**App Name :** MoneyTap personal loan app

**Package Name :** com.mycash.moneytap.app

**Version Code :** 211

**Version Name :** 3.3.5

**Compile SDK version : 28**

**Build version code : 28**

**Minimum SDK version : 16**

**Target SDK version : 28**

1. This application is being validated for **spyware PHA**
2. Let us understand the definition of spyware again

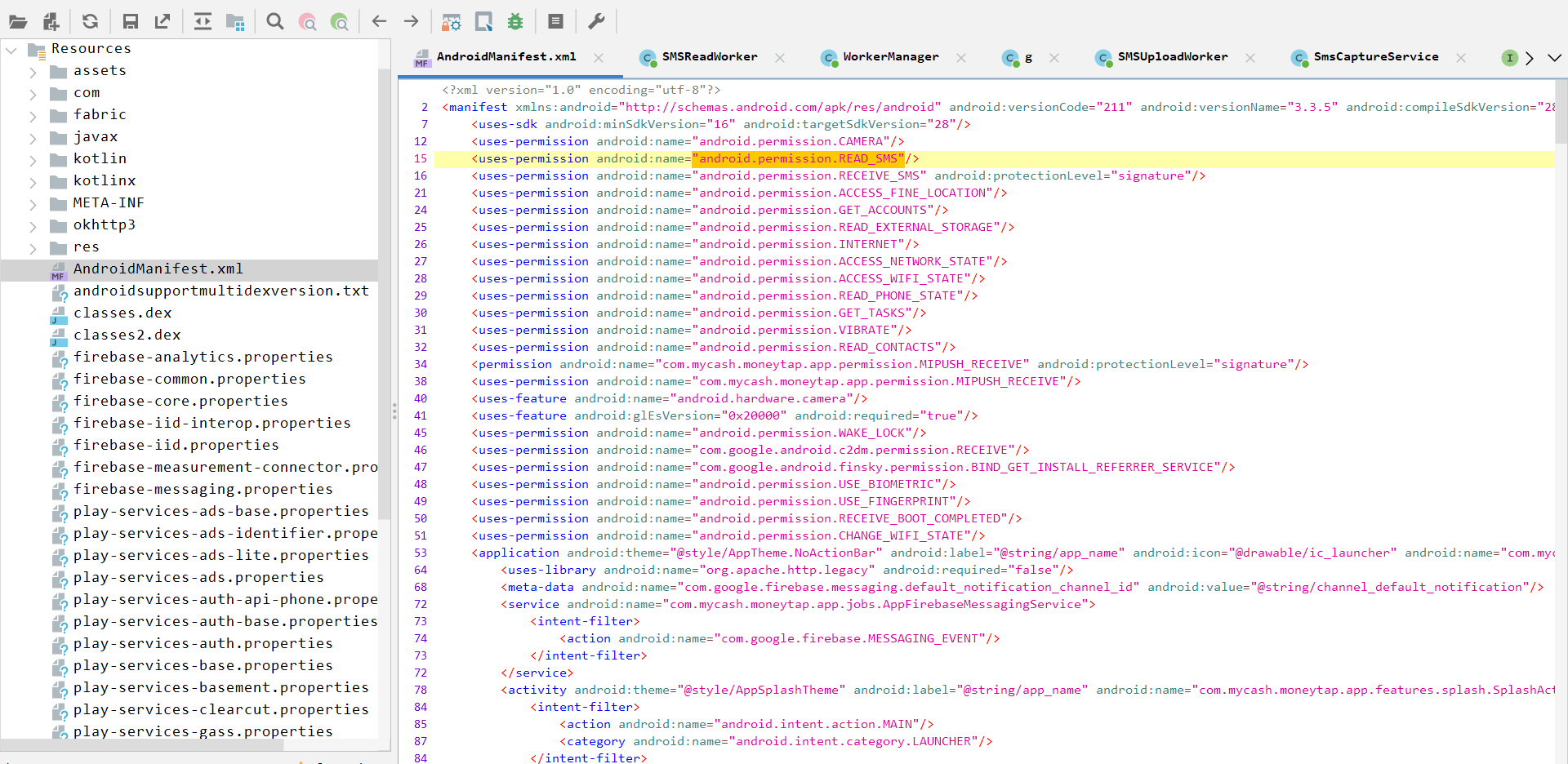
Spyware track the activities of user’s mobile phone and collects the data like contacts, sms, call logs, history, files, audio etc and send it to the third parties or hacker without user consent.

