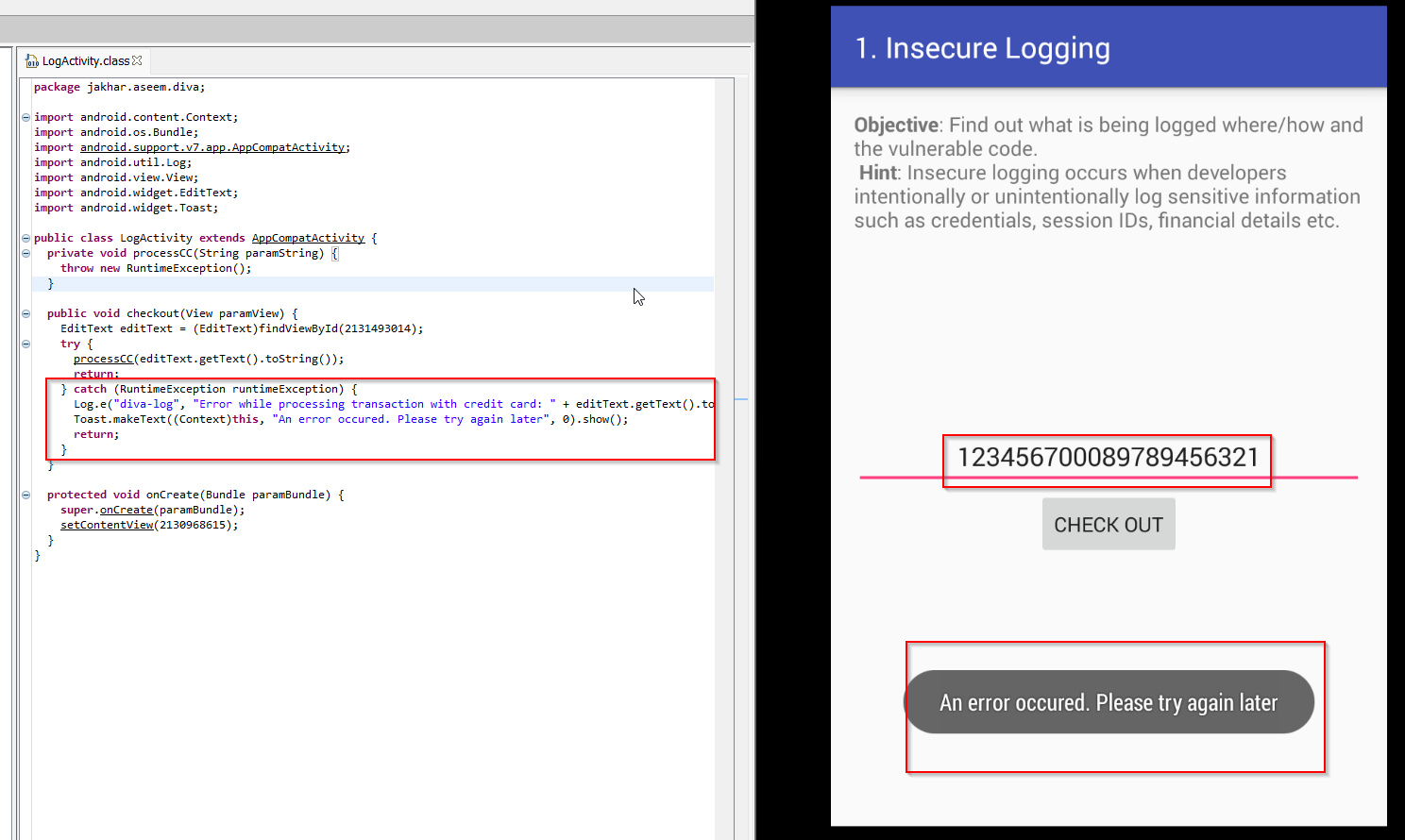
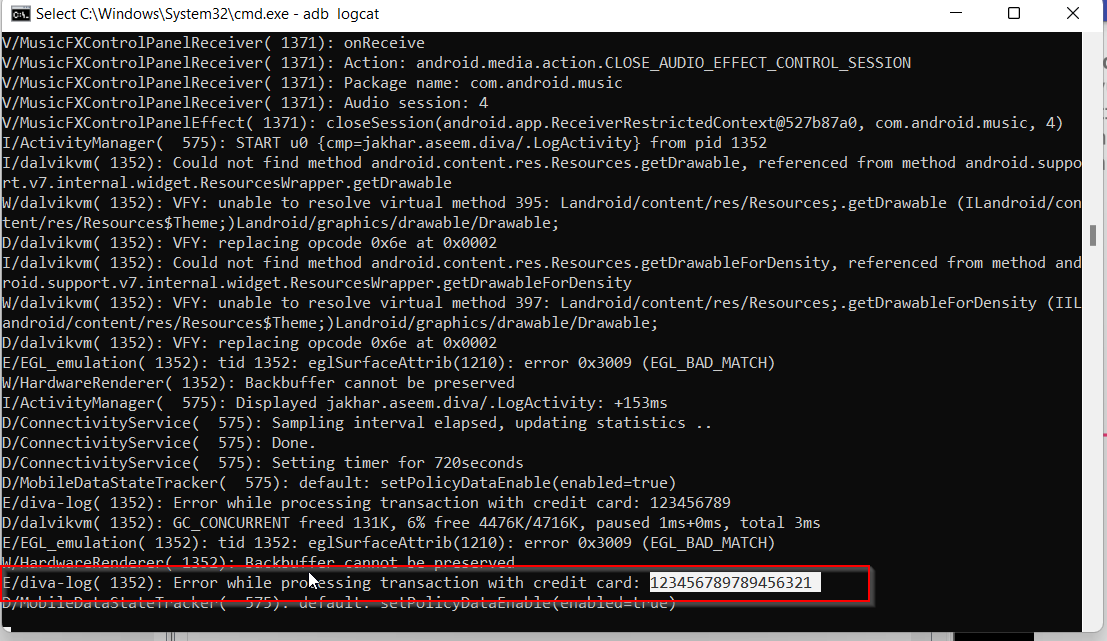
**Application Name: Diva.apk**

**1.INSECURE LOGGING**

All the data user will be giving to application is **not** **encrypted** and stored in plain text. anyone can read for credential using **log**

open command prompt in platform-tool directory.

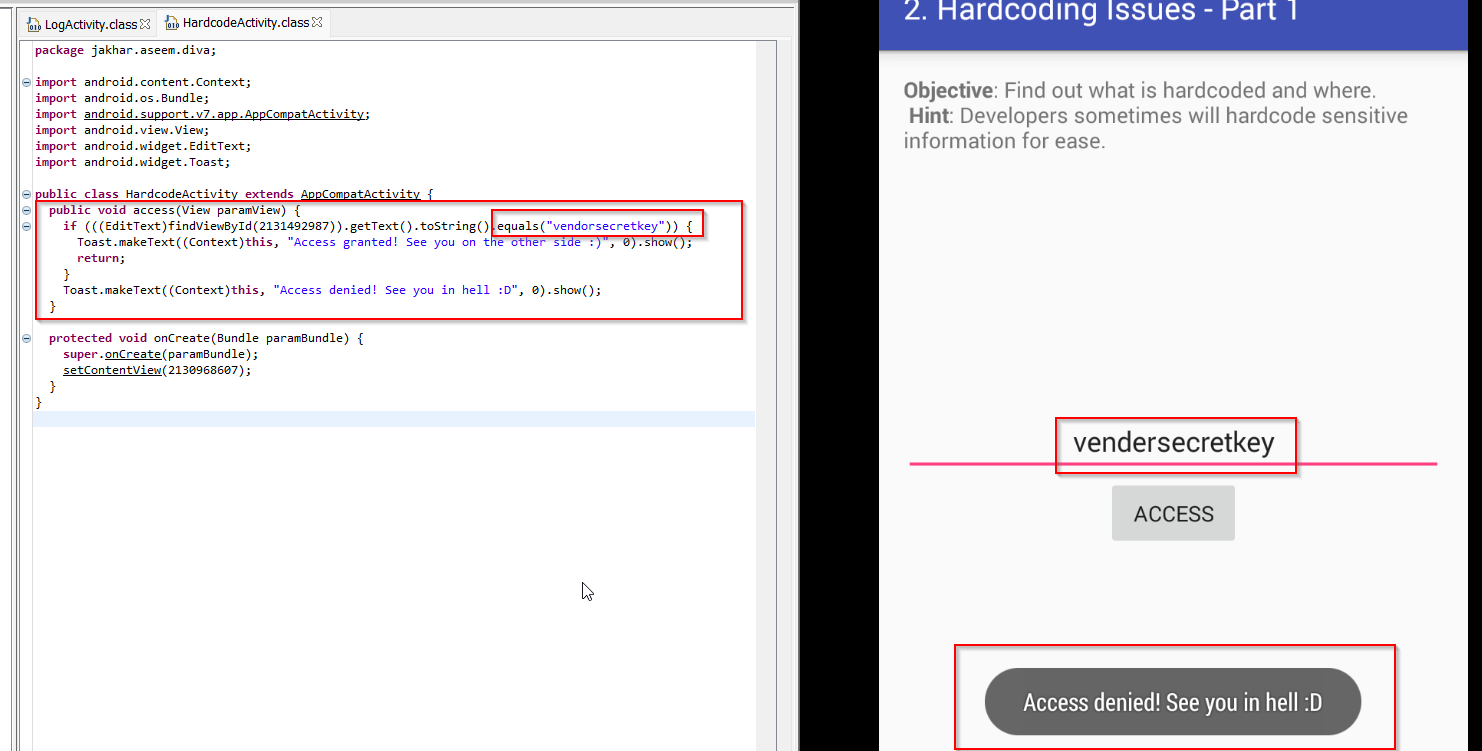
**Code** : adb logcat



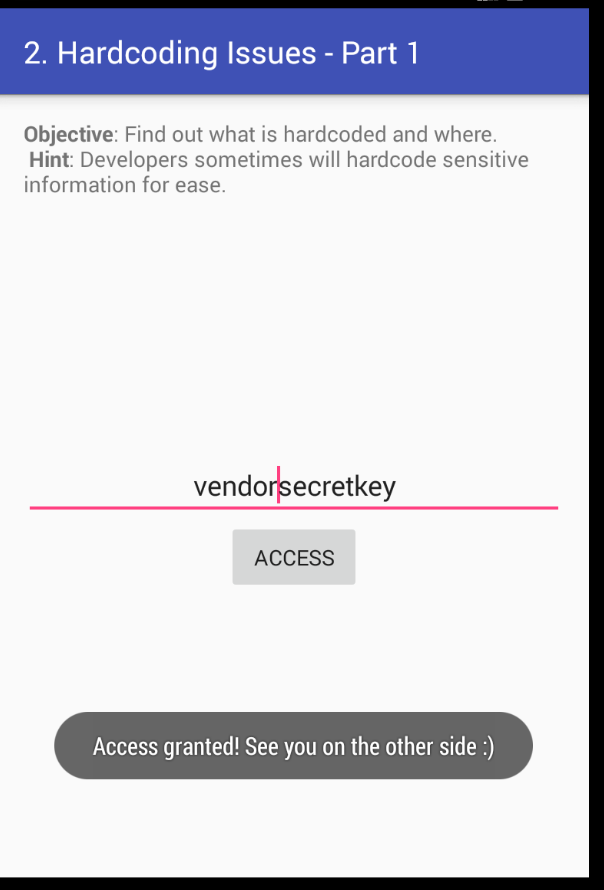
**2.HARDCODING ISSUES PART-1**

HardcodeActivity.class

To access user, have to enter vendor key. key is hardcoded in source code anyone can use this key to access application



