Auditoría WiFi

Departamento de Sistemas Telemáticos y Computación (GSyC)

gsyc-profes (arroba) gsyc.es

Diciembre de 2013

Interfaces WiFi para auditoría

- Las tarjetas WiFi en principio están pensadas para uso *normal* (modo *managed*), esto es, enviar y transmitir datos.
  El fabricante suele incluir drivers para Windows y MacOs, y casi siempre hay drivers para Linux
- Pero para hacer auditoría, es necesario que el driver soporte el modo monitor
  - Normalmente en Windows no es posible
    En Windows se suele usar hardware dedicado, como AirPcap
    También hay hardware independiente aunque mucho más limitado, como Wifi Robin
  - En Linux es relativamente sencillo, hay muchos interfaces que lo permiten (no todos)
  - Un interfaz WiFi en modo promiscuo recibe todo el tráfico con su mismo SSID
    En modo monitor, recibe todo el tráfico, de cualquier SSID

Modo monitor en Linux:

- Es necesario saber cuál es el *chipset* de nuestra tarjeta
  - Si es PCI
    `lspci -vv`
  - Si es USB
    `lsusb -vv`
- Averiguar si está soportado
- Configurarlo, posiblemente cambiando y/o parcheando el driver (suele ser un módulo)

Una buena solución es emplear una distribución de Linux orientada a auditoría, *backtrack*, que incluye muchos drivers listos para funcionar y que puede:

- Instalarse como una distribución ordinaria
- Usarse como *distro live*
- Usarse en una máquina virtual.
  En este caso, para acceder al interfaz de red a bajo nivel es imprescindible que sea USB

- *backtrack* es la distribución que usaremos, es posiblemente la más madura. Desarrollada por empresa norteamericana especializada en formación, *offensive security*
- Pero hay otras *distros* interesantes, como las desarrolladas desde el portal español `seguridadwireless.net`:
  wifislax. Mas pesada. Apareció primero. Basada en backtrack
  wifiway. Más ligera, más reciente, hecha desde cero
  - Incluyen información y herramientas sobre vulnerabilidades en las claves por omisión usadas por proveedores españoles
    P.e. las claves WEP de las redes de telefónica con el nombre `WLAN_XX`
    (`WLAN_XXXX` ya son redes WPA)

En la documentación de aircrack-ng hay mucha información sobre tarjetas soportadas y drivers
`www.aircrack-ng.org`
En nuestras prácticas usaremos interfaces *eRize ERZW54-USB04RS* o bien interfaces *Alfa AWUS036H*

- Llevan el *chipset* realtek RTL8187L
- Pueden usar o bien el módulo (el driver) rtl8187 o bien el r8187
  - El módulo r8187 es más antiguo, algo inestable, no está incluido en las últimas versiones de backtrack. Soporta el modo ad-hoc
  - El módulo rtl8187 es más moderno y estable, es el recomendado para Kismet. Aunque no soporta modo ad-hoc

Incorporan una pequeña antena, reemplazable por cualquier otra con el conector estándar RP-SMA

Dispositivos USB en VirtualBox

VirtualBox puede capturar un dispositivo USB del *host* y conectarlo al *guest*

- Esta función solo la tiene la versión *freeware*, que es la que podemos descargar del web de VirtualBox
- La versión OSE (*Open Source Edition*) no incluye acceso a USB

Para poder conectar un dispositivo USB al *guest*, es necesario que cuando se inicia el *host*, el usuario pertenezca al grupo `vboxusers`

- Lo comprobamos con `id MI_LOGIN`

Conectamos el interfaz al *host*.

- Comprobamos que lo reconoce

```
koji@mazinger:~$ lsusb
Bus 002 Device 003: ID 046d:c315 Logitech, Inc. Classic Keyboard
Bus 002 Device 002: ID 046d:c018 Logitech, Inc. Optical Wheel Mouse
Bus 001 Device 005: ID 0bda:8187 Realtek Corp. RTL8187 Wireless Adapter
```

  Nos fijamos en el número de bus y de dispositivo (*device*)

- Comprobamos que tenemos permiso de lectura y escritura en el fichero `/dev/bus/usb/XXX/YYY`, donde
  XXX:Bus
  YYY:Dispositivo
- Comprobamos que podemos ver el contenido de `/dev/vboxusb`

