Pescando sin sedal

Wardriving

Incubus

Jueves 15 julio 2004.

Tiempo ha pasado desde la película "War Games", donde un jovencísimo Mattew Broderick nos sorprendía cuando eramos niñ_s, accediendo a la red o redes a través de un curioso método de búsqueda de números de telefónos de acceso a modems, PBX, etc... Los cuales le permitian salir hacia fuera sin coste alguno.

Esa táctica fue bautizada como **Wardialing** y actualmente se sigue usando buscando nºs de teléfono de coste gratuito como los nºs 900, etc, los cuales esconden por detrás routers y líneas descuidadas.

La llegada de la tecnología WiFi, nos ha traido una nueva forma de buscar accesos a Internet y a redes privadas, soló que esta vez con unos métodos aún más curiosos que los antes citados, el **Wardriving**.

Inspirado en el wardialing, el wardriving consiste en descubrir redes inalámbricas mientrás que caminas o conduces por tu ciudad.Con el objetivo de encontrar acceso a Internet gratuito o entrar en otras redes públicas o privadas.

Este artículo te muestra las técnicas empleadas, las herramientas usadas y algunos que otros detalles.

Para fines más "eticos", este texto te valdrá para localizar nodos de redes inalámbricas ciudadanas, las cuales te permitiran en la mayoria de los casos un acceso libre y gratuito a Internet y a la red paralela construida por la comunidad wireless.

Los aparejos para salir de pesca

Antes de "meternos en el agua", necesitaremos tener un portatil con wifi integrado o con alguna bahía PCMCIA (también podemos usar una PDA), para poder introducir en ella una tarjeta WiFi. Es recomendable que las tarjetas que se utilizen tengan la posibilidad de conectarles una antena externa, ya que ello nos facilitara la tarea en nuestro rastreo para descubrir redes a cierta distancia.

Las antenas que usemos es cosa nuestra, existen ya antenas comercializadas especiales para estas practicas, aunque lo más divertido y económico es hacerte una tú mism_.

El equipo de pesca

En la construcción casera de antenas dispones de todo tipo de imagineria popular cuyos materiales los puedes conseguir directamente de los residuos domesticos que generamos todo los días en nuestros hogares (tetra bricks, latas de sardinas, botes de patatas fritas, etc...) En las webs de las comunidades inalámbricas puedes encontrar facilmente manuales de construcción de todo tipo de antenas caseras.

Originalmente el wardriving se practica yendo en automovil de tal forma que mientras que conduces por la ciudad , el portatil con sus herramientas adecuadas irá detectando redes allá por donde pases. Como no todo el mundo dispone de automovil, mucha gente practica este deporte "underground" caminando...incluso he llegado a ver algunos practicandolo en tandems X−).

Hay gente más sofisticada, que usa dispositivos GPS a la vez para ir registrando las coordenadas geográficas donde haya descubierto una red y posteriormente añadiendolos en un mapa de la zona en cuestión.

Cargando al carrete

Durante todo este texto, vamos a tratar el tema del wardriving sobre GNU/Linux, ya que este sistema nos ofrece herramientas muy versatiles, y tampoco nos gustaría que se nos vaya todo al carajo cuando hayamos descubierto algo y estemos trabajando en el acceso hacia la red (ese tipo de cosas que ocurren usando Micro$oft) X−)

Las tarjetas WiFi, tienen varias formas de funcionamiento:para funcionar como cliente punto a punto, para convertir tu computador en un punto de acceso,etc... Pero el que actualmente nos interesa es que nuestra tarjeta pueda funcionar sobre nuestro Linux−box en modo monitor. Este modo nos permitirá, como el mismo indica, poner nuestra tarjeta en modo de monitorización para detectar conexiones wireless.

Para saber si nuestra tarjeta funciona en nuestro sistema GNU/Linux en modo monitor, tendremos que comprobarlo con la siguiente línea de mandato.

```
$iwpriv eth0
eth0      Available private ioctl :
         force_reset      (8BE0) : set   0       & get   0
         card_reset       (8BE1) : set   0       & get   0
         set_port3        (8BE2) : set   1 int   & get   0
         get_port3        (8BE3) : set   0       & get   1 int
         set_preamble     (8BE4) : set   1 int   & get   0
         get_preamble     (8BE5) : set   0       & get   1 int
         set_ibssport     (8BE6) : set   1 int   & get   0
```

```
        get_ibssport     (8BE7) : set   0       & get   1 int
        monitor          (8BE8) : set   2 int   & get   0
        dump_recs        (8BFF) : set   0       & get   0
```

Suponiendo, claro esta **eth0**, como el disposito wifi. La información que nos interesa que aparezca es la que indica el valor **monitor**.

