Activities Access

Background

Android activity navigation is quite like how Internet navigation works. In web-based applications, many pages are public. For example, we could access many pages by adding its name after the domain name, like so: http://host/home.html. In Android, we could do the same thing -- we could navigate to any activity using "am" package commands.

This could be quite the security problem. There are certain activities whose access should be restricted. Take a bank app that has two activities: the first activity for login and the second activity for making transactions the user must login to do transactions.

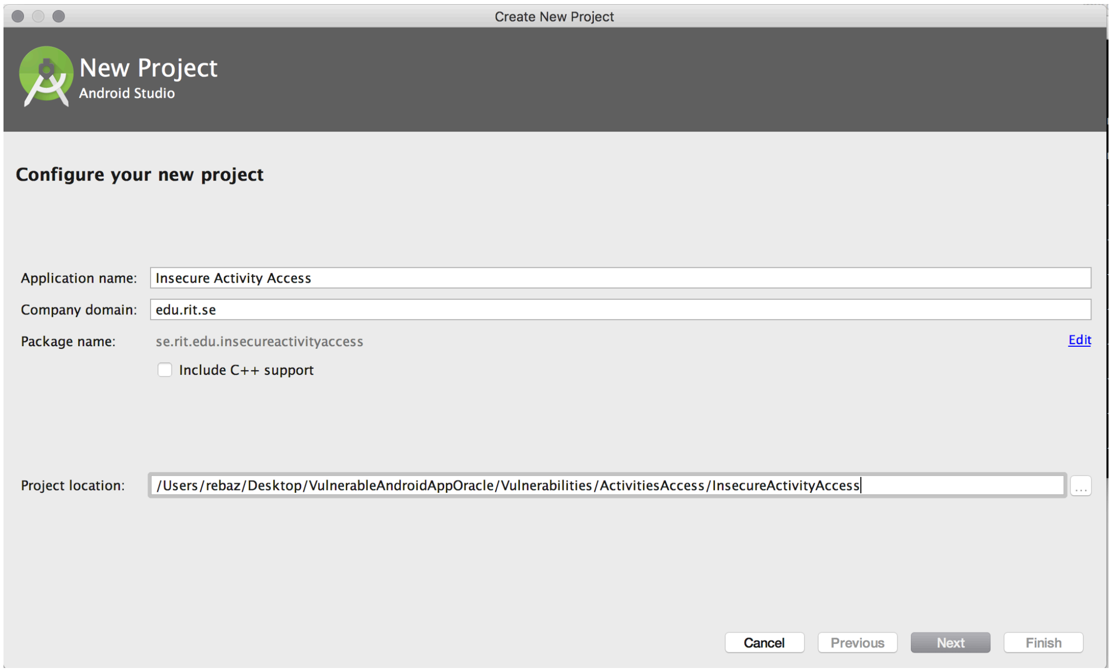
The security weakness here is that hackers could possibly circumvent login and navigate to the restricted-access activities by using "am" package commands.

Activity Instructions

We will illustrate the problem by creating an app and exploiting it ourselves.

i. Create an app that asks the user to enter a username and password. Upon correct entry of login credentials, the user will be redirected to the second page to change their password.
ii. Show how an attacker might access the activity to change the user password without login.
iii. Explain techniques we might use to defend ourselves.