Hacemos una de estas dos cosas

1. Añadir un filtro a la configuración de la máquina virtual, para que siempre capture el dispositivo
   Desde la ventana principal de VirtualBox
   `detalles | USB`
   Pulsamos el icono que representa un usb con '+' y seleccionamos el dispositivo
2. Desde la ventana de la máquina virtual en VirtualBox
   `dispositivos | dispositivos USB`

Tras esto, veremos el dispositivo en el *guest* con la orden `lsusb`

Carga del módulo

El driver del interfaz normalmente será un módulo (que se carga cuando se necesita, no está integrado en el núcleo)
Para comprobar si el módulo que necesitamos está cargado:
`lsmod | grep 8187`

- Para borrar el módulo cargado
  `modprobe -r <nombre_modulo>`
- Para cargar un módulo nuevo, podemos hacer cualquier de estas dos cosas
  1. `modprobe r8187`
     Esto tiene efecto inmediato, pero desaparece al reiniciar la máquina
  2. Añadimos el nombre del módulo al fichero `/etc/modules`
     Esto es persistente, pero solo tiene efecto tras reiniciar la máquina
- Comprobados que el módulo está cargado

Configuración de los interfaces

Para configurar un interfaz de red en modo Ad-Hoc, editamos `/etc/network/interfaces`

```
auto <interfaz>
iface <interfaz> inet static
address <tu_direccion>
netmask <tu_mascara>
wireless-essid <tu_essid>
wireless-mode Ad-Hoc
```

- Tras editar el fichero, es necesario desactivar y activar el interfaz (o reiniciar el demonio *networking*)
- Para averiguar el nombre del interfaz (p.e. wlan0 o wlan1) y comprobar su estado, usamos las órdenes
  `ifconfig`
  `iwconfig`

Kismet

Para saber qué AP están visibles, podemos usar
`iwlist <interfaz> scan`
Pero Kismet da información mucho más completa. Es un detector de redes, capturador de tramas (*packet sniffer*) y detector de intrusiones para redes IEEE 802.11

- Licencia GNU. Libre y gratuito
- Detector de redes: Captura balizas de los AP
- Captura tramas de WSTA (*packet sniffer*)
  Permite extportar todo el tráfico capturado a tcpdump, Wireshark o Airsnort
- Funciones básicas de detección de intrusiones
- Disponible para Linux y otros Unix. Soporte muy limitado en Windows

- Herramienta madura, muy popular y muy útil, apenas hay alternativas
- En tiempos de Windows XP se usaba NetStumbler. En Windows vista y Windows 7, InSSIDer
- Funcionamiento completamente pasivo (a diferencia de NetStumbler)
- Permite conexión con un GPS, útil para el *wardriving* (Búsqueda de redes desde un vehiculo en movimiento)

Inconvenientes

- Bugs frecuentes (tal vez en los drivers)
- Naturalmente, es necesario poner el interfaz en modo monitor
- ¡No tiene manual de usuario! Solo un *readme*. El significado de cada opción hay que adivinarlo, o buscarlo en algún tutorial no oficial, posiblemente obsoleto
- El interfaz de usuario no es del todo intuitivo (aunque tampoco resulta especialmente complicado)

Estructura de Kismet

Tres elementos, que pueden estar o no en el mismo ordenador

- Drone. Captura el tráfico
- Servidor. Analiza el tráfico
- Cliente. Representa gráficamente la información procesada

Puesta en marcha de Kismet

Usaremos Kismet en BackTrack Linux en una máquina virtual

- Comprobamos que el interfaz es visible en el bus USB
  `lsusb`
- Hay un bug en VirtualBox en el manejo del interfaz USB que se evita
  - Desconectado físicamente el interfaz en el *host*
  - Volviendo a conectar
  - (Comprobamos que vuelve a ser visible)

- Para la comunicación cliente-servidor es necesario que la dirección *localhost* funcione. (Comprobamos que responde al ping, para ello es necesario que el interfaz `lo` esté activo)
- Ponemos el interfaz inalámbrico en modo monitor.
  Un interfaz se puede configurar de dos formas, pero no deben mezclarse
  - Bajo nivel: ifconfig
  - Alto nivel: `/etc/network/interfaces`, ifup, ifdown
    Aquí configuraremos a bajo nivel, así que desactivamos (convirtiendo en comentario) la configuración del interfaz en `/etc/network/interfaces`

```
ifconfig <INTERFAZ> down      # Si estaba activo
iwconfig <INTERFAZ> mode monitor
ifconfig <INTERFAZ> up
```

