Auditoría de redes WiFi

Redes de Ordenadores Móviles - GSyC

Índice

0. Introducción

En esta práctica nos enfrentamos a los siguientes objetivos:

1. Configurar una red inalámbrica 802.11 en modo ad-hoc con cifrado WEP.
2. Interceptar el tráfico de la red inalámbrica con una máquina no asociada a la red.
3. Romper la clave de cifrado de la red inalámbrica.

Para poder interceptar el tráfico se necesita una tarjeta en modo monitor. No todos los drivers soportan este modo, por lo que hay que ser cuidadoso a la hora de escoger hardware y sistema operativo. En esta práctica usaremos máquinas virtuales con BackTrack Linux 5 y NICs (*network interface card*, tarjeta de red) USB con chipset Realtek 8187. Las máquinas correrán sobre la versión gratuita de VirtualBox.

Emplearemos tres máquinas virtuales:

1. Las máquinas ALICE y BOB, que serán las que pertenezcan a la red WiFi en modo ad-hoc. No es necesario que tengan el chipset Realtek 8187, basta con que se puedan poner en modo ad-hoc, por lo que pueden ser dos ordenadores cualquiera. Alguno de ellos puede ser incluso un *smartphone*

2. La máquina SID, que será la que realice la intercepción. Es la única que necesita el modo monitor.

1. Configuración de una red 820.11 ad-hoc con cifrado WEP

1.1. Configuración de la máquina virtual ALICE

1.1.1. Configuración del hardware

1. Iniciar la máquina virtual y hacer login como `root`.
2. Conectar la NIC al host y activar el dispositivo en el menú Dispostivos → Dispositivos USB.
3. Comprobamos con `dmesg` y `lsusb` que la tarjeta es visible. Un bug en VirtualBox hace que sea necesario desconectar y reconectar la NIC al USB.
4. Al conectar la NIC, el kernel automáticamente carga el driver `rtl8187`. Puede comprobarse con `lsmod` que el módulo está cargado. Este driver sirve para usarse en modo monitor, pero no en modo ad-hoc. Para el modo ad-hoc tenemos que reemplazarlo por el `r8187`. Para ello basta con hacer:

   ```
   $ modprobe -r rtl8187
   $ modprobe r8187
   ```

5. Por último, con `iwconfig` podemos comprobar que tenemos disponible una interfaz `wlanX` (esto es, wlan0, wlan1, etc) que representa la nueva NIC.

Opcional: Estos cambios no son permanentes, por lo que si se reinicia la máquina, habrá que volver a sustituir los módulos. Para configurar la carga de módulos de forma permanente basta con:

1. Evitar la carga de `rtl8187` añadiéndolo a la lista negra `/etc/modprobe.d/blacklist`
2. Forzar la carga de `r8187` añadiéndolo a `/etc/modules`

Para comprobar que el hardware está funcionando puede realizarse un escaneo de las redes cercanas:

```
$ iwlist wlanX scan
```

Si este comando arroja resultados, el hardware está funcionando correctamente.

1.1.2. Configuración de la red

Una vez que tenemos la interfaz `wlanX` disponible, debemos configurarla para crear la nueva red ad-hoc. Los datos de configuración necesarios serán:

1. La IP de la máquina en la nueva red. Puede ser del tipo 10.0.0.x
2. El ESSID de la red.
3. La clave WEP, representada en hexadecimal. Utilizaremos una clave de 40 bits, como por ejemplo `1a1b1c1d1e`. Esto es, 10 valores hexadecimales (0,1,...9,A,B,...F)
4. El canal WiFi.

La configuración de las interfaces de red se hace en el archivo `/etc/network/interfaces`. Debemos editarlo de esta manera:

```
auto wlanX
iface wlanX inet static
    address 10.0.0.x
    netmask 255.255.255.0
    wireless-essid <ESSID>
    wireless-mode ad-hoc
    wireless-key <CLAVE>
    wireless-channel <CANAL>
```

Tras ello se levanta la nueva interfaz empleando la orden:

```
$ ifup wlanX
```

IMPORTANTE: Para poner el interfaz en modo Ad-Hoc es necesario emplear el driver `r8187` tiene errores y si se configura una red en modo ad-hoc y se invoca `iwconfig`, puede provocar un kernel panic y habrá que reiniciar la máquina.