1. Project Creation
   a. Follow the screens below to create a new project:



Configure your new project

Application name: Insecure Activity Access

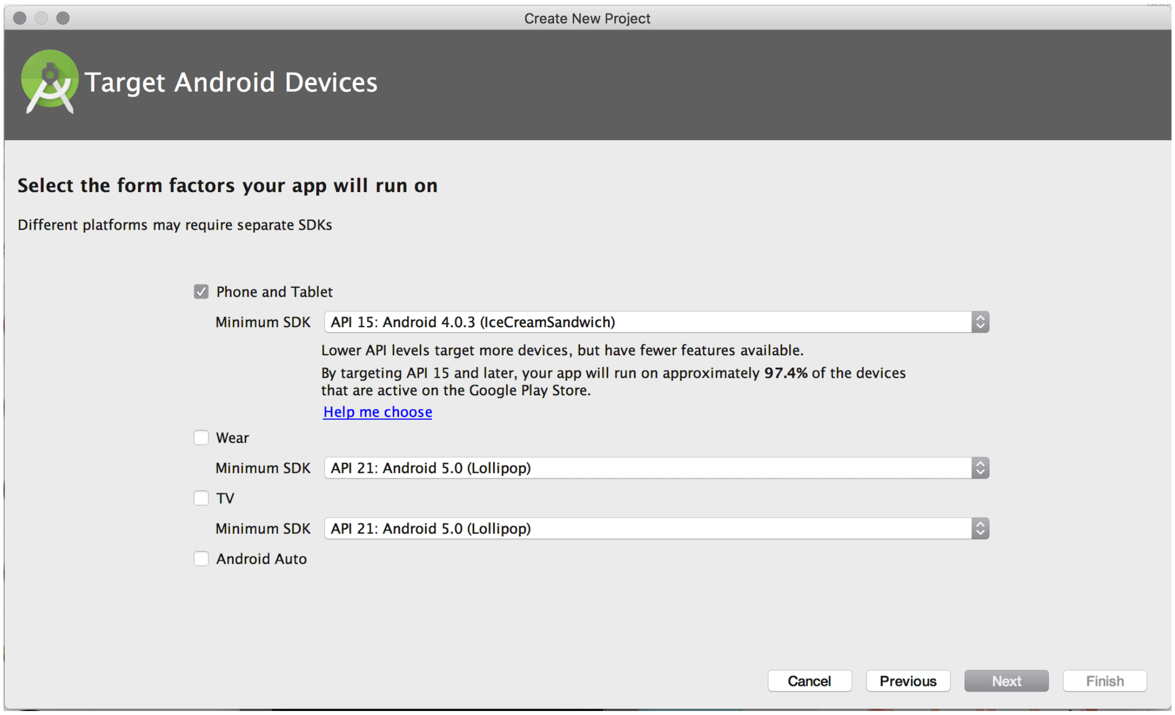
Company domain: edu.rit.se

Package name: se.rit.edu.insecureactivityaccess

□ Include C++ support

Project location: /Users/rebaz/Desktop/VulnerableAndroidAppOracle/Vulnerabilities/ActivitiesAccess/InsecureActivityAccess

Name the project “Insecure Activity Access”.



Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available.
By targeting API 15 and later, your app will run on approximately **97.4%** of the devices that are active on the Google Play Store.