- Comprobamos con `ifconfig` e `iwconfig` que el interfaz está activo y en modo monitor

- En el fichero de configuración de kismet

```
/etc/kismet/kismet.conf   (Debian, Ubuntu)
/usr/etc/kismet.conf      (Backtrack 4)
/usr/local/etc/kismet.conf      (Backtrack 5)
```

indicamos el interfaz inalámbrico y el driver a emplear (para kismet, mejor el rtl8187, no el r8187)
`ncsource=<INTERFAZ>type=<MODULO>`
ejemplo: `ncsource=wlan0:type=rtl8187`

Hay (al menos) dos formas de lanzar kismet

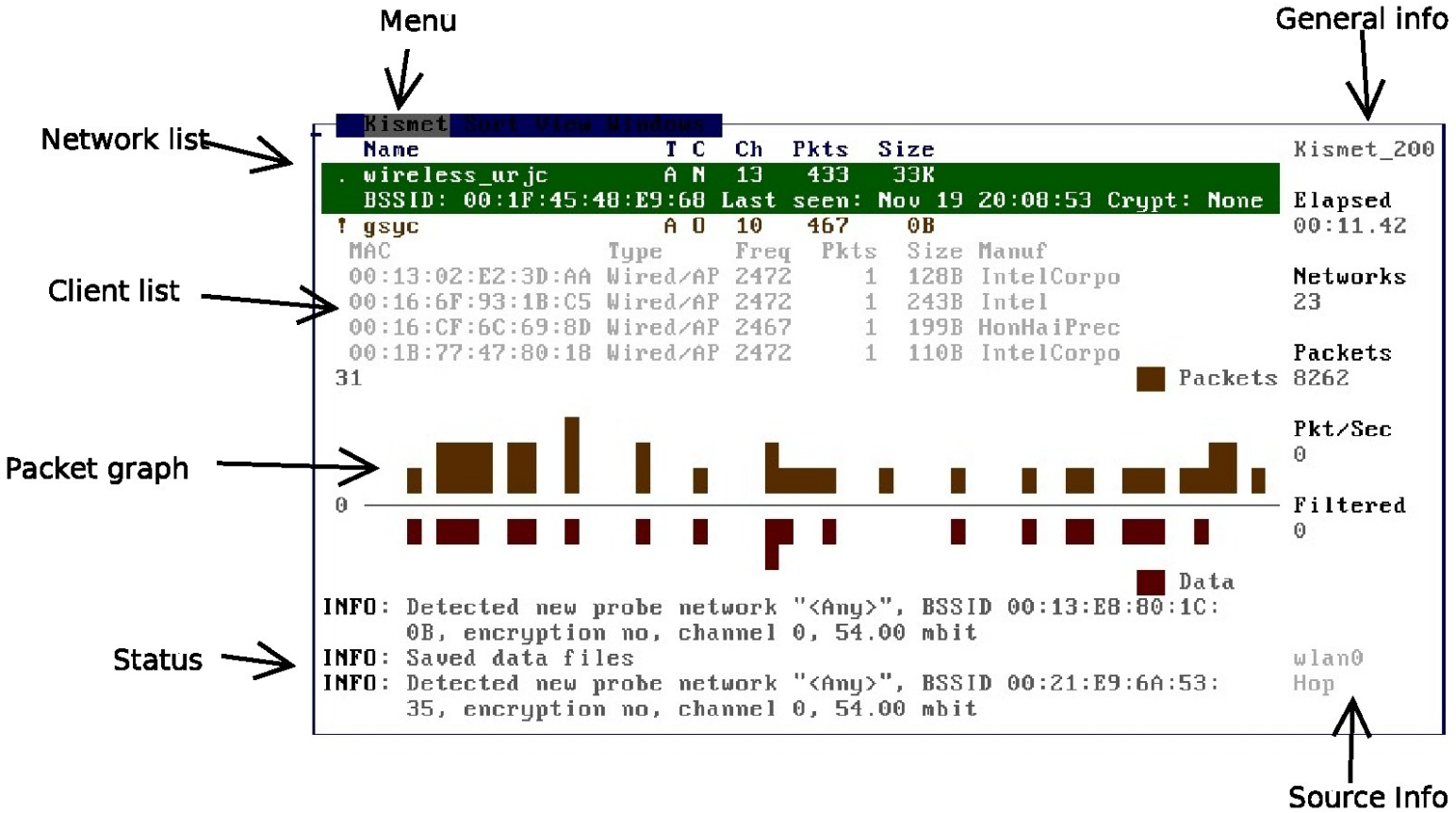
1. Desde una sesión del usuario `root`, ejecutar `kismet`.
   Drone, servidor y cliente pertenecen al `root`. Funciona, aunque advierte sobre posible riesgo
2. Separando el cliente del resto
   - Desde una sesión del usuario `root`, ejecutar `kismet_server`.
     Esto inicia drone y server
   - Desde una sesión de usuario, ejecutar `kismet`