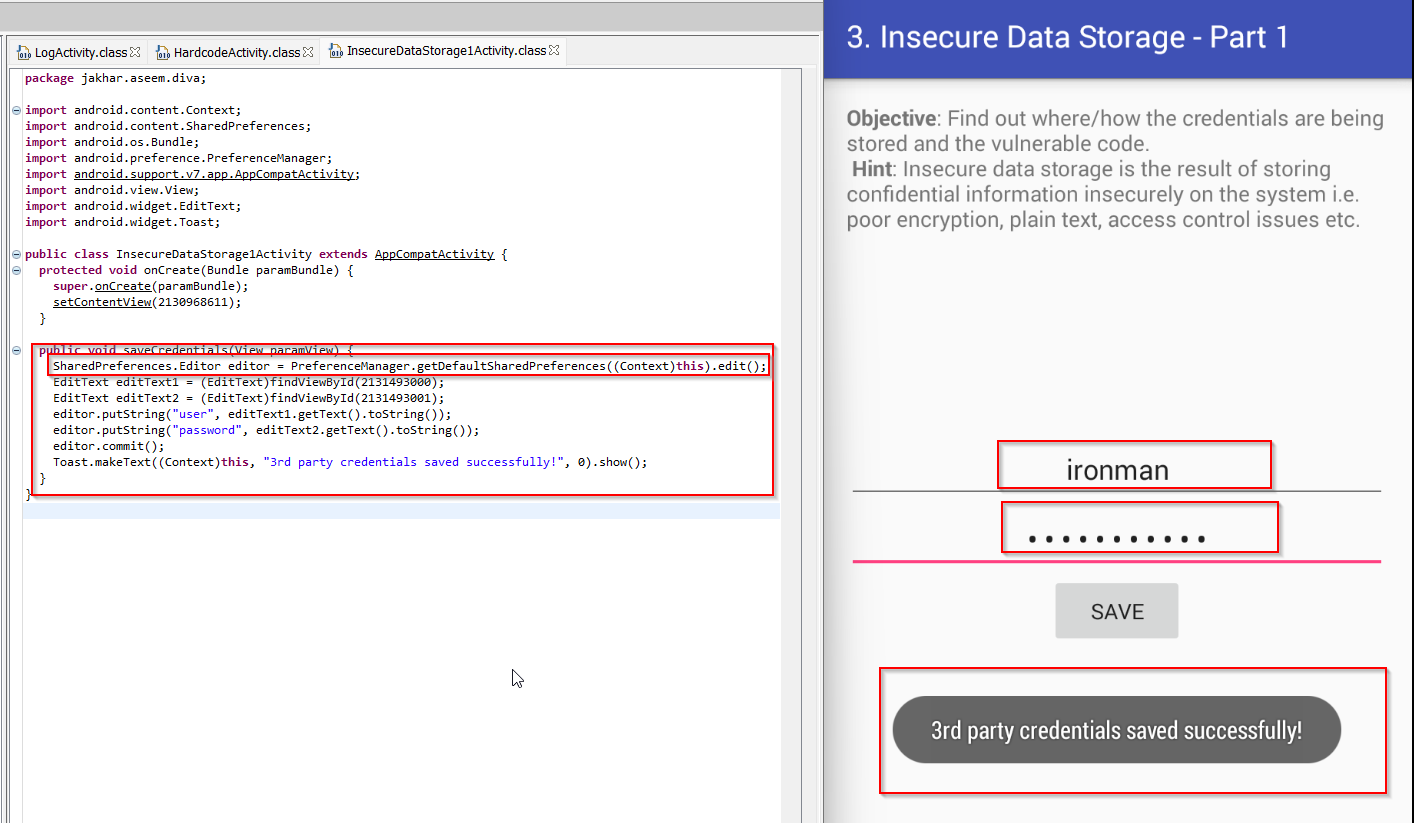
**Key**: vendorsecretkey



**3.INSECURE DATA STORAGE PART-1**

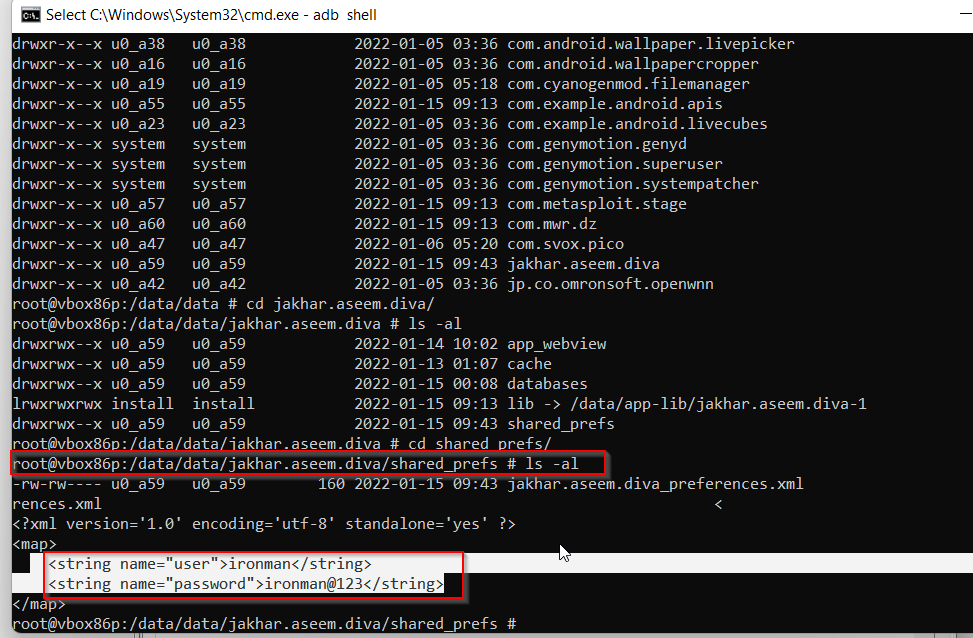
InsecureDataStorage1Activity.class

User have to save their username and password in application. As application is using Sharedpreferences in string data type. the user will not have to fill in details again and again. as data is stored in plain text anyone can view it



Credentials is stored within the shared\_prefs folder (source code)

**Path**: /data/data/lakkhar.aseem.diva/shared\_prefs

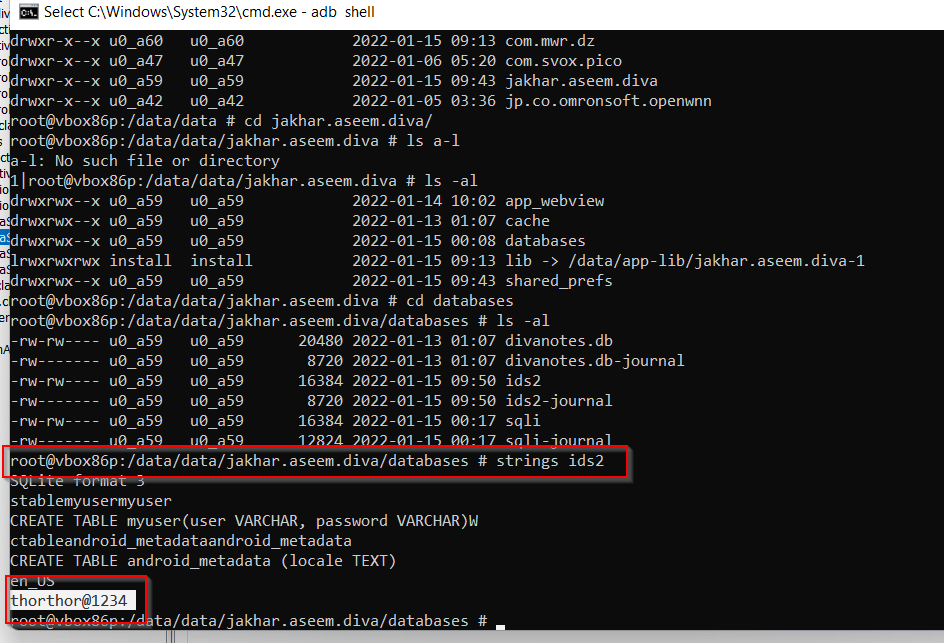


**4.INSECURE DATA STORAGE PART-2**

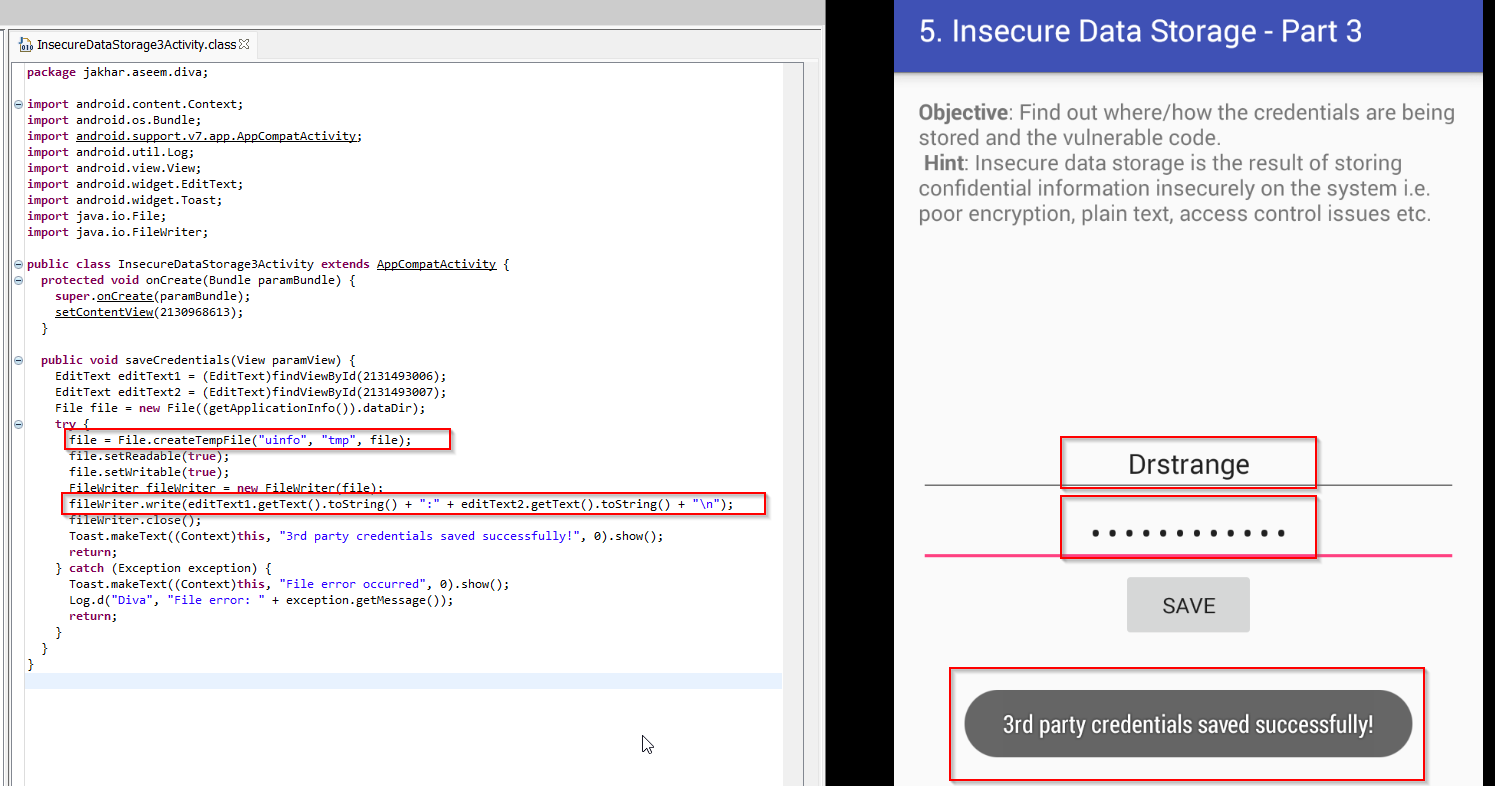
User have to save their username and password in application. As application is using Sharedpreferences in string data type. As codes Username and Passwords are stored in database “ids2”. anyone can access database