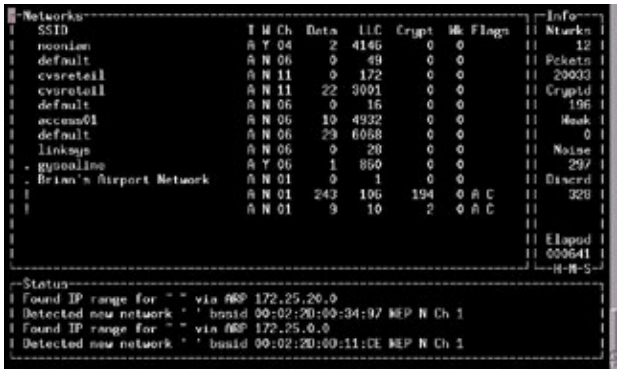
Si tu tarjeta no soporta ese modo en Linux(generalmente suele pasar en las que tienen el chip Hermes, tarjetas orinoco o avaya), nos tocara compilar un nuevo kernel y parchear los módulos del paquete pcmcia−cs.

Pero para no volver a escribir la historia os remitó a este documento de la comunidad wifi **madridwireless.net**:

- **Howto sobre Orinoco (Hermes) en modo monitor**

Una vez que nuestra tarjeta interactua en modo monitor, nos dispondremos a bajarnos unas herramientas útiles para poner "rayos−x" en nuestras computadoras ;−)

La herramienta más usada en GNU/Linux, para detencción de redes wifis es **kismet (www.kismetwireless.net)**, la cual nos podremos bajar las fuentes de su web o como gente comóda como yo hacer un *apt−get install kismet* en nuestra consola de **Debian** (he dicho que es la mejor distribucción Gnu/Lnux? ;−))



kismet en acción

Esta herramienta corre bajo consola con una bonita interfaz ncurses, y nos mostrará información detallada de las redes que detectemos:

- si el equipo detectado esta en modo cliente o en modo punto de acceso(ap)
- información detallada de las caracteristicas de transmisión
- número e información sobre los clientes conectados a dicha red
- si transmite datos bajo protección WEP
- analisis de los paquetes transmitidos para descubrir el rango de subred en el que actua la red.
- etc...

Tambíen funciona con otras aplicaciones para usarlo con GPS o para que nos de divertidos avisos sonoros cuando detecta una red. En definitiva es una herramienta imprescindible.

Existen otra buena herramienta , como **Wellenreiter(www.wellenreiter.net)** , que funcionan en modo GUI lo cual tiene el inconveniente de compilar más fuentes para ello. Yo personalmente prefiero con creces kismet.

Como también nos iremos encontrando con redes protegidas con WEP, necesitaremos un analizador de paquetes encriptados con WEP para que posteriormente nos muestre la clave WEP usada en dicha red, es decir una herramienta de crackeo de claves WEP.

La herramienta por excelencia es **Airsnort(airsnort.shmoo.com)**.Un GUI que a la vez que nos detecta redes wifi, realiza una analisis para sacar en formato ASCII o hexadecimal la clave WEP.

Much_s os preguntareis por que usar kismet si Airsnort ya nos detecta redes wifi, la respuesta es sencilla: Airsnort no da información detallada de la red encontrada.

Para instalar airsnort, te bastara con realizar un *apt−get install airsnort* si usas Debian, compilarte las fuentes que hay en la web oficial o buscar un paquete rpm.

La captura del salmón

Antes de comenzar el escaneo, es recomendable cambiar la dirección MAC de nuestra tarjeta. Con ello evitaremos los sistemas de restrinción por MAC, consiguiendo con ello dificultar la identificación de nuestra tarjeta.

```
$ifdown eth0
$ifconfig eth0 hw ether 00:00:AC:DC:00:00
$ifup eth0
```

Es recomendable comprobar que se ha cambiado la MAC correctamente, ejecuntando **ifconfig**.

Para poner en marcha **kismet**, previamente deberemos configurar la aplicación en el fichero **etc/kismet/kismet.conf**. Allí deberemos indicar detalles tan imprescindibles como el tipo de tarjeta que usamos, la interfaz de red en la cual funciona, si queremos habilitar sonidos de aviso, etc... Dicha configuración es muy sencilla, así que no es necesario contar grandes detalles de ello.