1. Permissions responsible for spyware :

* android.permission.READ\_CONTACTS
* android.permission.RECEIVE\_SMS
* android.permission.READ\_SMS
* android.permission.READ\_EXTERNAL\_STORAGE
* android.permission.READ\_CALL\_LOG
* android.permission.RECORD\_AUDIO
* android.permission.READ\_CALENDER
* android.permission.ACCESS\_COARSE\_LOCATION
* android.permission.ACCESS\_FINE\_LOCATION
* com.android.browser.permission.READ\_HISTORY\_BOOKMARKS

1. Lets start doing RE.
2. Open the application in the JADX .

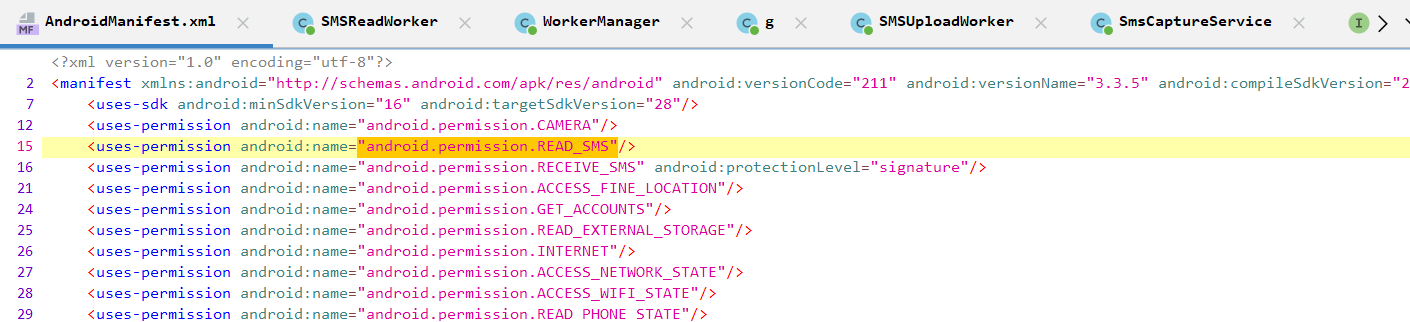
After opening, you can see the whole list of class files as mentioned in the screenshot below.



1. Double click the androidmanifest.xml

After opening the androidmanifest.xml file in JADX

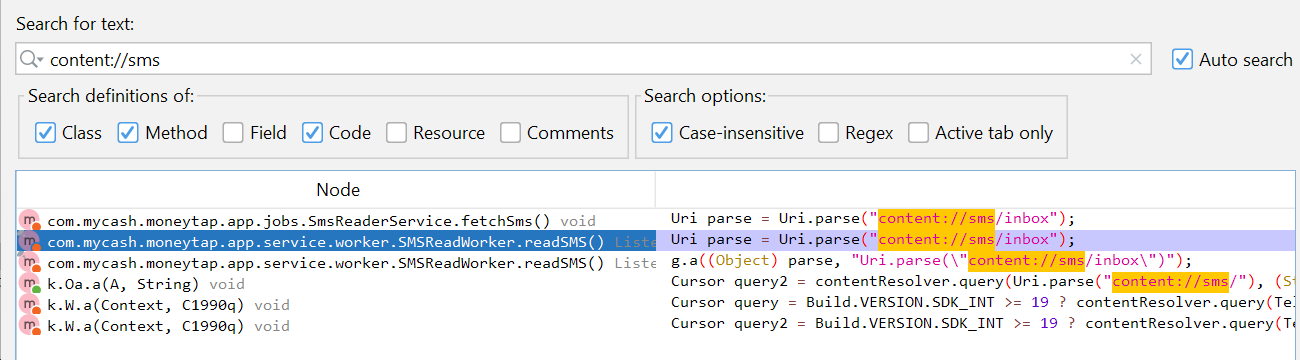
you can find **READ\_SMS**  at **line 15 .**



Here we can see that READ\_SMS is there

We will check for READ\_SMS.

1. Now click on  button in JADX to find for sms usage.
2. Now search content://sms as shown in below screenshot.



Now will check readSMS( ) method as it is returning list.