   Este enfoque es más seguro, pero en la versión actual parece haber un bug y se ven las redes pero no los clientes

Interfaz gráfico de Kismet

- El interfaz gráfico nativo está basado en ncurses, una librería para desarrollar interfaces con menús y ventanas sobre terminales de texto
  - Se accede al menú principal con `Esc` o con la virgulilla (`~`), pulsando `AltGr 4`
  - Se elige una opción de un menú con `Alt` + Letra destacada
  - El foco se desplaza de un botón a otro con la tecla `tab`
  - Se hace clic sobre el botón que tiene el foco pulsando la tecla `Intro`

Ventana principal de Kismet

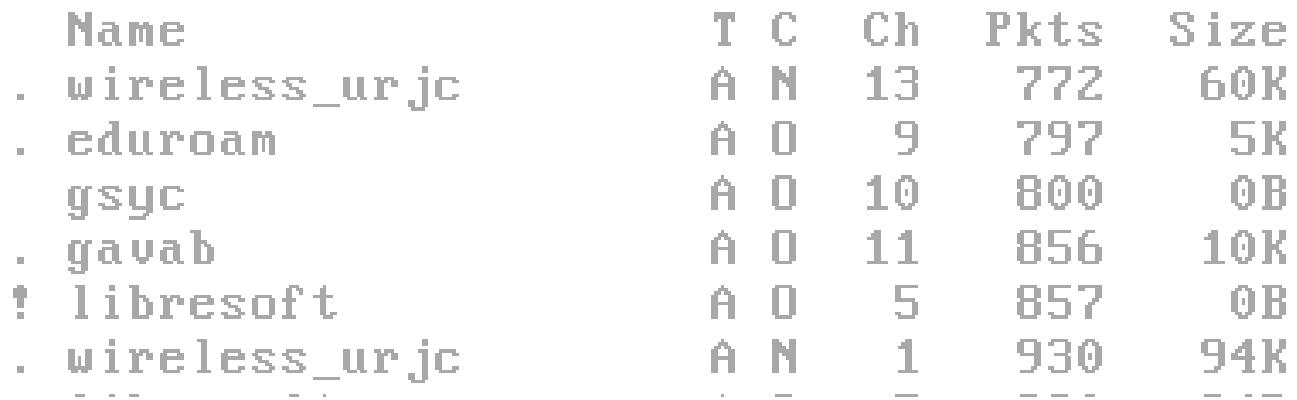


La ventana principal está dividida en paneles

- Network list: Lista de redes detectadas. (Si no caben en pantalla, se usan los cursores arriba/abajo para desplazarse). De esta manera se destaca una red
- Client list: Clientes percibidos en la red destacada. También se pueden usar los cursores. (`Tab` para pasar de un panel a otro)
- General info: Tiempo de funcionamiento, redes y paquetes detectados
- Packet graph: Representación de los datagramas percibidos
- Status: Información de la actividad (Clientes que nuevos, nodos con comportamiento sospechoso...)