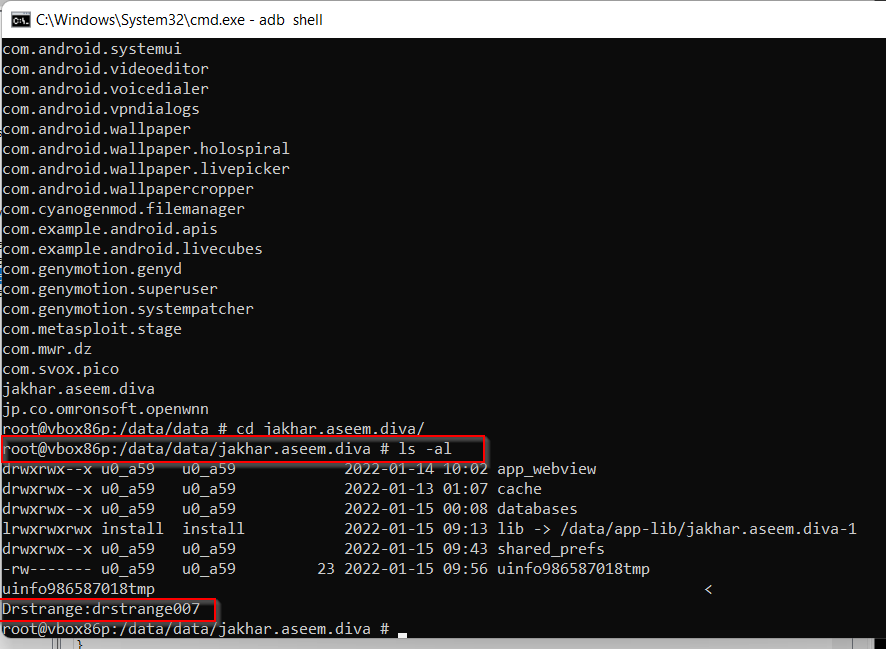
**Path**: /data/data/jakhar.aseem.diva/database

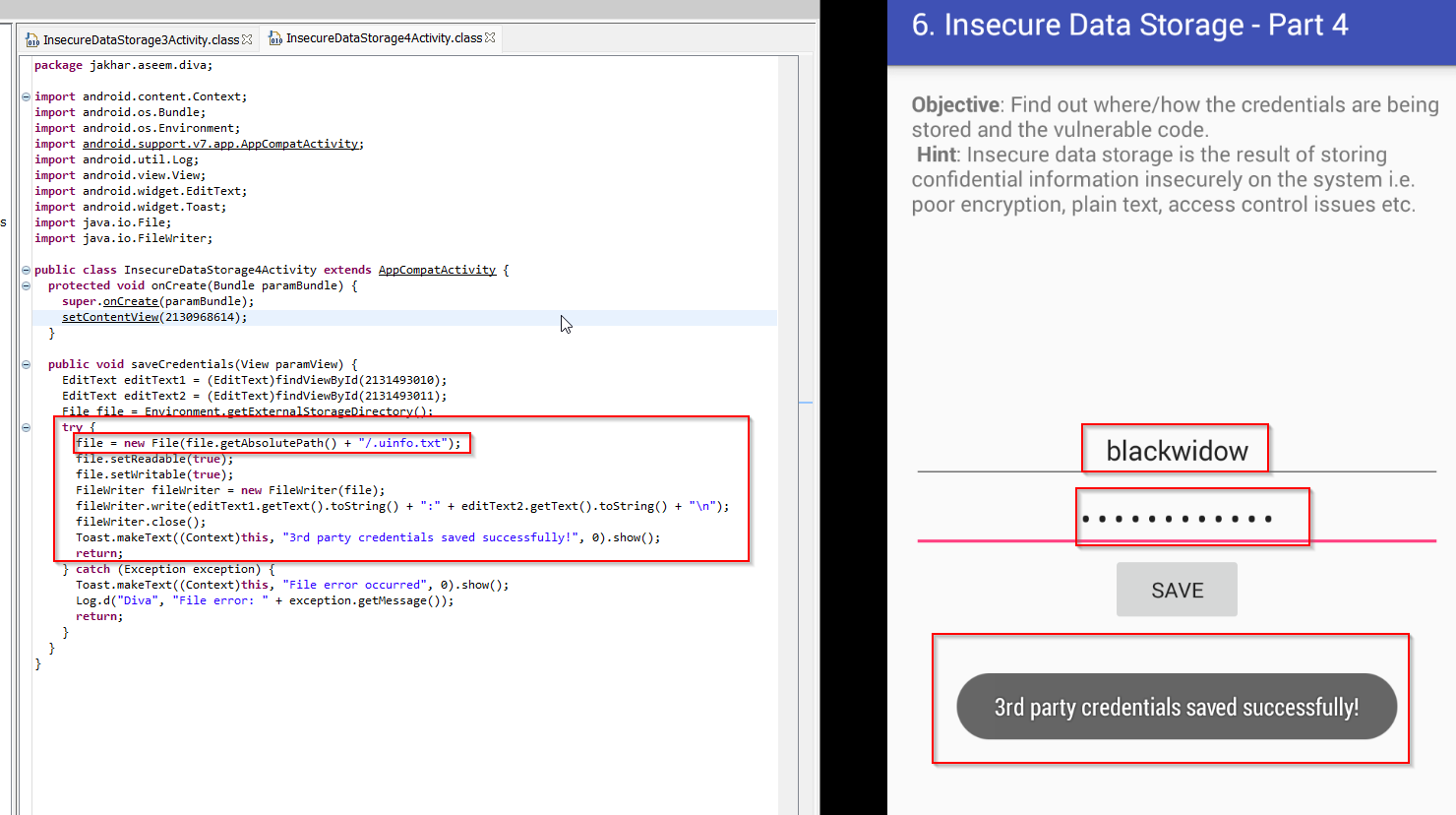
**Code**: strings ids



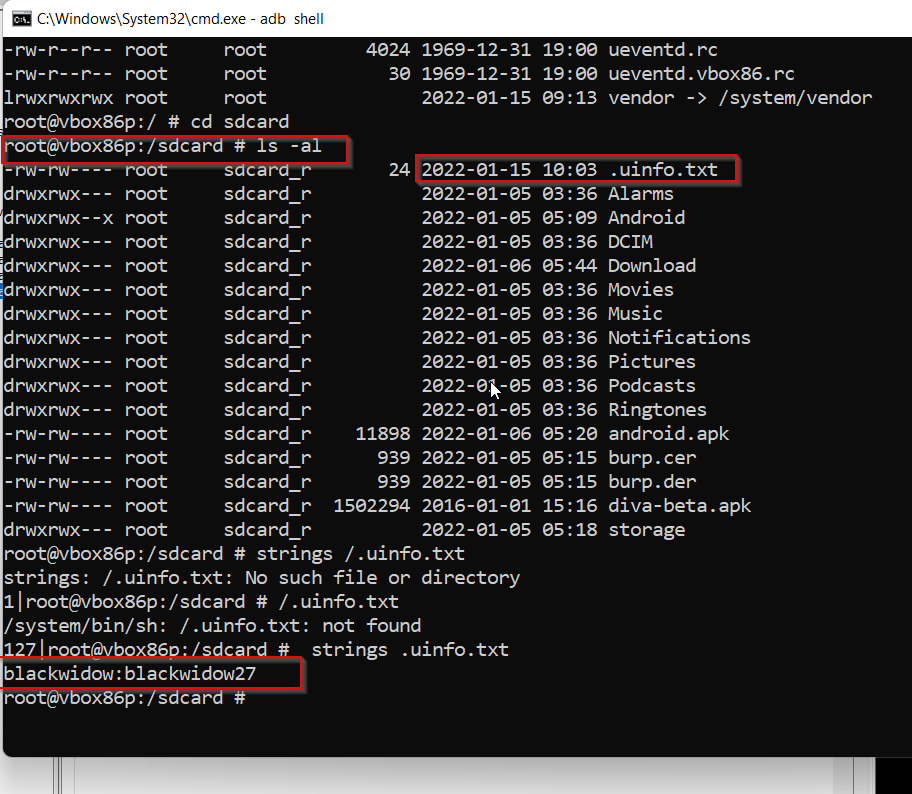
**5.INSECURE DATA STORAGE PART -3**

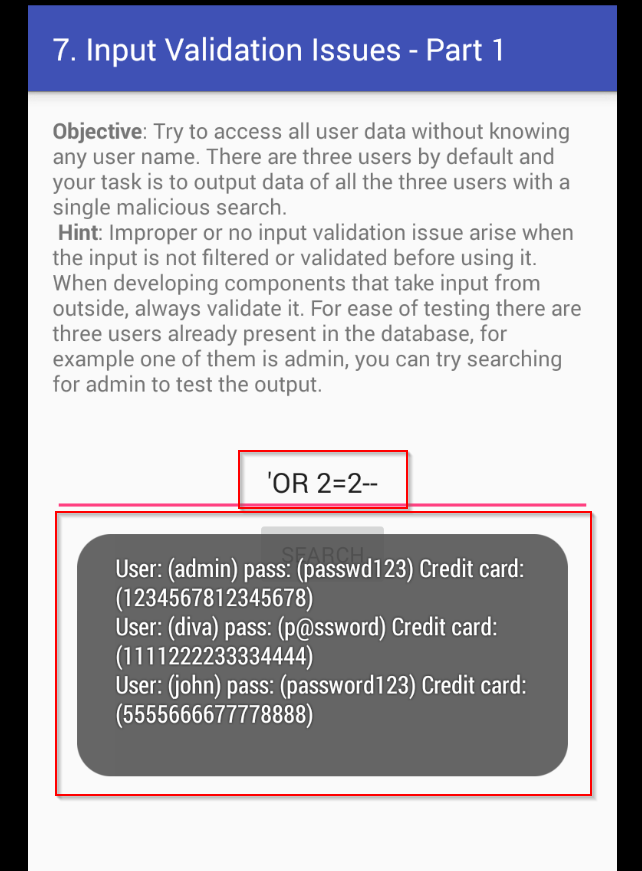
User have to save their username and password in application. As application is using Sharedpreferences in string data type. As codes Username and Passwords are stored in **tmp** file named as **uinfo**

**path**: /data/data/jakhar.aseem.diva

**6.INSECURE DATA STORAGE PART - 4**

User have to save their username and password in application. As application is using Sharedpreferences in string data type. As codes Username and Passwords are stored in **/.uinfo.txt** directory its under root directory . All the credential are stored in plaintext.