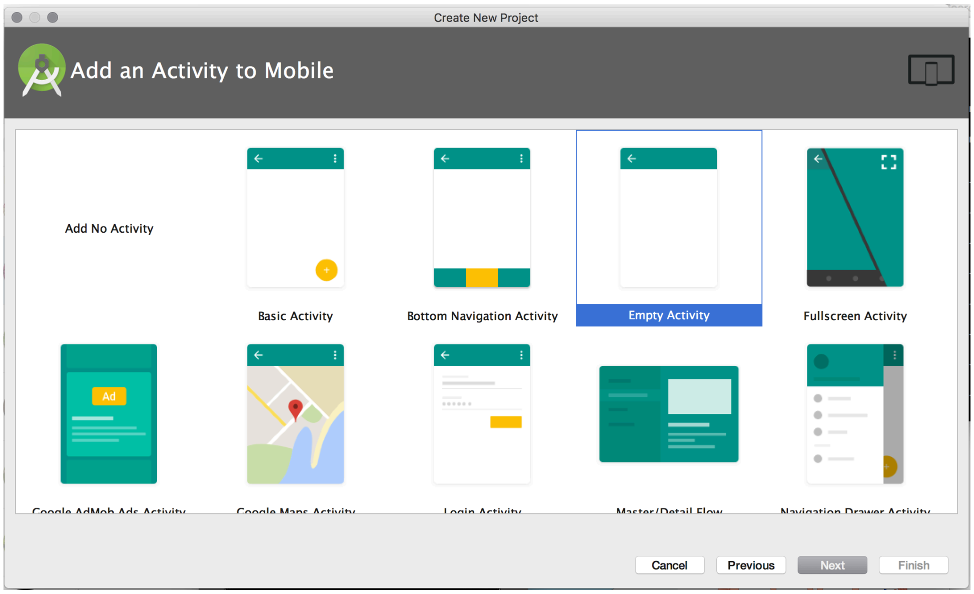
□ Wear

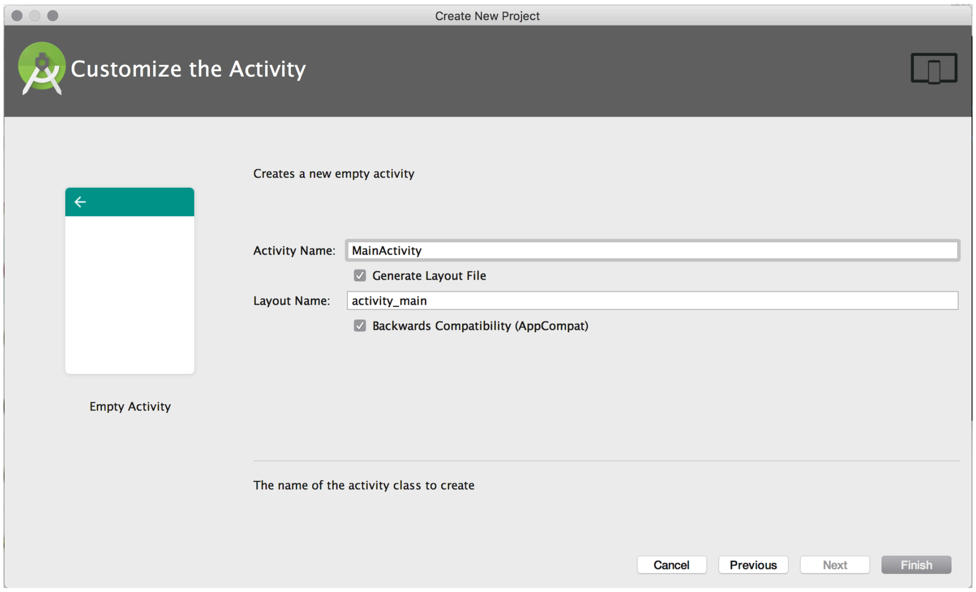
Minimum SDK API 21: Android 5.0 (Lollipop)

□ TV

Minimum SDK API 21: Android 5.0 (Lollipop)