Una vez configurado todo ejecutamos en nuestra consola la orden **kismet**.

En otra consola podemos ver que verdaderamente la tarjeta esta en modo monitorización:

```
$ifconfig eth0
eth0      Link encap:UNSPEC  HWaddr 00−00−AC−DC−00−00−00−00−00−00−00−00−00−00−00−00
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:344 errors:10 dropped:354 overruns:0 frame:10
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:71400 (69.7 KiB)  TX bytes:0 (0.0 b)
          Interrupt:9 Base address:0x100
```

El campo UNSPEC nos muestra la dirección MAC con otros bloques de ceros añadidos, ello nos indica que nuestra tarjeta esta en modo promiscuo, monitorizando y por lo cual funcionando.

Cuando ejecutamos kismet, nos encontramos con una ventana central, la cual esta dividida en varias columnas, las cuales nos informaran según vaya capturando redes de diversa información:

- Essid de la red detectada.

- El modo de funcionamiento del dispositivo Wifi detectado, nos lo indica con la columna **T**. Dicha bandera, nos ofrecera diferentes valores como [A] si es un punto de acceso, [H] si esta en modo ad−hoc, [P] si es un dispositivo en modo "probe request", etc...

- Uso de encriptación WEP, marcada con la columna **W**.La cual nos indicara con [Y] en caso afirmativo y nos pondrá la información de dicha red con carácteres de distinto color.

- El número de paquetes capturadados, viene indicado por la columna **Packets**.

- El canal viene marcado por la columna **Ch**

- La columna con la leyenda **Flags**, nos informará trás el analisis de los paquetes con diversos valores de banderas el tipo de clase de red que estamos investigando. Para ello nos mostrará valores por ejemplo como T3 en caso de tratarse de tráfico ip , U3 en caso de tratarse de tráfico UDP , etc...

- **Ip range**, sin duda esta columna es la información que más nos interesa, ya que nos dira el rango IP de la red o dispositivo detectado.Mientras que se hace un analisis exhautivo de las tramas detectadas este valor aparecera con valor 0.0.0.0

Podemos interactuar con el programa para obtener más recursos.

Con la tecla **S** nos permitira seleccionar la red detectada.Una vez seleccionada podemos ver más información pulsando otras teclas. Por ejemplo:

- La tecla **I** nos muestra información más detallada sobre el dispositivo de red detectado.

- La tecla **C** nos muestran las MACS y diversos detalles sobre los clientes conectados a dicha red.

- Con la tecla **H** se nos ofrecerá un buen manual de ayuda del uso del programa.

A grandes rasgos y basicamente es la información es la información que nos servira kismet.

Obtener la información que nos interesa nos puede costar tiempo, ya que kismet necesitará hacer analisis de tramas para podernos mostrar la información que más nos interesa, es decir el rango de direccionamiento ip. Por lo cual cuanto más paquetes capturemos más rapidamente nos dará esos datos.

Antes de que me olvide, podemos probar una vez que hemos detectado una red que este funcionando en modo ap, conseguir ip a traves de un cliente dhcp como **pump** o **dhclient**.Quizás tengamos suerte y detrás exista un DHCP−Server, el cual nos facilitará el rápido acceso a la red dandonos una ip dinámica ;−)

Para ello basta con teclear:

```
$pump −i eth0
```

Si vemos que trás un rato, no se nos asigna ip dinámica, tendremos que esperar a que kismet nos de el rango ip.

Una vez que consigamos el rango ip, con una herramienta como **ipcalc** podremos saber la mascara de red y el broadcast.

Supongamos que detectamos una ip 10.64.6.25 en un dispositivo en modo ap (el modo que nos interesa verdaderamente), en canal 11 y con essid "HACKME", podemos comprobar los datos de la red de estas