El menú *view* de la ventana principal permite mostrar u ocultar paneles (útil con un terminal pequeño)

Network list



- !: Actividad últimos 3 seg .: Actividad últimos 6 seg
- Columna T:
  [A]: Access point
  [H]: Ad-Hoc
  [G]: Grupo de redes wireless
  [P]: *probe request* (tarjeta no asociada a AP)
- Columna C: (Cifrado)
  [Y]: Wep yes
  [N]: Wep no (Red abierta)
  [O]: Other (WPA, WPA2)
- Columna Ch: (Channel)

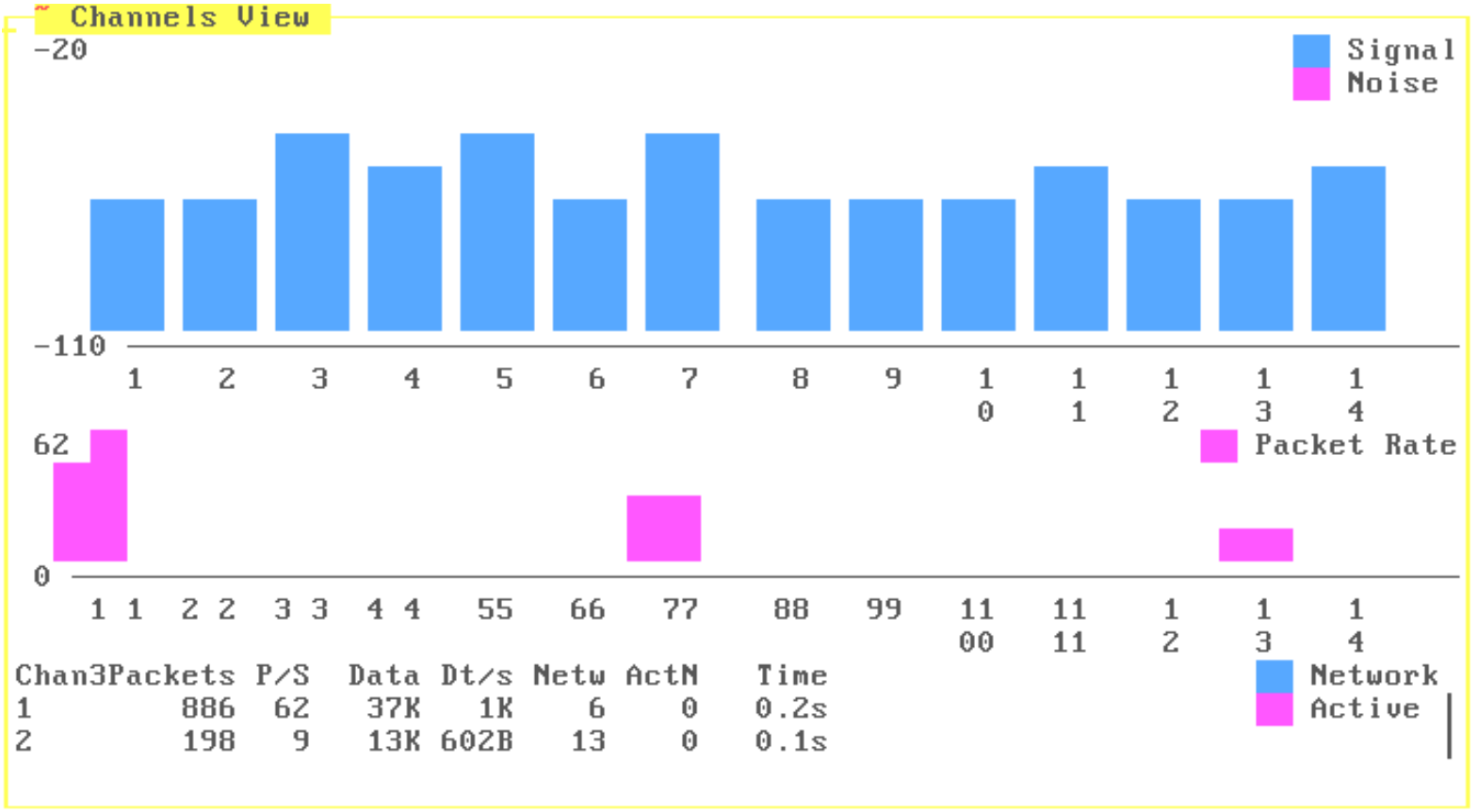
Desde el menú *windows* de la ventana principal, se puede ir a las diferentes ventanas secundarias de Kismet