□ Android Auto





Creates a new empty activity

Activity Name: MainActivity

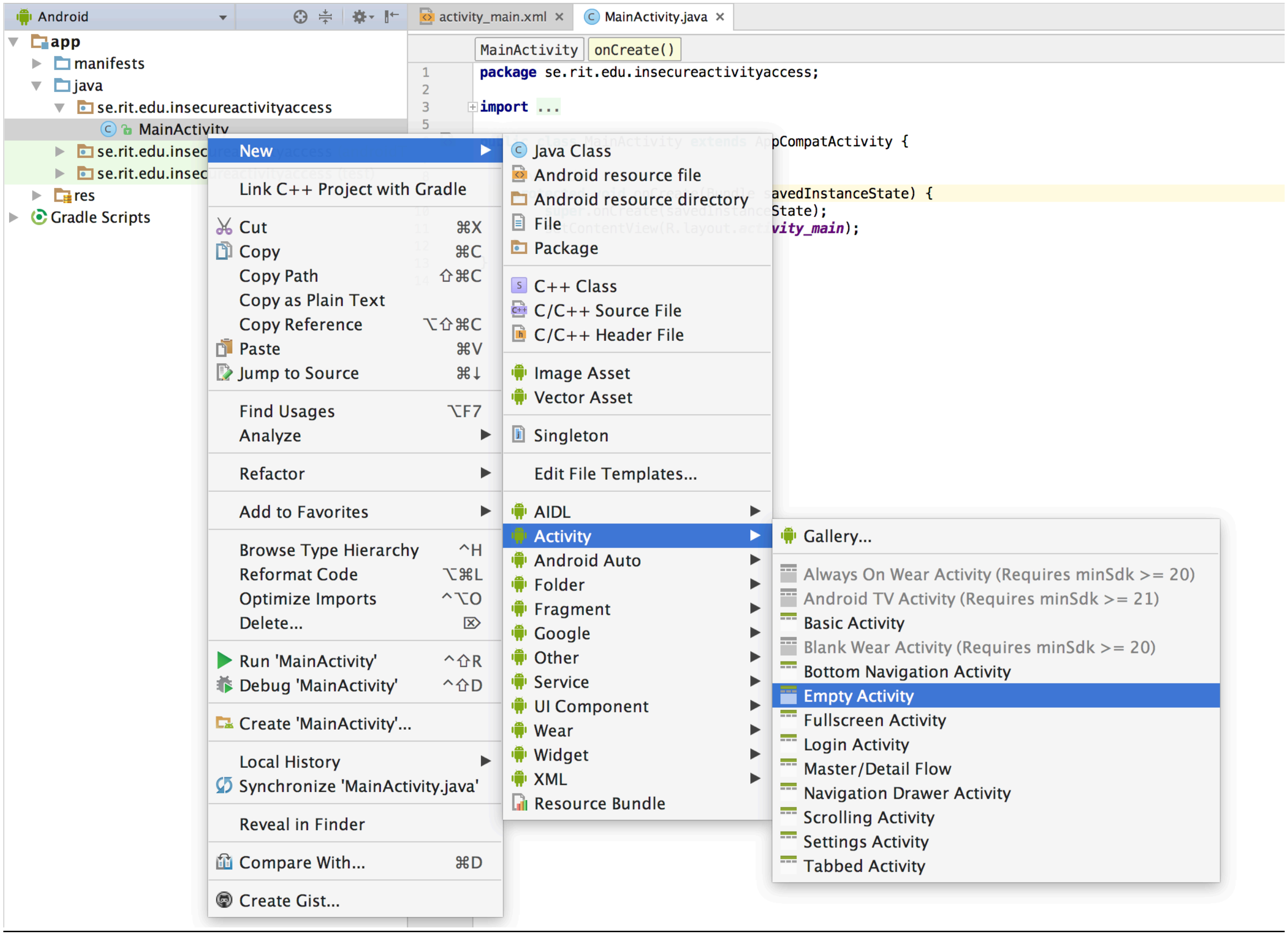
☑ Generate Layout File

Layout Name: activity_main

☑ Backwards Compatibility (AppCompat)

The name of the activity class to create

2. <u>Add an Activity</u>
   a. Add a new activity called “Main2Activity.java” by clicking on “ActivityMain.java”, found under “app/java/package_name_here/MainActivity” as shown in the image below



```
package se.rit.edu.insecureactivityaccess;

import ...
```

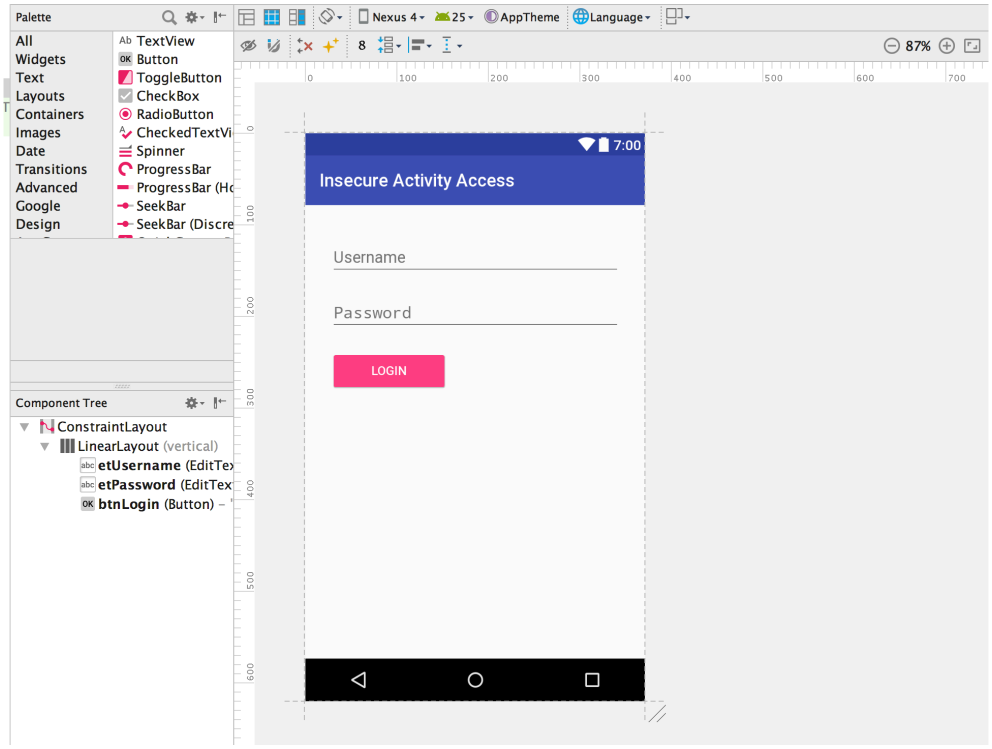
3. Construct User Interface

a. Open activity_main.xml and clear all and paste the following code

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="30dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
```