1. Here result of contentResolver( ) is assigned to query and then value is set into smsQueueModel

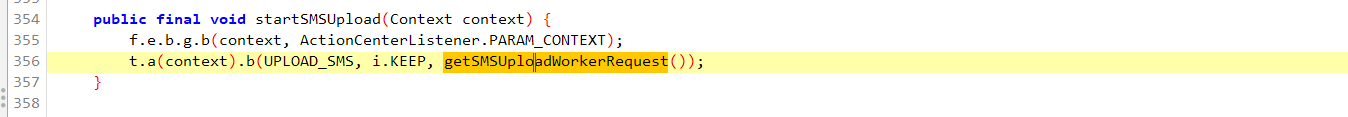
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Now we will check the declaration of startSMSUpload( )



1. Now we will check the declaration of getSMSUploadWorkerRequest( )

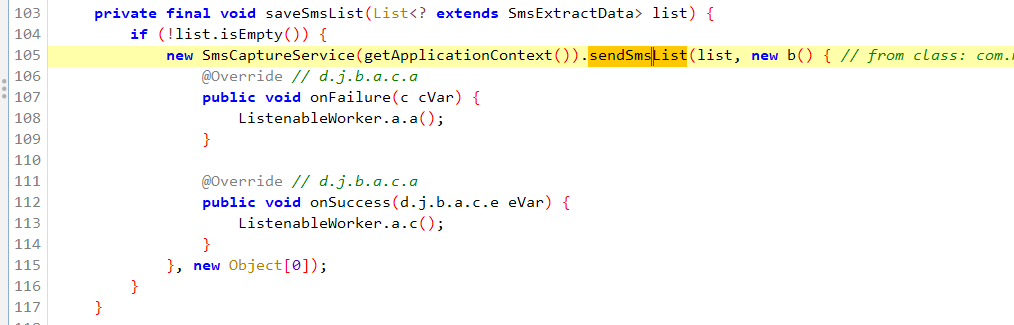
A text on a white background

Description automatically generated

Here we can see SMSUploadWorker.class

1. Now we will check the declaration of SMSUploadWorker.class

In this class, there is saveSmsList( ) in which sendSmsList( ) is defined.

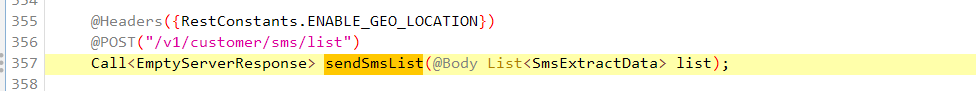


1. Now we will check the declaration of sendSmsList( )

A computer screen shot of a computer code

Description automatically generated

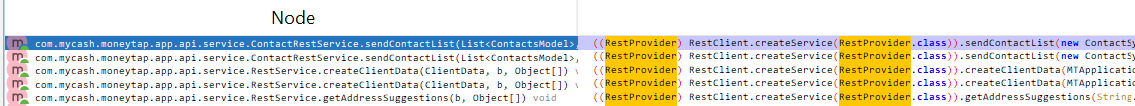
1. Here we can find the POST method of sendSmsList( ) which is highlighted as shown below.





1. Now we will check the usage of interface of sendSmsList which is RestProvider.

We will go for first one which is highlighted.



1. We will check the declaration of MTService which is highlighted below.

A computer screen shot of a computer screen

Description automatically generated

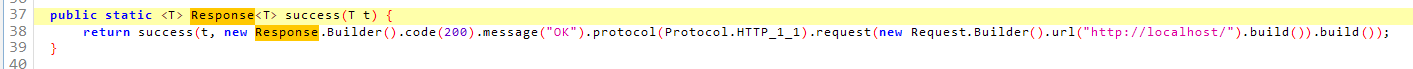


1. We will check Response of onResponse( ) method.





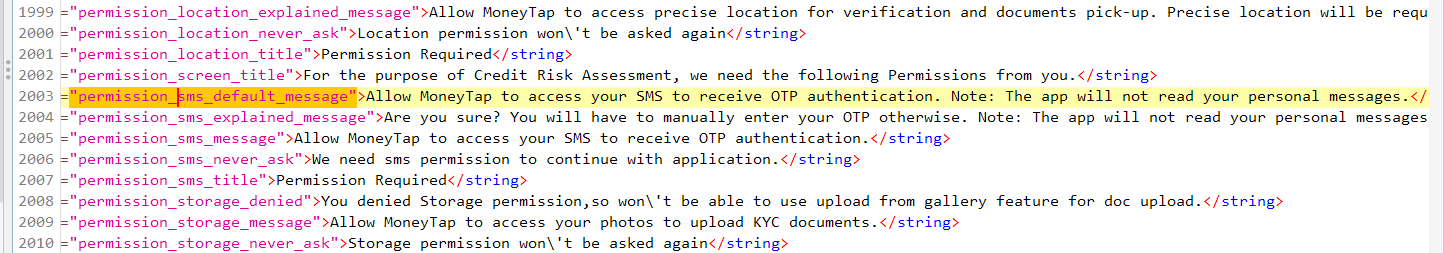
1. Here we can see that some http request is made but not clear.



Now we will check disclosure part.

1. Search sms in search bar

We will find res/value/string.xml



Here we can see that sms permission is requested by app.

So app is not spyware.