**Path**: cd /sdcard

**7.INPUT VALIDATION PART -1**

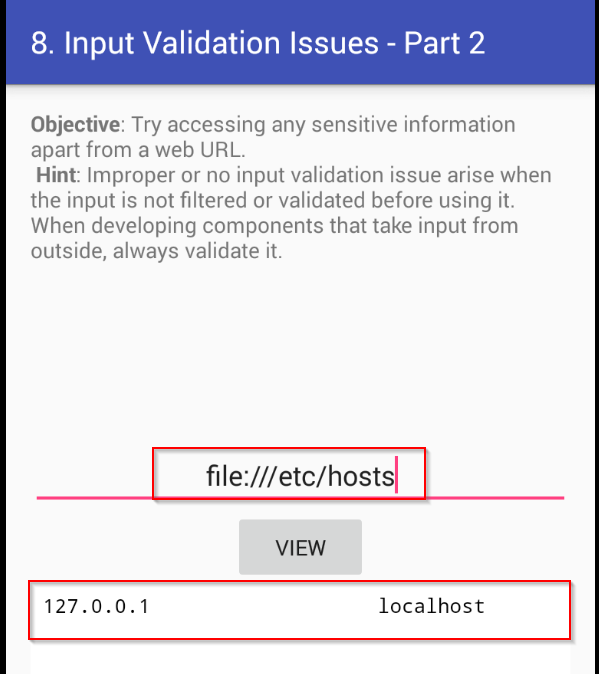
With the help of SQL command be where able to display the users’ credentials

**Query**: ‘ OR 2=2 --

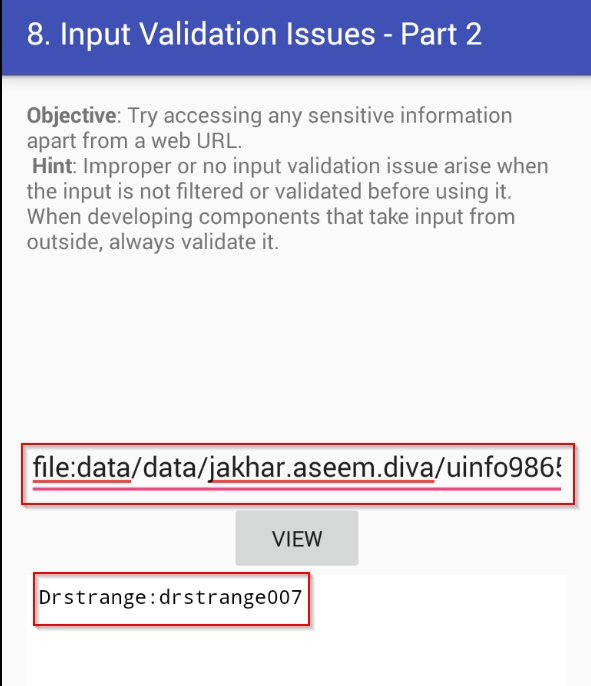
**8.INPUT VALIDATION PART -2**

With the help of this page, we were able to retrieve sensitive information of application from different locations

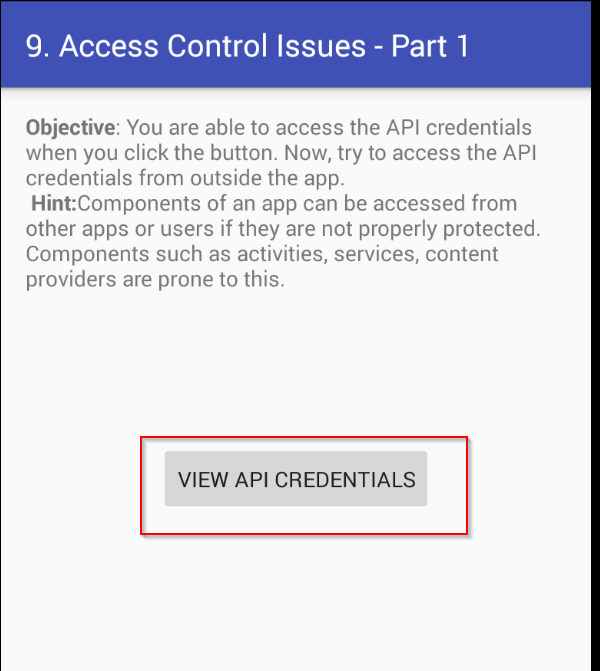
**Code**: file:///etc/hosts



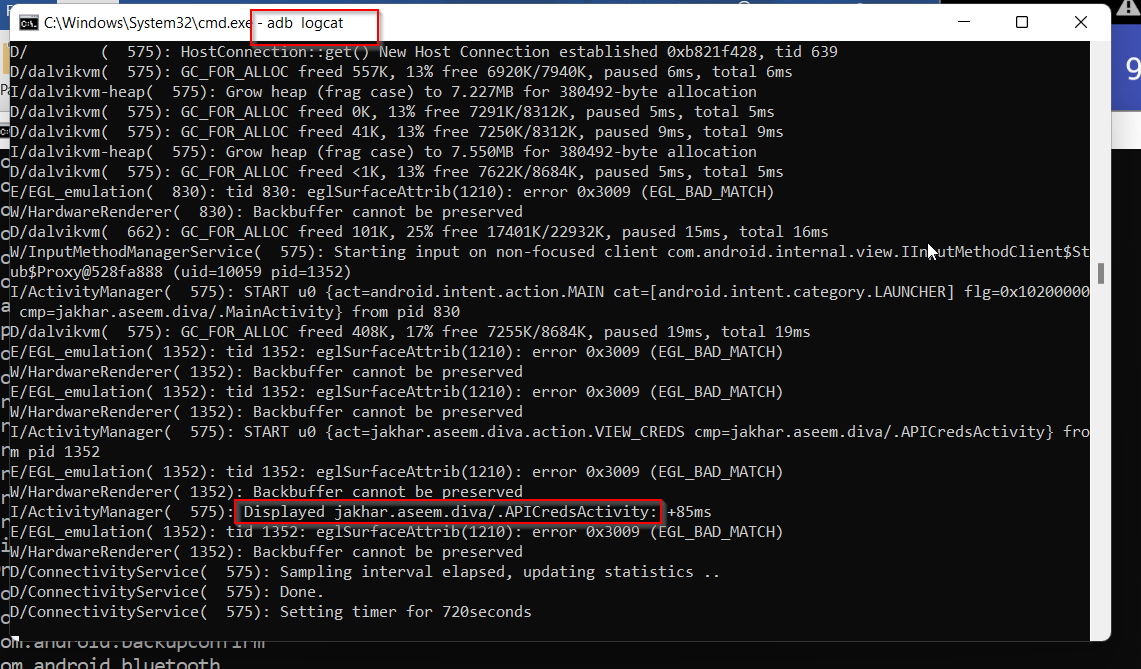
**Code** = file: data/data/jakhar.aseem.diva/uinfo986587018tmp



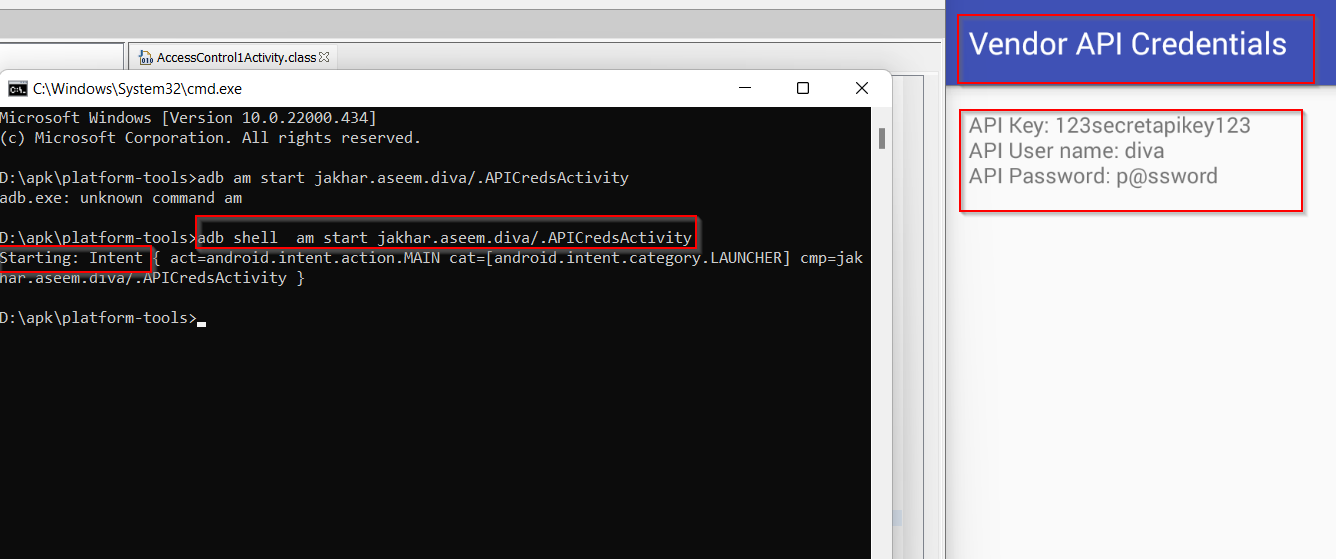
**9.ACESS CONTROL ISSUES PART-1**



After clicking on **VIEW API CREDENTIALS** button , we took look in logcat to see which activities are used and can try to access them.

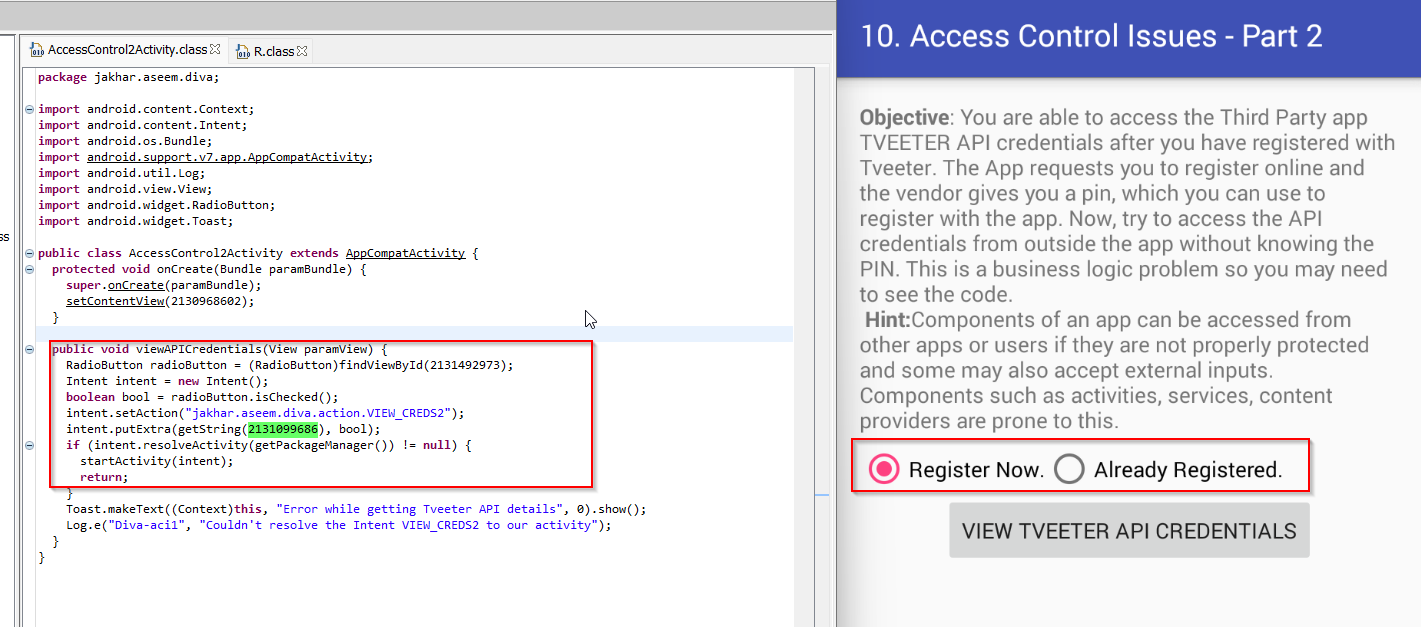


As we can see **jakhar.aseem.diva/.APICredsActivity** is used.