```
        tools:context=".MainActivity">

        <LinearLayout
          android:layout_width="328dp"
          android:layout_height="495dp"
          android:layout_weight="1"
          android:orientation="vertical"
          tools:layout_editor_absoluteX="8dp"
          tools:layout_editor_absoluteY="8dp">

          <EditText
            android:id="@+id/etUsername"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Username"
            android:inputType="textPersonName" />

          <EditText
            android:id="@+id/etPassword"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:ems="10"
            android:hint="Password"
            android:inputType="textPassword" />

          <Button
            android:id="@+id/btnLogin"
            style="@style/Widget.AppCompat.Button.Colored"
            android:layout_width="133dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:text="Login" />
        </LinearLayout>

</android.support.constraint.ConstraintLayout>
```

This is what the layout should look like:

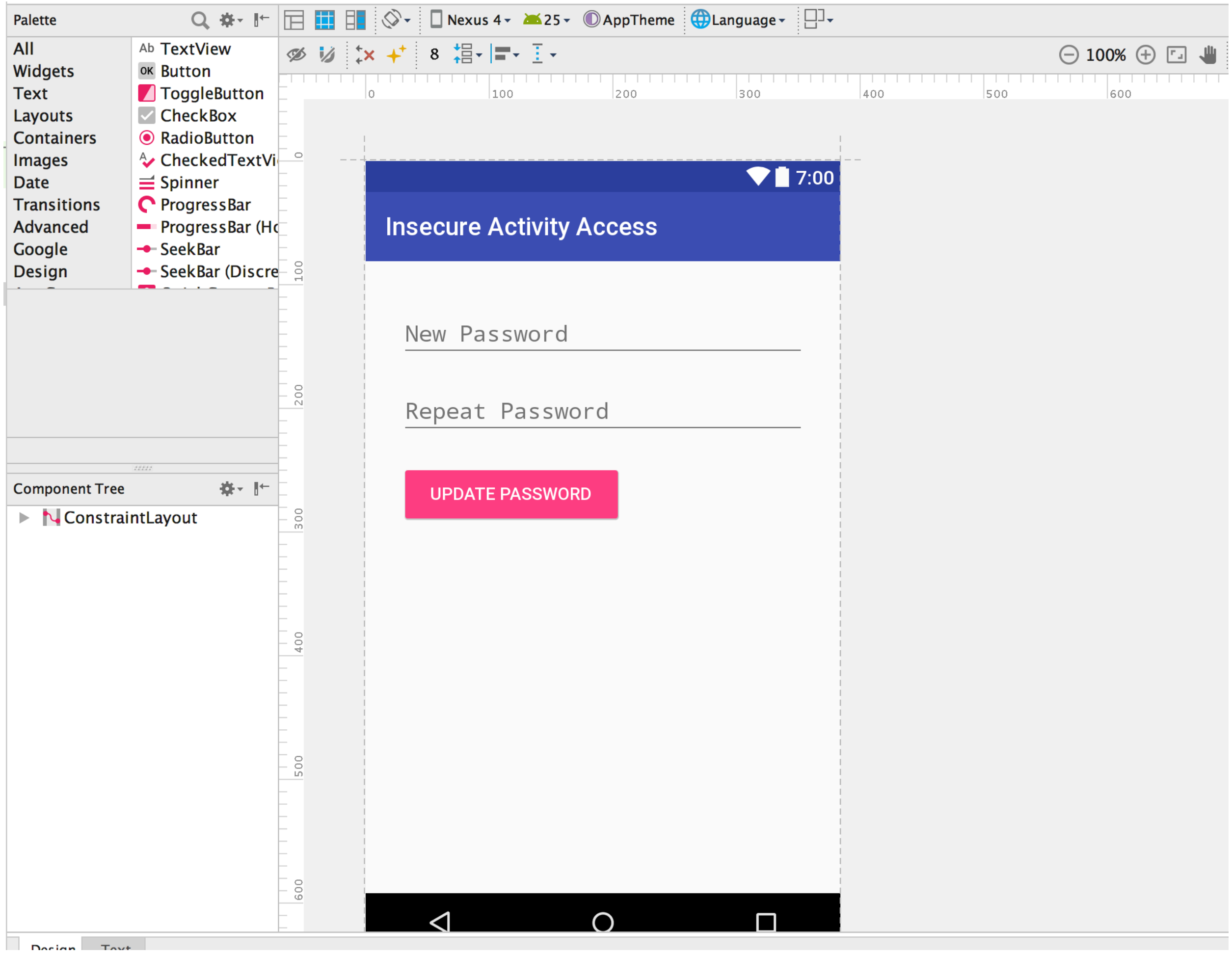


b. Construct the layout for the second activity by pasting the following to activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:layout_marginTop="30dp"
   android:layout_marginLeft="20dp"
   android:layout_marginRight="20dp"
   tools:context=".Main2Activity">

   <LinearLayout
      android:layout_width="328dp"
      android:layout_height="495dp"
      android:layout_weight="1"
      android:orientation="vertical"
```

```
    tools:layout_editor_absoluteX="8dp"
    tools:layout_editor_absoluteY="8dp"
    android:weightSum="1">

    <EditText
      android:id="@+id/etUsername"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:ems="10"
      android:hint="New Password"
      android:inputType="textPassword" />

    <EditText
      android:id="@+id/etPassword"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="20dp"
      android:ems="10"
      android:hint="Repeat Password"
      android:inputType="textPassword" />

    <Button
      android:id="@+id/btnLogin"
      style="@style/Widget.AppCompat.Button.Colored"
      android:layout_width="180dp"
      android:layout_height="wrap_content"
      android:layout_marginTop="20dp"
      android:text="Update Password"
      android:layout_weight="0.01" />
  </LinearLayout>

</android.support.constraint.ConstraintLayout>
```