- Network Details
- Client List
- Network Note. Permite hacer anotaciones sobre las redes
- Channel Details
- GPS Details
- Alerts. Muestra la actividad *sospechosa*. Programable

Para volver a la ventana principal desde una ventana secundaria, hay que elegir en el menú la opción *close window*, que suele estar en el primer menú de la izquierda [1]

[1] Excepto en GPS, donde hay que pulsar OK, y en Network Note, donde hay que pulsar cancelar

Ventana Channels View



La ventana Channels View muestra la distribución del tráfico por canales

En el directorio desde donde se lanzó kismet, se guardan todos los log, en distintos formatos

- dump: Formato similar al .cap de airodump
- network: Archivo de texto con los datos de las redes detectadas
- csv: Formato CSV, importable p.e. desde hojas de cálculo
- xml: Formato XML, metalenguaje estándar muy extendido
- weak: Paquetes débiles detectados (con vulnerabilidad provocada por bug WEP), formato AirSnort
- cisco: Formato CDP (Cisco Discovery Protocol)
- gps: Si hay gps disponible, guarda las coordenadas de las redes

Otros bugs

Con las versiones de Kismet y del driver que usamos actualmente en el laboratorio, también se presentan los siguientes bugs

- Las redes se muestran correctamente. Pero para que muestre los clientes, tenemos que activar en el menú principal de Kismet
  `sort | SSID`
- Kismet se cuelga activando
  `sort | Auto-fit`

Hacking Ético

- *Hack* en inglés ordinario, significa cortar en rodajas, cortar en tajos, trocear, desbrozar...
- A partir de los años 60, en informática, se le da el significado de *truco*: Usar, configurar o programar un sistema para ser usado de forma distinta a la esperada habitualmente
- *Hacker* es una persona que se *cuela* en un ordenador o red de ordenadores. Es una palabra que no tiene una definición universalmente aceptada
  - En el lenguaje, prensa y medios generalistas, *hacker* suele tener connotacciones negativas
  - En la comunidad especializada, para una persona que comete delitos se usa *cracker*
    Si bien cualquiera que accede a un sistema sin autorización, incluso para *echar un vistazo* está causando un daño objetivo, puesto que compromete el sistema, la víctima no conoce el alcance de la invasión y si es responsable, debe ponerse en el peor escenario posible

Clasificación de los hackers

Según su motivación

- *White hat hacker*. Motivación legítima, *hacking* ético. Prueba su propio sistema o el de otro, por el que está contratado, con autorización previa explícita
  - *Red team*: Atacantes
  - *Blue team*: Defensores
- *Grey hat hacker*. Invade un sistema sin autorización del responsable, notifica posteriormente que ha podido burlar la seguridad. Tal vez solicite una cantidad de dinero *razonable*
- *Black hat hacker*. *Cracker*. Delincuente. Actividades de vandalismo, fraude, robo de identidad, piratería...

Según su nivel de conocimientos, en los extremos están

- Elite: Minoría más avanzada que es capaz de descubrir técnicas nuevas
- *Neophyte, noob, newbie*: Principiante
- *Script kiddie*: Principiante que no sabe lo que hace, usa ciegamente aplicaciones desarrolladas por otros sin comprenderlas ni saber adaptarse a un mínimo cambio

Delitos contra el secreto de las comunicaciones

Código penal español:

Artículo 197.1

El que, para descubrir los secretos o vulnerar la intimidad de otro, sin su consentimiento, se apodere de sus papeles, cartas, mensajes de correo electrónico o cualesquiera otros documentos o efectos personales o intercepte sus telecomunicaciones o utilice artificios técnicos de escucha, transmisión, grabación o reproducción del sonido o de la imagen, o de cualquier otra señal de comunicación, será castigado con las penas de prisión de uno a cuatro años y multa de doce a veinticuatro meses.