Now we know which activity is called to view API Credentials. We will use **Activity Manager**(am) to directly call that activity without going inside application

**Code**: adb shell am start jakhar.aseem.diva/.APICredsActivity

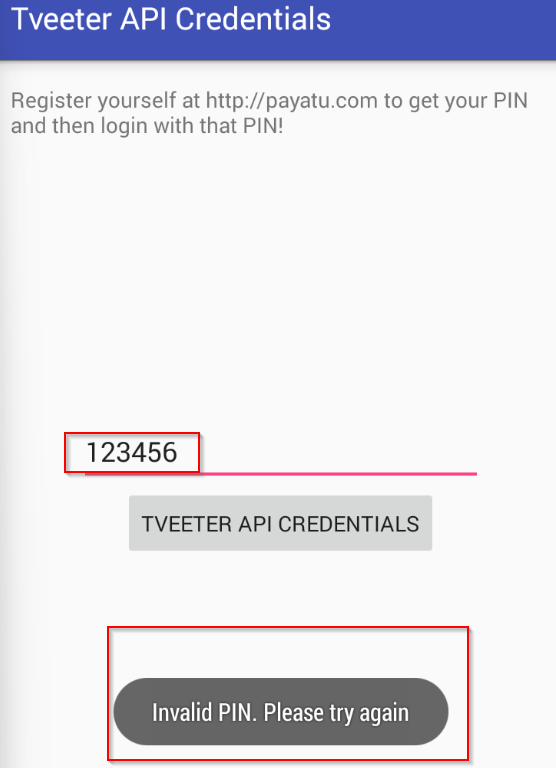
Right side of image we can see activity is called using AM

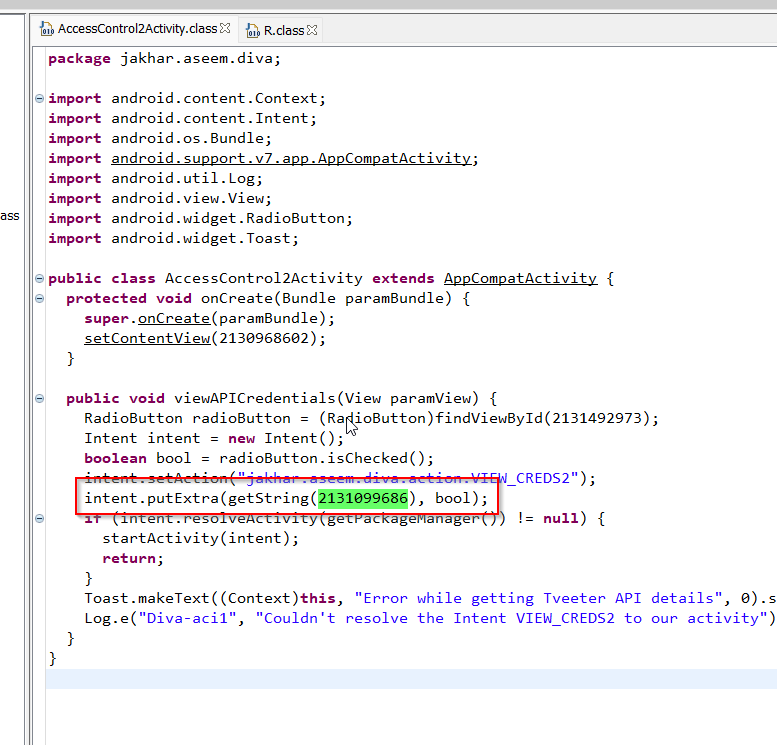
**10. ACCESS CONTROL ISSUES PART-2**

Clicking on the ‘**Already Registered’** provides us with the API credentials, username and password.

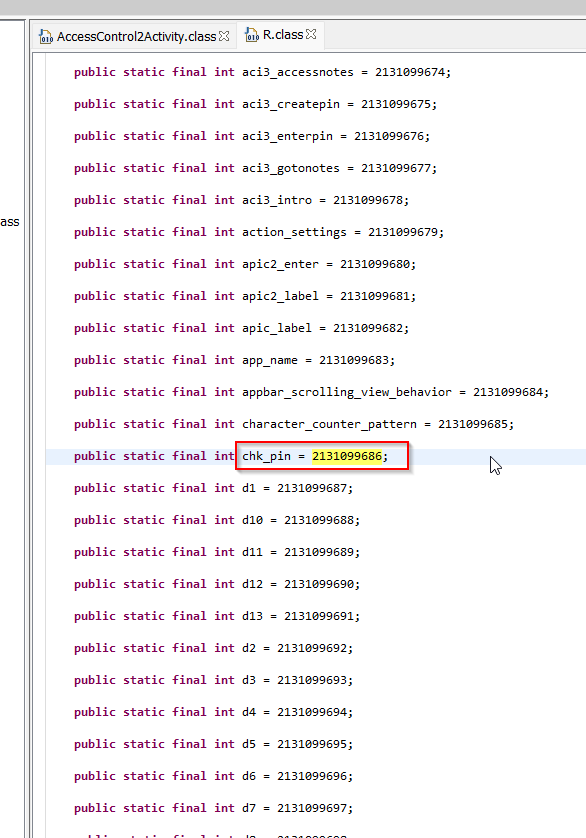
Clicking on the ‘**Register Now**’ asks us to enter the pin

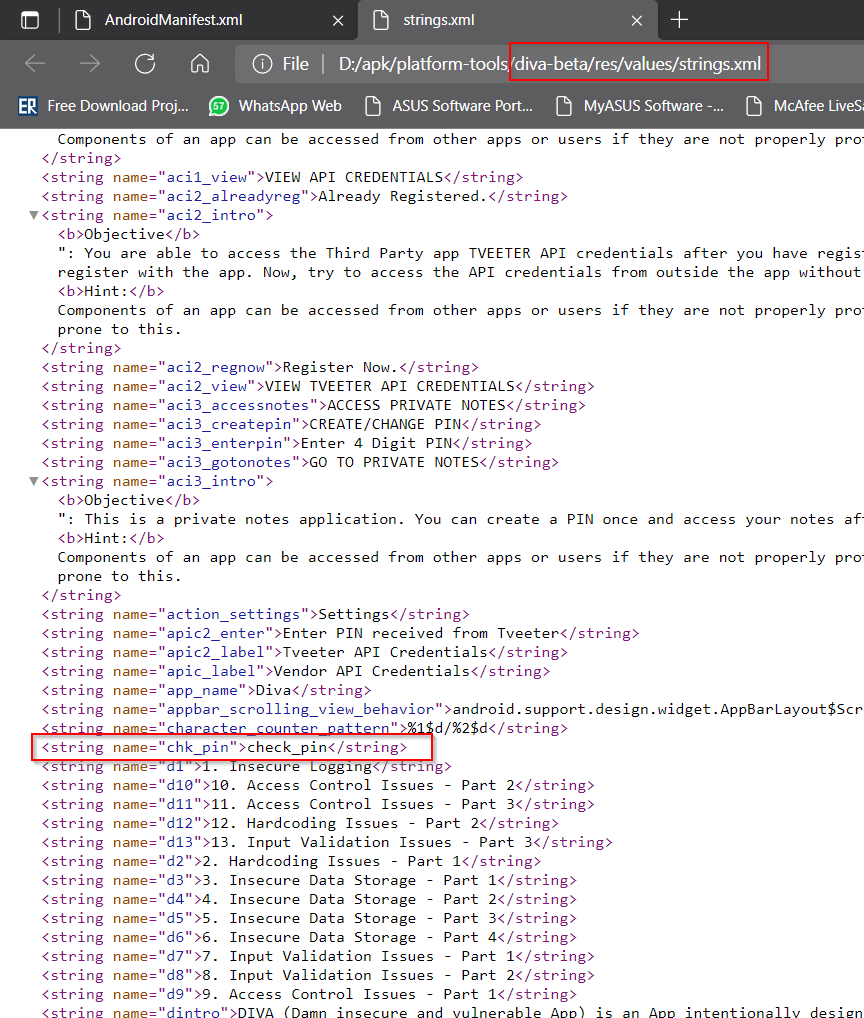
As per code If input should be not equal to 0 values to get access, but we try **123456** but application says **Invalid pin, please try again.**



we got ID (**2131099686**) in source code; we will check value of it inside **R.class** file

