`CH  6 ][ Elapsed: 48 s ][ 2026-01-13 17:24 ][ WPA handshake: AA:D9:48:E9:15:1E` ← ***BSSID***

Executar na consola3:

`# Ctrl^C` #Envíar 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 dicionario 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-o1.cap**) almacenado no directorio **capturas**. Este ficheiro contén un **handshake** de WPA/WPA2 RSN, e o comando tenta descifrar a clave de acceso utilizando un dicionario de contrasinais. O dicionario de contrasinais está especificado no ficheiro **rockyou.txt**, que está no directorio **/tmp**. O proceso consistirá en probar cada palabra no dicionario para ver se coincide co contrasinal usado na rede Wi-Fi, permitindo obter a clave se está no dicionario.

No escenario WPA2-EAP, ferramentas clásicas como airodump-ng e aircrack-ng deixan de ser efectivas para obter acceso á rede, xa que non existe unha clave compartida nin un handshake reutilizable. Isto evidencia unha mellora significativa de seguridade fronte a WPA2-PSK e obriga a empregar técnicas máis complexas como Evil Twin, ataques Man In The Middle(MITM)ou explotacións de malas configuracións do cliente.

```
                                Aircrack-ng 1.7

[00:52:06] 14345517/14344392 keys tested (4574.90 k/s)

Time left: -576045712 day, 15 hours, 21 minutes, 36 seconds    100.01%

                            KEY NOT FOUND


Master Key     : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Transient Key  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC     : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

(2) Contrasinal non atopado (non funciona EAP como PSK).

En **WPA2-EAP** o que captura `airodump-ng` **non é un handshake reutilizable** como en WPA2-PSK, senón tráfico **RSN/802.1X (EAPOL)** que **non permite derivar unha clave** para un ataque por dicionario.