diferentes formas.

```
$ipcalc 10.64.6.26

Address:   10.64.6.26           00001010 .01000000.00000110.00011000
Netmask:   255.0.0.0 = 8        11111111 .00000000.00000000.00000000
Wildcard:  0.255.255.255        00000000 .11111111.11111111.11111111
=>
Network:   10.0.0.0/8           00001010 .00000000.00000000.00000000 (Class A)
Broadcast: 10.255.255.255       00001010 .11111111.11111111.11111111
HostMin:   10.0.0.1             00001010 .00000000.00000000.00000001
HostMax:   10.255.255.254       00001010 .11111111.11111111.11111110
Hosts/Net: 16777214             (Private Internet RFC 1918)

$ipcalc 10.64.6.25/30

Address:   10.64.6.26           00001010.01000000.00000110.000110 01
Netmask:   255.255.255.252 = 30 11111111.11111111.11111111.111111 00
Wildcard:  0.0.0.3              00000000.00000000.00000000.000000 11
=>
Network:   10.64.6.24/30        00001010.01000000.00000110.000110 00 (Class A)
Broadcast: 10.64.6.27           00001010.01000000.00000110.000110 11
HostMin:   10.64.6.25            00001010.01000000.00000110.000110 01
HostMax:   10.64.6.26           00001010.01000000.00000110.000110 10
Hosts/Net: 2                    (Private Internet RFC 1918)
```

Una vez obtenido estos datos es sencillo engancharnos a esa red Wifi, solo nos bastará con poner las configuraciones adecuadas en nuestra tarjeta wifi. Recordad antes salir de kismet y tirar la interfaz para que salga de modo monitor. Esto puede valeros de ejemplo, aunque quizás nos toque probar varias máscaras de red.

```
$ifconfig eth0 down
$ifconfig eth0 hw ether 00:00:AC:DC:00:00
$ifconfig eth0 10.64.6.28 netmask 255.255.255.252 up
$iwconfig eth0 essid "HACKME" mode managed channel 11
```

Tendremos que esperar una respuesta en la consola que nos indique que nos hemos conectado.

Otra forma de ver si hemos "capturado el pez" es comprobarlo con el monitor **wavemon**.

Para intentar salir hacia Internet o la red, tendremos que descubrir la pasarela. Si el rango de red que hemos capturado es del tipo 10.64.6.26/24, por costumbre el router hacia Internet se escondera detrás en la primera dirección de la red, por lo cual seguramente sea 10.64.6.1.

En caso de que la subred sea 10.64.6.26/30, con los datos que nos da ipcalc, es facil pensar que la pasarela se encuentre en 10.64.6.25...aunque podemos equivocarnos.

Aunque nunca me ha ocurrido esta situación, podemos tener problemas si la red usa control de acceso por direcciones MAC, por lo que no podremos salir hacia la red. Para ello podremos usar el programa **arping** de tal forma que se nos intente dar respuesta desde la pasarela.

```
$arping −i 10.64.6.25
```

Si obtenemos respuesta, podremos intentar expulsar a algún cliente con el programa **airjack** y apropiarnos de su MAC, o realizar labores detectivescas para ver si los equipos clientes usan MAC similares, donde podremos seguir la pista del fabricante de los clientes para ponernos una MAC apropiada. Ello lo podremos hacer haciendo comprobaciones con el fichero **/etc/pcmcia/wireless.opts**

Si me equivocó en el método, indicadlo por favor en los comentarios al texto.

Otro metodo más sencillo, y muy intrusivo por cierto, es tirar de una herramienta como **ettercap**. Dicha herramienta es un sniffer magnifico, que nos mostrará todos los equipos que estan conectados a la red, de tal forma que podremos detectar facilmente donde se esconde la pasarela entre otras bastantes cosas.

En fin que sin gran esfuerzo detectivesco, probando pings al broadcast y a direcciones de Internet,etc... encontraremos lo que buscamos y donde estamos metidos.

Así una vez que una vez descubierta la pasarela de salida hacia Internet, solo tendremos que añadir la ruta.

Supongamos que la pasarela es 10.64.6.25 con netmask 255.255.255.224, entonces teclearemos lo siguiente para salir hacia esa red privada y hacia Internet. Comprobando posteriormente que las rutas han sido bien añadidas.

```
$route add −net 10.64.6.24 netmask 255.255.255.252 eth0
$route add default gw 10.64.6.25 netmask 0.0.0.0
$route −n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.64.6.24      0.0.0.0         255.255.255.252 U     0      0               0 eth0
0.0.0.0         10.64.6.25      0.0.0.0         UG    0      0        0 eth0
```

El pez se nos oculta

Hasta ahora todo ha sido sencillo relativamente, el caso se nos complica si la red donde queremos "pescar" esta encriptada en WEP, ya que kismet no nos mostrará nada revelante al estar el tráfico cifrado.

Ello nos lleva a sacar otro as de la manga, y mostrar **airsnort**.

Airsnort es un GUI que nos mostrara información básica del escaneo de red que estemos realizando y con paciencia (algunas muchas veces mucha paciencia) nos reventará la clave WEP usada en dicha red trás un minucioso ánalisis de las tramas encriptadas capturadas.