Id (**2131099686)** is used for chk\_pin

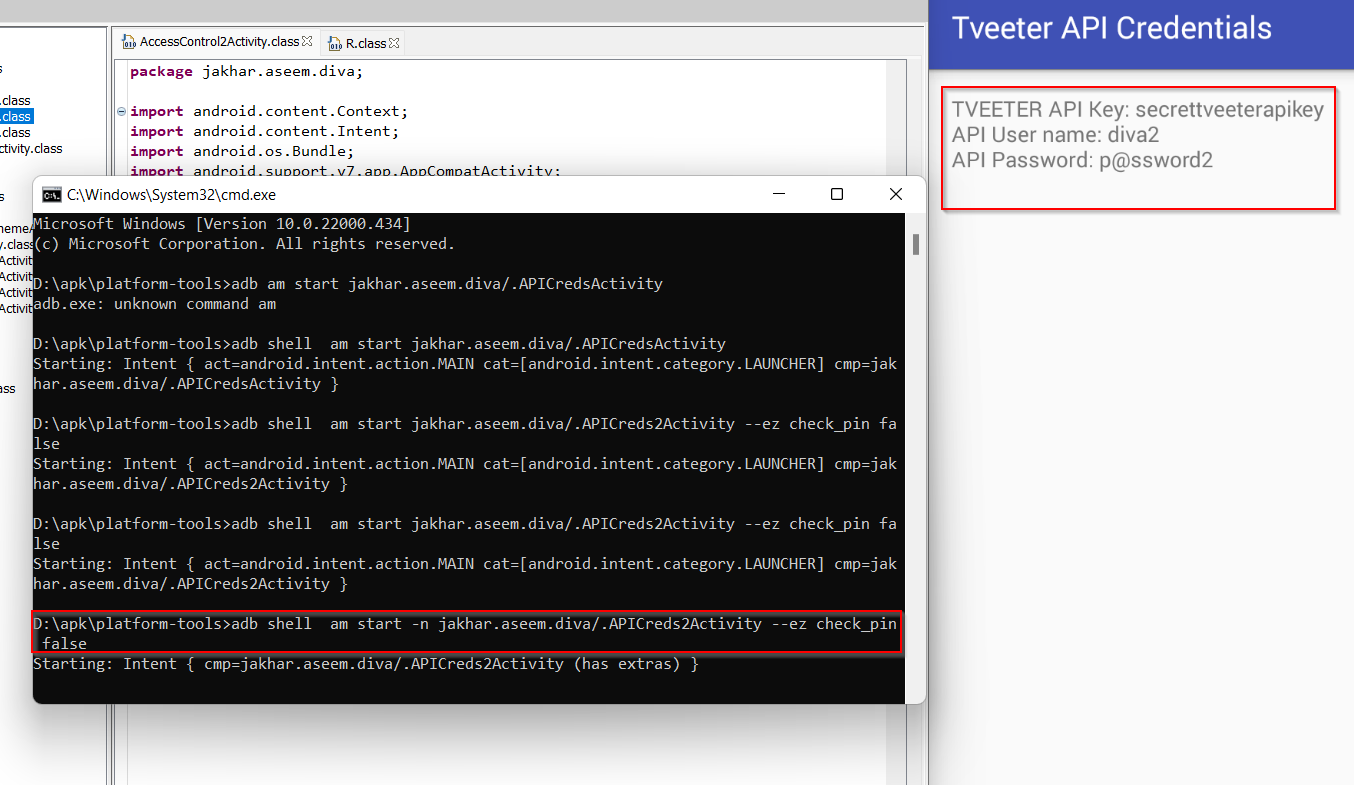


we notice actually value of **chk\_pin** is **check\_pin**

**path**: deva-beta/res/values/strings.xml

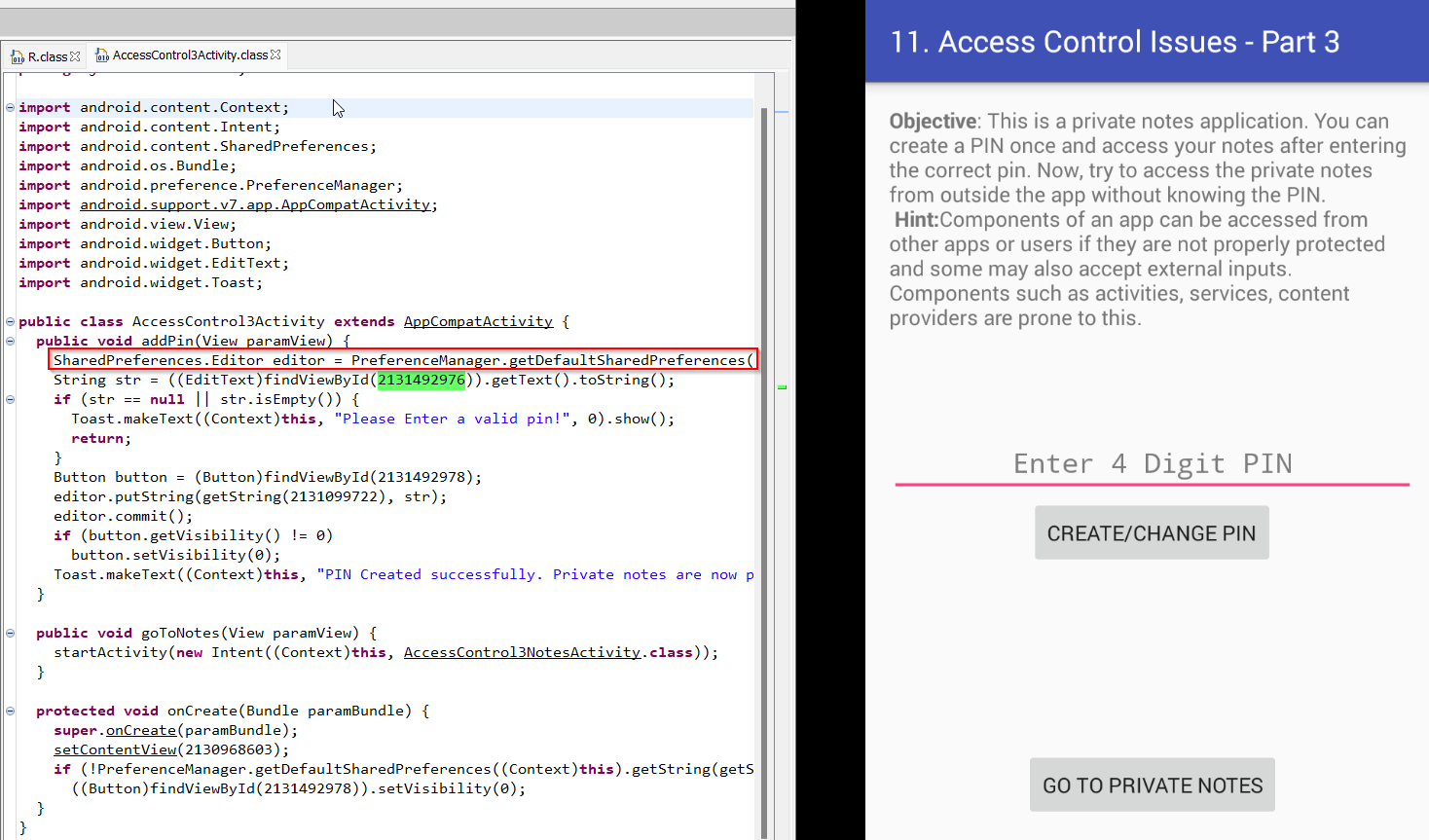
Now we know check\_pin is checking pin to get access, will disable its value. it takes **Boolean** values default it is true we will make it **false** using **AM**

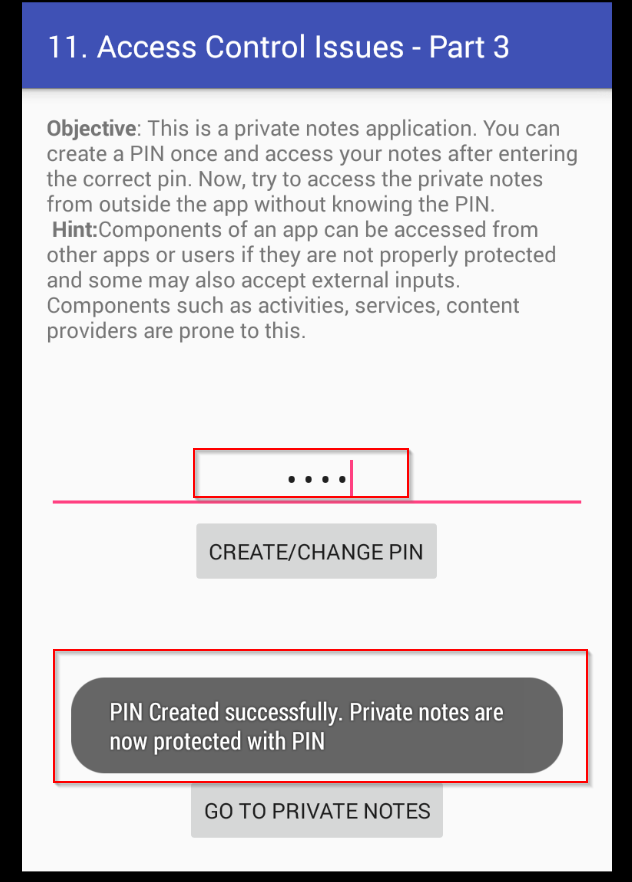
**Code**: adb shell am start -n jakhar.aseem.diva/.APICreds2Activity --ez check\_pin false

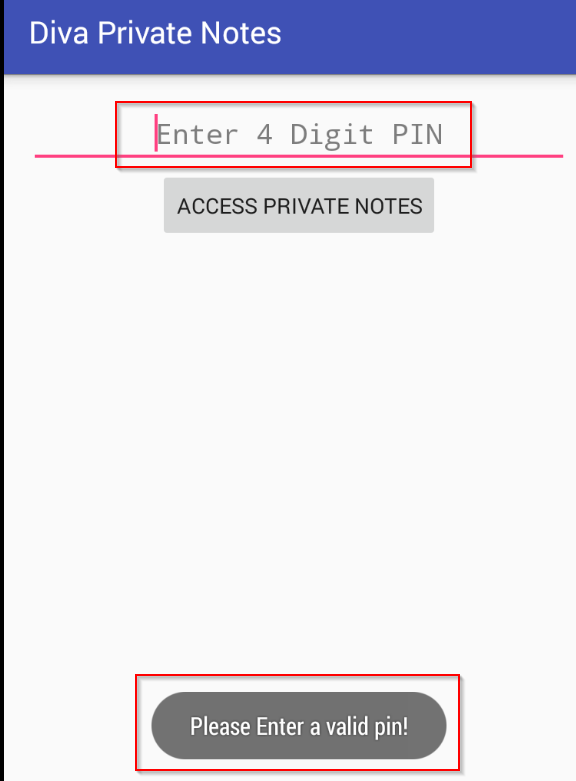
**--ez** <data> <Boolean> **-n** input activity name to be carried out

We have successfully bypassed Requirement of API Credentials

**11.ACCESS CONTROL ISSUES PART-3**

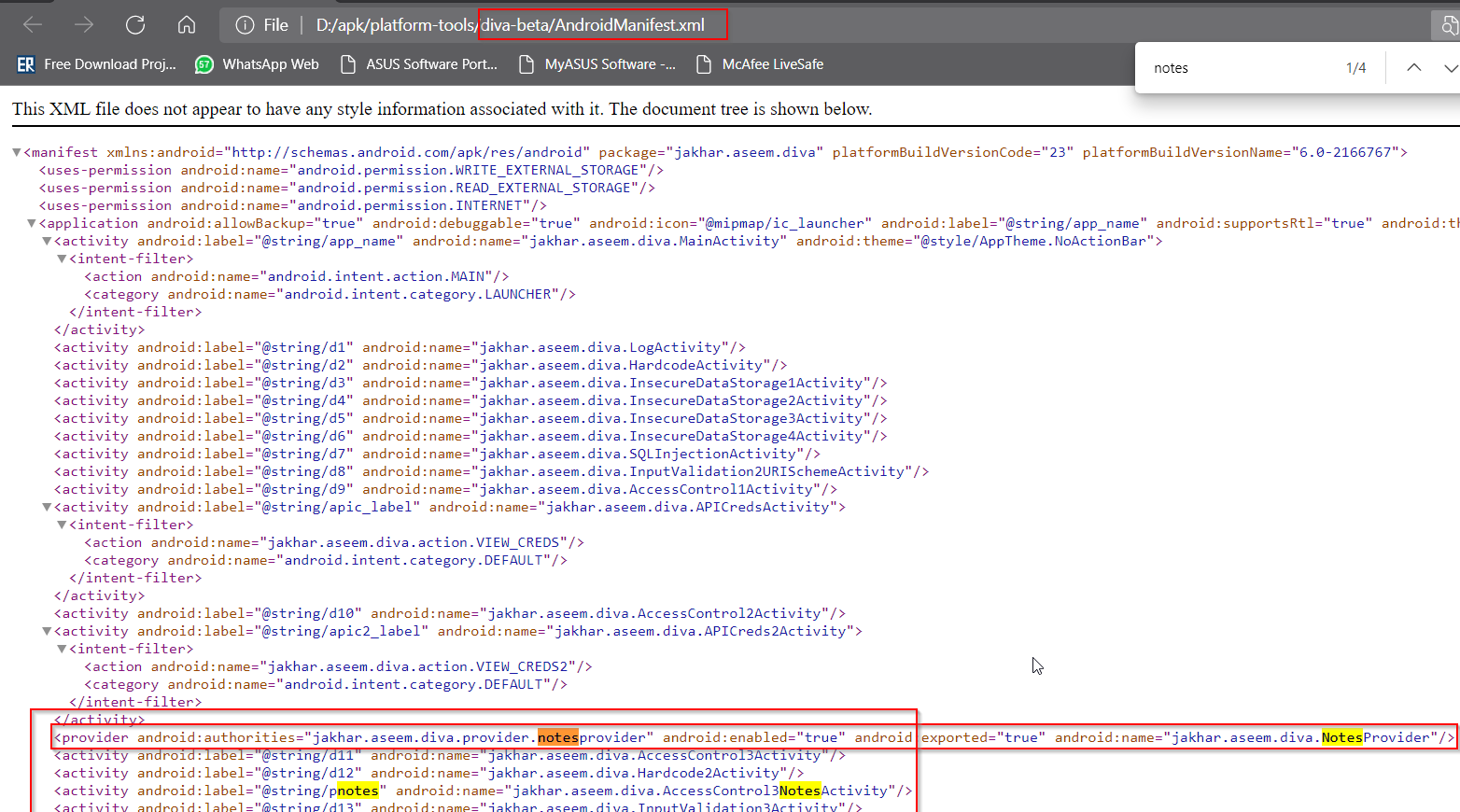
In this page application ask for pin to view Notes of users

we created new pin for us



when we try to access private notes, it says **Please enter valid pin**

as of sources code this page is using Sharedpreferences. We try to find activities of this notesprovider inside AndroidManifest.xml file