The second activity should look like this



4. Code

Open MainActivity.java, found under “app/java/your_package_name”, and add the following code:

a. Add the following code inside the **onCreate** method:

```
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // username and password hardcoded for testing purpose
        final String USERNAME = "admin";
        final String PASSWORD = "admin";


        // initialize  user name  instance with the real input in xml
        final EditText etUsername =
(EditText)findViewById(R.id.etUsername);
```

```
    // initialize  password  instance with the real input in xml
    final EditText etPassword =
(EditText)findViewById(R.id.etPassword);

    // initialize login button instance
    final Button btnLogin = (Button)findViewById(R.id.btnLogin);
    btnLogin.setOnClickListener(new View.OnClickListener() {
      public void onClick(View v) {

        // collect user's username input
        String username = etUsername.getText().toString();

        // collect user's password input
        String password = etPassword.getText().toString();

        // compare values
        if(USERNAME.equals(username) &&
PASSWORD.equals(password))
        {
          Toast.makeText( MainActivity.this,
              "You are logged in successfully",
              Toast.LENGTH_LONG).show();

          Intent intent = new Intent(getApplicationContext(),
Main2Activity.class);
          startActivity(intent);
        }

        else {
          Toast.makeText( MainActivity.this,
              "Invalid credentials",
              Toast.LENGTH_LONG).show();
        }
      }
    });


  }
```

b. Add the following imports to the file, below the package declaration.

```
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

Open Main2Activity.java, and add the following code:

a. Add the following code inside the **onCreate** method:

```
        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main2);