1.2. Configuración de la máquina virtual BOB

La configuración de BOB será idéntica a la de ALICE salvo por la dirección IP. Para probar la conectividad de ALICE y BOB basta con hacer ping: si obtenemos respuesta, la red está correctamente configurada. En caso contrario, habrá que revisar la configuración de ambas máquinas y que el hardware está funcionando (`iwconfig wlanX o iwlist wlanX scan`).

En caso de que exista conectividad, habremos conseguido el primer objetivo propuesto en esta práctica.

2. Interceptando tráfico inalámbrico

2.1. Configurando el hardware de SID

La tercera máquina virtual, SID, será la encargada de analizar el tráfico de la red inalámbrica. En este caso podemos usar el driver rtl8187. No será necesario ninguna configuración de IP, ESSID, canal, etc. por lo que no será necesario añadir nada a `/etc/network/interfaces`.

En este caso, configuraremos la NIC en modo monitor. Para ello realizaremos los siguientes pasos:

1. Comprobar que existe una interfaz inalámbrica con el comando `iwconfig`
2. Levantar dicha interfaz inalámbrica con `ifup wlanX`
3. Configurarla en modo monitor con `iwconfig wlanX mode monitor`

Para comprobar que se pueden analizar las redes cercanas puede emplearse una herramienta como Kismet.

2.2. Capturando paquetes

Para capturar los paquetes podemos emplear un sniffer como `airodump-ng`, que se puede iniciar empleando el comando:

```
$ airodump-ng wlanX
```

Esta aplicación realiza un escaneado de los canales WiFi y muestra en pantalla los BSSIDs, ESSIDs y direcciones MAC de los paquetes que intercepta. En esta aplicación debemos ver que existen varias redes alrededor, entre las cuales está la ad-hoc de ALICE y BOB. Apuntaremos el BSSID, las direcciones MAC de los clientes y el número de canal de dicha red. También es interesante comprobar la columna `#Data` que indica el número de paquetes de datos interceptados de cada red.

NOTA: la tecla ESPACIO detiene el refresco de la pantalla para facilitarnos copiar los datos que nos interesen.

Es posible que el número de canal de la red sea diferente al configurado para ALICE y BOB. Esto ocurre cuando el canal configurado está utilizado por una estación cercana y las NIC negocian un cambio de canal a otro con menos interferencias.

Al escanear todos los canales, es de esperar que algunos paquetes se pierdan, por lo que será necesario restringir la captura de paquetes al canal deseado. Para ello detenemos `airodump-ng` pulsando `ctrl-c` y relanzamos la aplicación añadiendo un parámetro adicional:

```
$ airodump-ng --channel <CANAL> wlanX
```

En este caso, `airodump-ng` únicamente escaneará el canal dado, por lo que capturará todo el tráfico de la red a analizar. Se puede comprobar este hecho, por ejemplo, haciendo un ping desde ALICE a BOB y comprobando que se capturan dos paquetes (el `echo request` y el `echo reply`).

En este punto habremos conseguido el segundo objetivo de la práctica.

3. Ruptura de clave WEP

Para romper la clave de cifrado WEP es necesario capturar una gran cantidad de paquetes de datos. Lo primero que haremos será relanzar `airodump-ng` para volcar los paquetes capturados a un archivo:

```
$ airodump-ng --channel <CANAL> -w <ARCHIVO_CAPTURA> wlanX
```

El archivo generado tendrá extensión `.cap` y contendrá todos los paquetes interceptados.

La herramienta `aircrack-ng` se emplea para analizar este archivo de capturas y tratar de obtener la clave WEP de una red. Lo ejecutaremos en paralelo con `airodump-ng` (es decir, en una terminal nueva) con el siguiente comando:

```
$ aircrack-ng <ARCHIVO_CAPTURA>.cap
```

La aplicación preguntará la red que se desea analizar y, tras elegirla, comenzará la ruptura de la clave. Únicamente resta esperar a que la red transmita suficiente cantidad de paquetes de datos como para que `aircrack-ng` pueda obtener la clave.

Para acelerar el proceso, y teniendo el control de las máquinas ALICE y BOB, podemos hacer que circule mucho tráfico entre ellas. Por ejemplo, podemos abrir varias consolas en cada máquina y desde todas ellas hacer *pings*. Concretamente la orden `ping -i 0.2 <MAQUINA>` hace que se envíe un paquete de prueba cada 2 décimas de segundo (en vez de uno por segundo, que es el valor por omisión)

Cuando el número de paquetes sea suficientemente alto, `aircrack-ng` habrá obtenido suficientes IVs para romper la clave, que mostrará por pantalla.