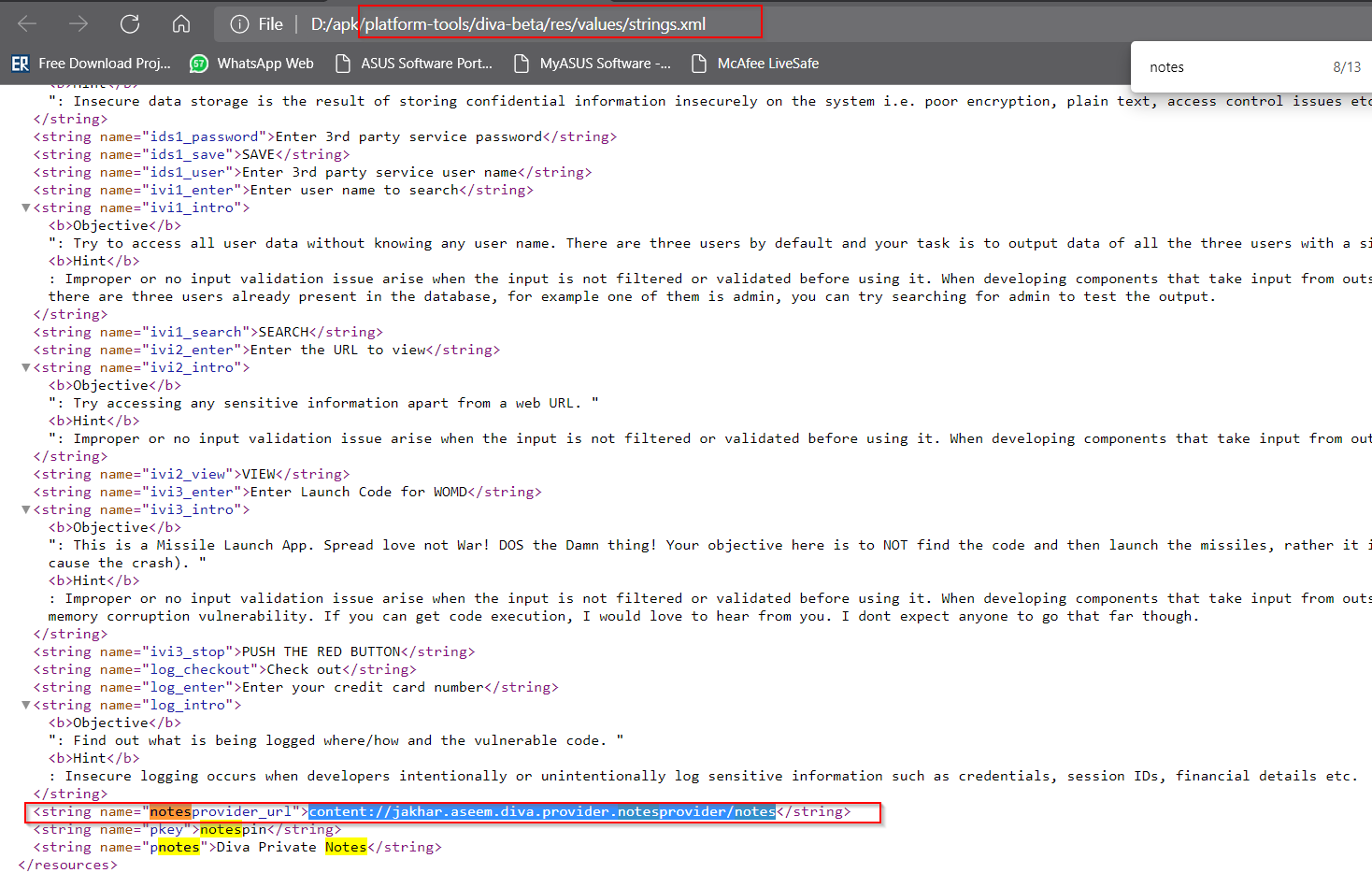


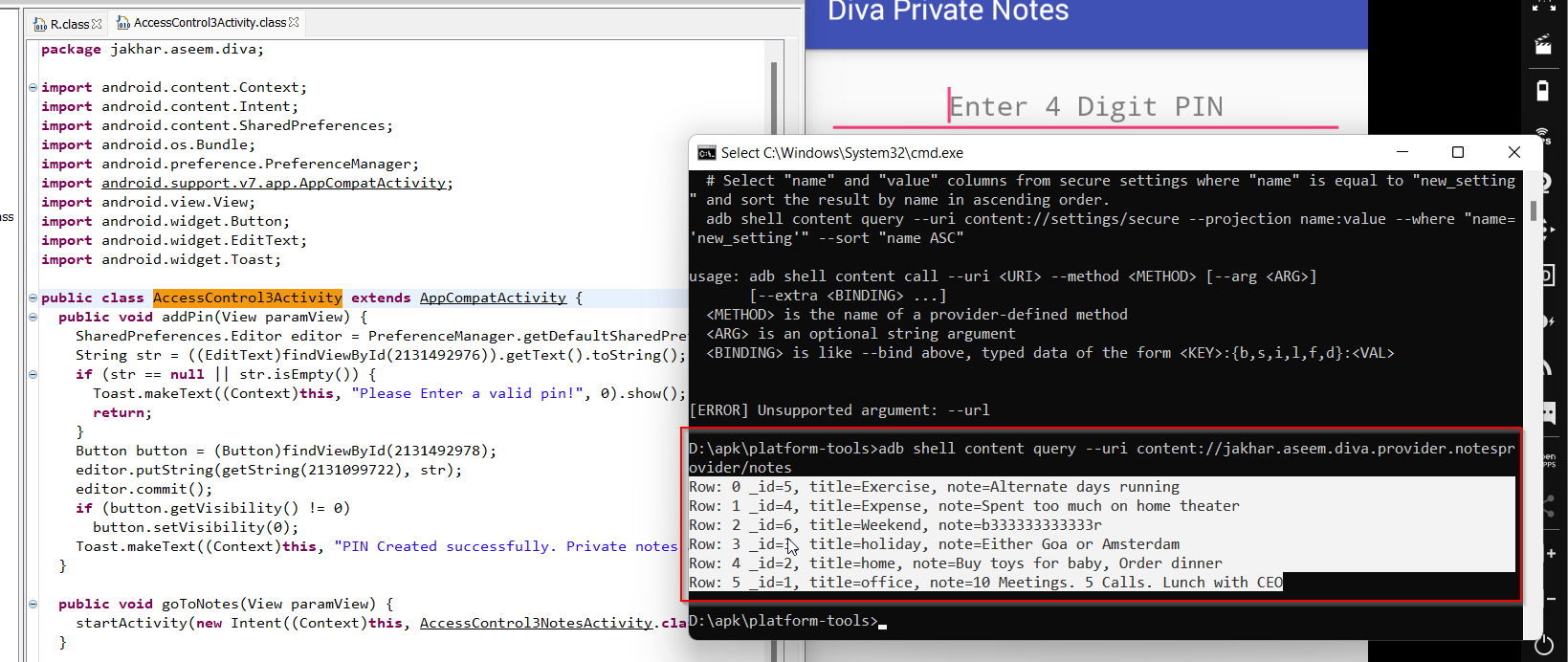
The AndroidManifest.xml shows the

content provider **jakhar.aseem.diva.provider.notesprovider; android:enabled=“true” and android:exported=“true”** which means the components of other applications can have access to it.

**Path**: diva-beta/res/values/string.xml

In this page we got actually path where notes are stored.

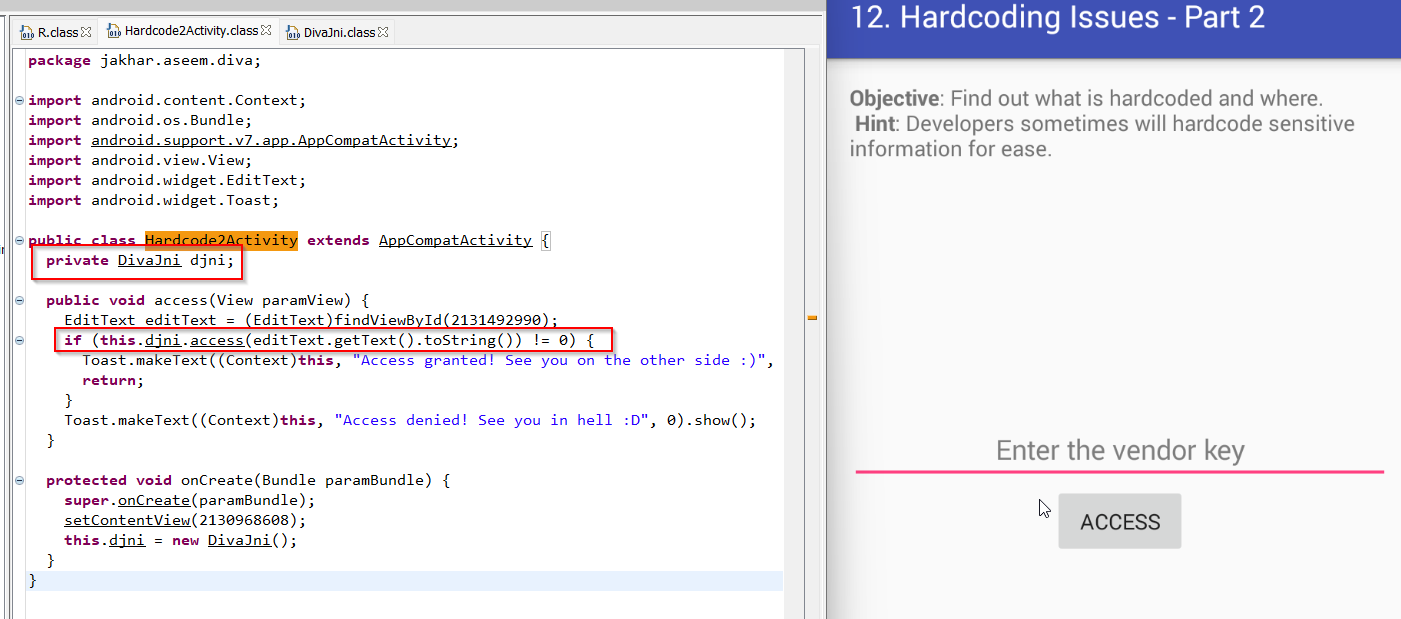


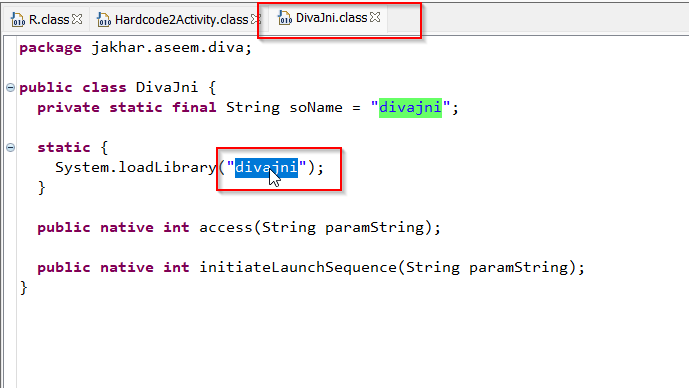
**Code**: adb shell content query –uricontent://jakhar.aseem.diva.provider.notesprovider/notes