Artículo 197.2

Las mismas penas se impondrán al que, sin estar autorizado, se apodere, utilice o modifique, en perjuicio de tercero, datos reservados de carácter personal o familiar de otro que se hallen registrados en ficheros o soportes informáticos, electrónicos o telemáticos, o en cualquier otro tipo de archivo o registro público o privado. Iguales penas se impondrán a quien, sin estar autorizado, acceda por cualquier medio a los mismos y a quien los altere o utilice en perjuicio del titular de los datos o de un tercero

Aircrack-ng

Aircrack-ng es un conjunto de herramientas para capturar y romper tráfico cifrado con WEP, WPA o WPA2-PSK

- GPL, Libre y gratuito. Disponible para Linux y parcialmente para Windows
- Herramienta prácticamente *standard*
- Gerix es un interfaz gráfico y asistente para aircrack-ng

WPA solo es vulnerable con contraseñas de baja calidad

Formado por

- aircrack-ng averigua claves WEP y WPA
- airdecap-ng descifra el tráfico una vez que se conoce la clave
- airmon-ng pone la tarjeta en modo monitor
- aireplay-ng inyecta tráfico en la red
- airodump-ng *sniffer* de paquetes
- packetforge-ng crea paquetes crifrados, para inyección
- airdriver-ng herramienta para manejar drivers de tarjetas inalámbricas
- tkiptun-ng realiza ataques WPA

Cifrado RC4

texto-claro XOR keystream = texto-cifrado
texto-cifrado XOR keystream = texto-claro
El keystream (flujo clave) es una secuencia pseudoaleatoria, generada a partir de clave secreta y vector de inicialización *Initialization Vector, IV*

- El algoritmo que genera la secuencia es conocido
- El vector de inicialiación se transmite en claro

Todos los ataques se basan en conocer un número suficiente de vectores de inicialización

Vectores de Inicialización

3 bytes, incluidos en cada paquete WEP
WEP puede usar claves de dos tamaños

- 64 bits (40 clave + 24 IV)
- 128 bits (104 clave + 24 IV)

Ataques contra WEP

- Método FMS (Fluhrer, Mantin and Shamir). Año 2001
- Método KoreK. Año 2004
  Necesita 300.000 IVs (64 bits) o 1.500.000 IVs (128 bits)
- Método PTW (Pychkine, Tews, Weinmann). Año 2007
  Necesita 20.000 IVs (64 bits) o 40.000 IVs (128 bits)
  Solo funciona con paquetes ARP

A priori no se puede conocer la longitud de la clave usada

Inyección de tráfico

Una respuesta de ARP contiene un nuevo IV
Para generar tráfico, se envían muchas solicitudes de ARP al AP

- Se puede capturar una solicitud legítima y reenviarla reiteradamente
- Se puede generar un ARP partiendo de cero

Para inyectar tráfico con éxito es necesario

- Estar cerca del AP. Una buena antena puede ser de ayuda
- Asociarse con el AP. Es suficiente una *fake authentication*: Podemos asociarnos al AP sin conocer la clave (aunque en principio no podremos enviar tráfico)

Otros ataques

No obtiene clave WEP, sino el PRGA (*pseudo random generation algorithm*), que permite generar paquetes cifrados y emplearlos luego en ataques de inyección

- Chopchop
- Fragmentación

El PRGA se almacena en fichero de extensión .xor

Ejecución de los ataques

```
ALICE=00:18:39:D3:EE:DA     # ESSID, MAC del AP
MALLORY=00:1A:EF:0A:07:CB   # MAC del atacante
INTERFACE=wlan0             # Interface del atacante
SSID=test_rom               # SSID del AP
CANAL=5                     # Canal que esté usando el AP
FICHERO_CAPTURA=/tmp/captura
```

Ponemos el interface en modo monitor, en el canal del AP

`airmon-ng start $INTERFACE $CANAL`

- Comprobamos que podemos inyectar trafico
  `aireplay-ng --test -e $SSID -a $ALICE $INTERFACE`
- En una nueva shell, capturamos el tráfico
  `airodump-ng --channel $CANAL -a $ALICE -w $FICHERO_CAPTURA $INTERFACE`
- En una nueva shell, hacemos una asociación falsa con el AP, y la renovamos periódicamente
  `aireplay-ng --fakeauth 6000 -o 1 -q 10 -e $SSID -a $ALICE -h $MALLORY $INTERFACE`
- Esperamos peticiones ARP para reinyectarlas
  `aireplay-ng --arpreplay -b $ALICE -h $MALLORY $INTERFACE`
- Búsqueda de la clave
  `aircrack-ng -b $ALICE $FICHERO_CAPTURA*.cap`

Los AP modernos no suelen ser vulnerables a ataques básicos como este