Lógicamente Airsnort, nos pondrá la tarjeta en modo monitor y para comenzar a capturar tramos encriptados por la red, tendremos que indicarle sencillamente la interfaz de red de nuestra tarjeta wifi y el chip que usa (prism2, orinoco, etc...)

El programa nos muestra la MAC del dispotivo capturado, la essid, si esta en modo WEP, el nº de paquetes capturados, el nº de ellos capturados, y el nº de ellos que verdaderamente interesan a airsnort.

Tenemos que tener mucha paciencia, para que el programa nos muestre en las columnas **PW:Hex** y **PW:ASCII** la clave usada en la red. Ten en cuenta que necesita capturar miles de paquetes para mostrarnos la "pálabra mágica".

Por suerte podremos ir guardando las capturas realizadas salvandolas en la opción del programa **Save crack file**, por si nos toca volver otro día a seguir intentandolo, con lo que usando la opción **Load crack file** no tendremos que empezar el analisis desde el principio.

Imaginemos que trás un rato más o menos largo, tenemos suerte y se nos descubren esta clave en **ASCII: fabada** y su equivalente en **hexadecimal:66 61 62 61 64 61** dentro de un ap con MAC igual a 00:02:2D:23:AD:12

Acto seguido almacenaremos dicha clave en **/etc/kismet/kismet.conf** con la línea: **wepkey=00:02:2D:23:AD:12,666162616461**. En definitiva, la MAC del ap más la clave WEP en su formato hexadecimal.

Una vez hecho esto, kismet ya podrá realizar su trabajo más fructiferamente.

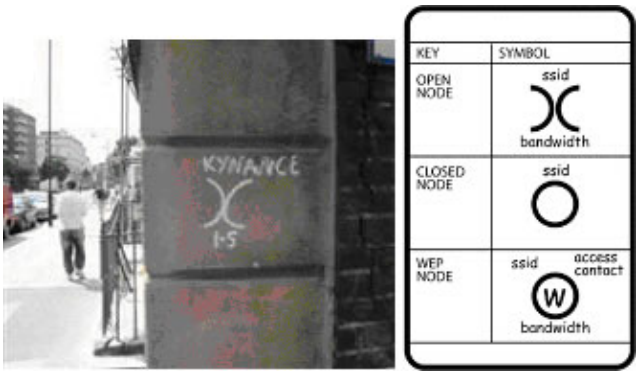
Cuando tengamos el rango ip que necesitamos, la forma de configurar nuestro dispositivo wifi se realizará igual como he contado más arriba. Sin olvidarnos de que ahora nos encontramos dentro de una red con WEP activado, por lo que tendremos que especificar la clave que usa dicha red.

```
$iwconfig eth0 essid HACKME mode managed channel 11
$iwconfig eth0 enc 666162616461
```

Pesca sin muerte

Como punto final, solo queria llamar a la responsabilidad del uso de estas técnicas.

Como autor no me hago responsable del uso que hagas de esta información.



Simbología callejera para wifers.

Si buscas una salida hacia Internet gratuita no abuses de la persona que se encuentra por detrás de ese equipo. Aprovecharte de todo su ancho de banda usando clientes p2p, bajadas de ficheros "gordos" o usarlo como plataforma para otras cuestiones escabrosas te convierte en un_ capull_ y puedes perjudicar gravemente a la persona propietaria de esa red o equipo. Una cosa es investigar y aprender estas cosas y otras ir jodiendo al prójimo.

Procura buscar salida a Internet a través de ciertos establecimientos. Locales como los Mc−Donalds, Starbucks, Burguer King, hoteles caros, etc... usan estas redes wifis para que sus empleados interactuen con PDA. Es más divertido saber que la conexión gratuita la pagan multinacionales o establecimientos de propietarios con mucho dinero, y pocas ganas de soltarlo con sus emplead_s. En esos sitios será interesante que utilices marcas que nos indiquen a la comunidad, que existe una red wifi con salida a Internet cerca. Para ello podras usar la simbología que se muestra en **http://www.warchalking.org**

Al fin y al cabo los bienes sustanciosos de estas compañias se reinvierten hacia l_s ciudadan_s en forma de ondas... no? ;−)

Feliz pesca y hacking chic_s.

Enlaces interesantes

- **http://www.wardriving.com**
- **http://wirelessanarchy.com/**
- **http://guerrilla.net/**
- **http://www.freenetworks.org/**
- **http://www.personaltelco.net**