**12.HARDCODING ISSUES PART-2**

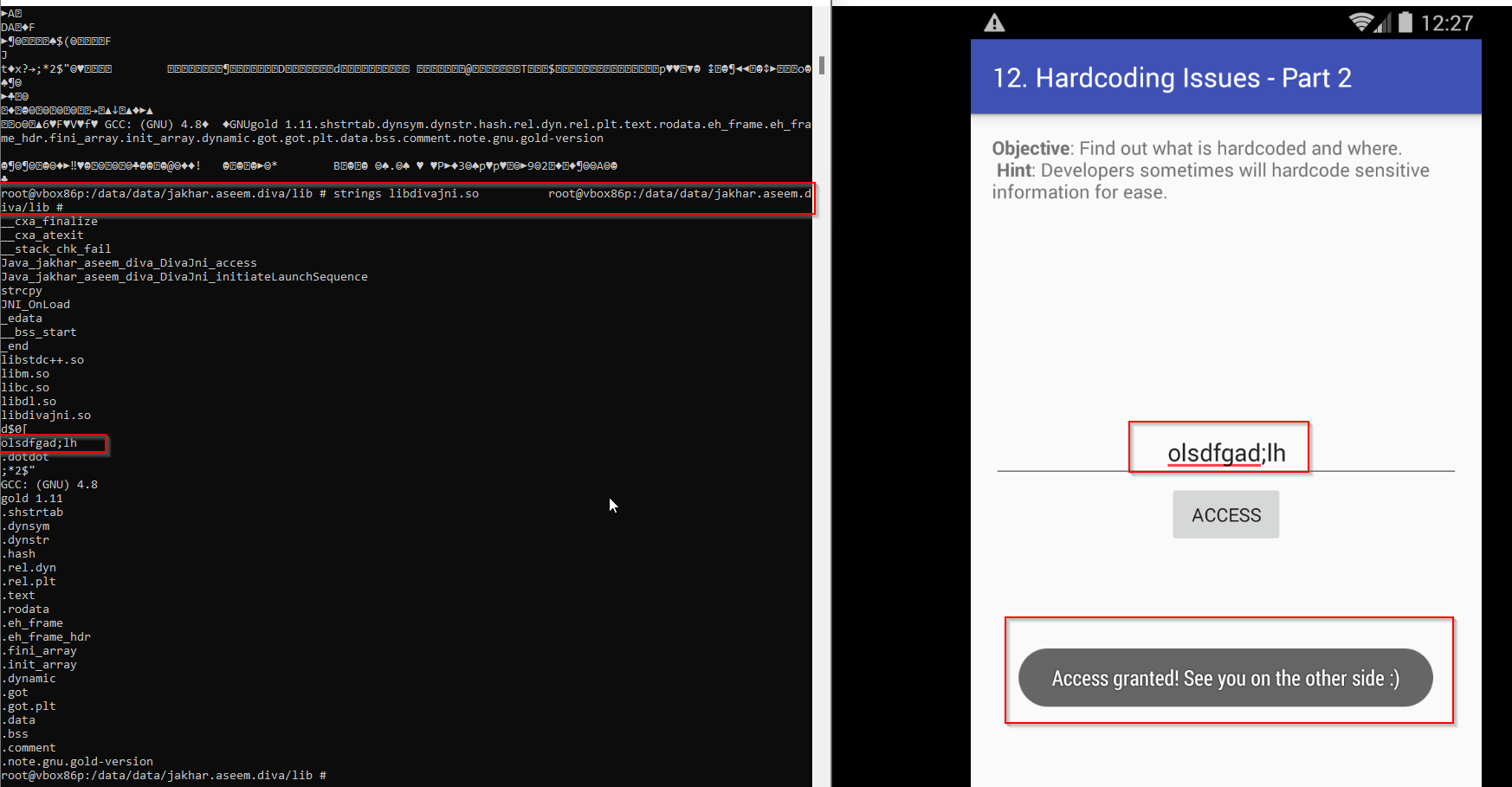
We have to find access code of vender

Here Divajni library is used.





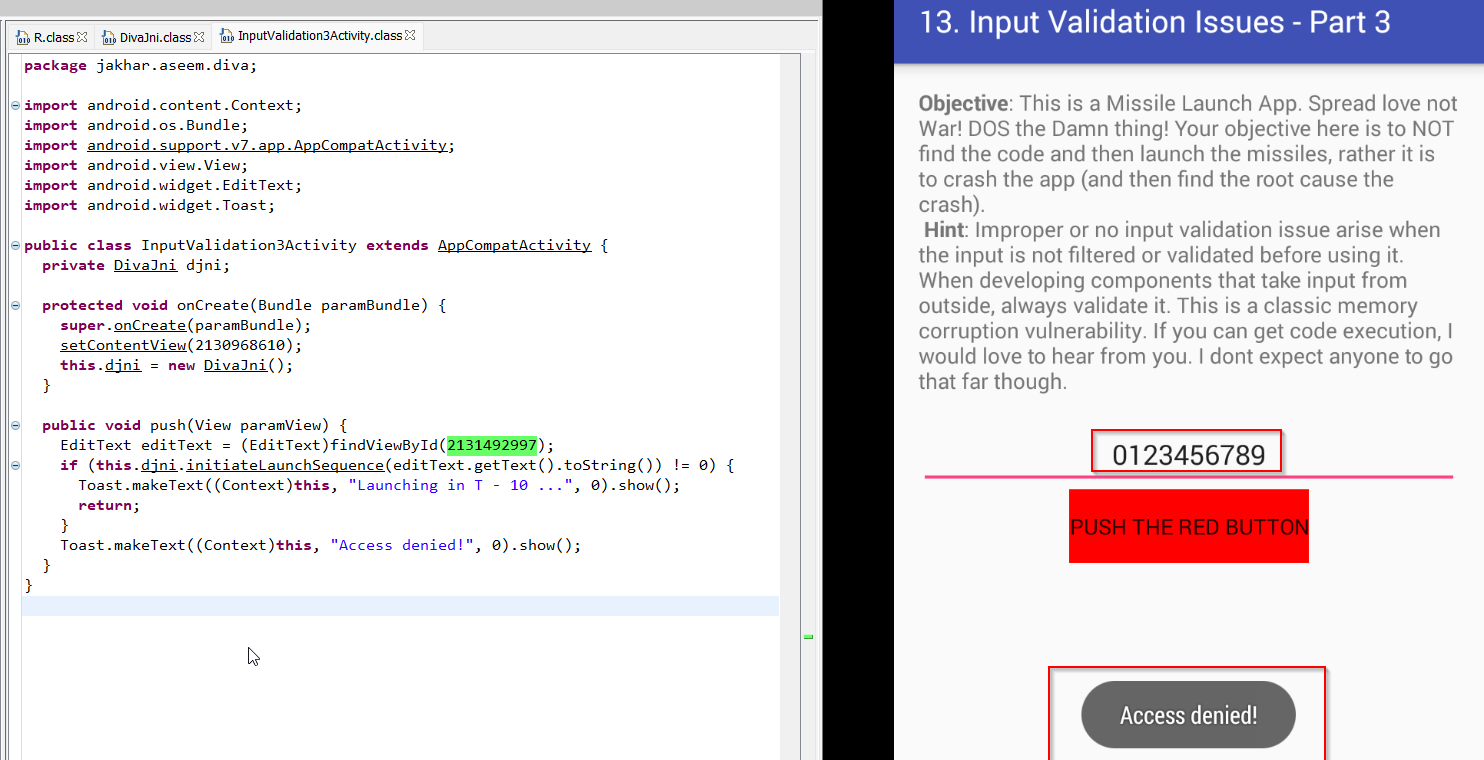
We will try find this library with shell

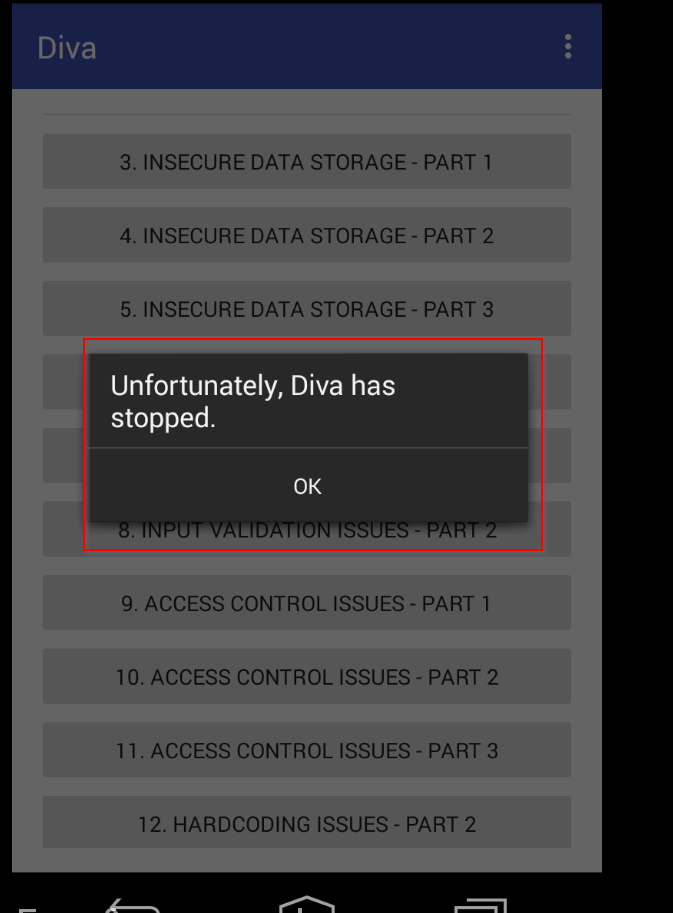
**path**: /data/data/jakhar.aseem.diva/lib

**code**: strings libdivajni.so

we got **olsdfgad;lh** as access code

**13.INPUT VALIDATION ISSUES PART-3**

As per source code we should get access if values are not equal to 0, but we got **Access denied!**

we put 20 numbers in input field and application crash. Application is vulnerable to **buffer overflow** .