            Bundle b = getIntent().getExtras();
        }
```

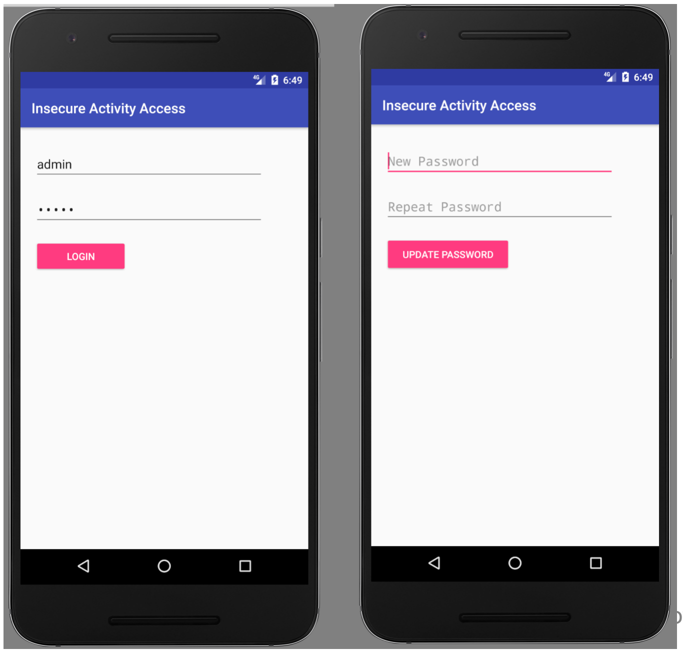
The above code achieves the following:

1. Upon creation of the MainActivity class, the **onCreate** method executes.
    a. Passes the Bundle named `savedInstanceState` to the superclass `AppCompatActivity`
    b. Initializes the variables declared in the first part with their corresponding layout objects.
    c. Creates a listener for the login button that upon click, will:
        i. Compare the credentials entered to the strings we have saved. If the credentials match what we have, we create an Intent with the second activity.
        ii. Start the second activity.
2. Upon creation of the Main2Activity class, the **onCreate** method executes.
    a. Passes the Bundle named `savedInstanceState` to the superclass `AppCompatActivity`
    b. Sets the content view to be the layout we designed for the second activity.
    c. Grabs the bundle that came with the Intent.

Exploitation Instructions

We shall see for ourselves how we can view the login credentials.

1. Run the app. Enter the login credentials of "admin" for username and "admin" for password (demonstration purposes only).



Insecure Activity Access

admin

•••••

Insecure Activity Access

New Password

Repeat Password

2. Using adb shell, view the saved preferences file by:
   - i. Open Terminal or Command Prompt.
   - ii. Run the following commands.
     - i. On Mac OS X:

       ```
       cd Library/Android/sdk/platform-tools
       ```

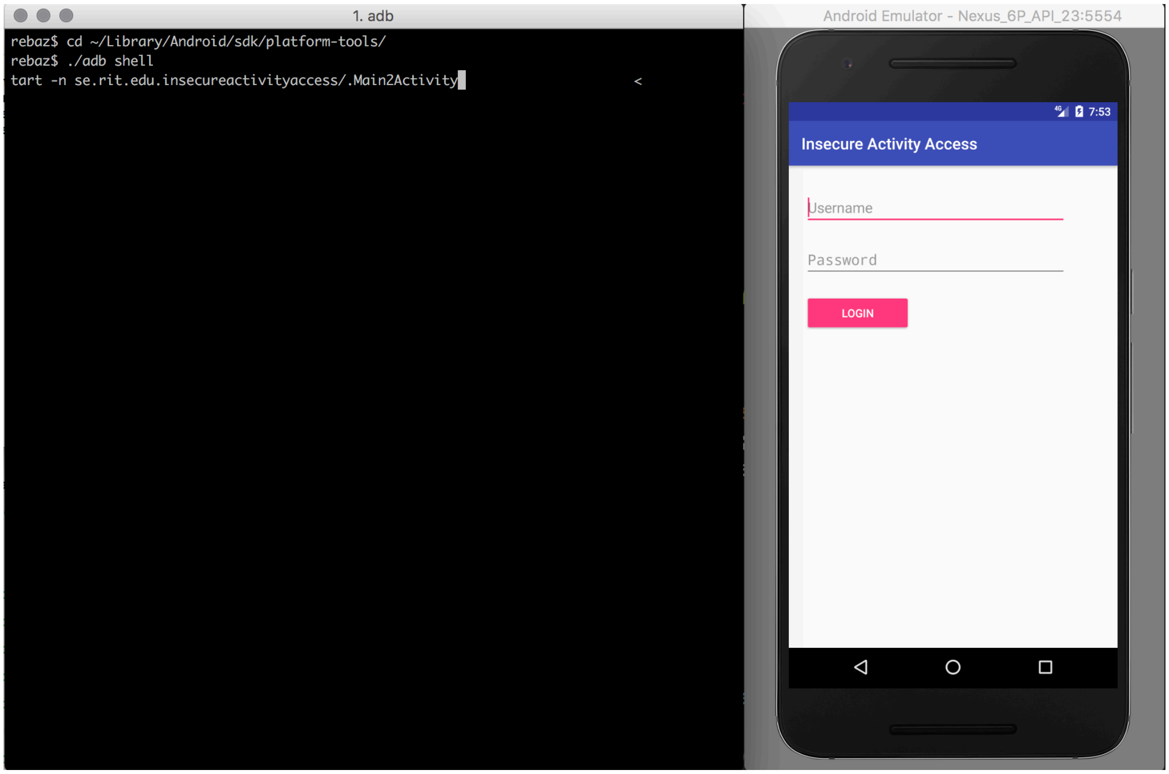
       On Windows:

```
cd C:\Users\YOUR_USERNAME_HERE\AppData\Local\Android\sdk\platform-tools
```

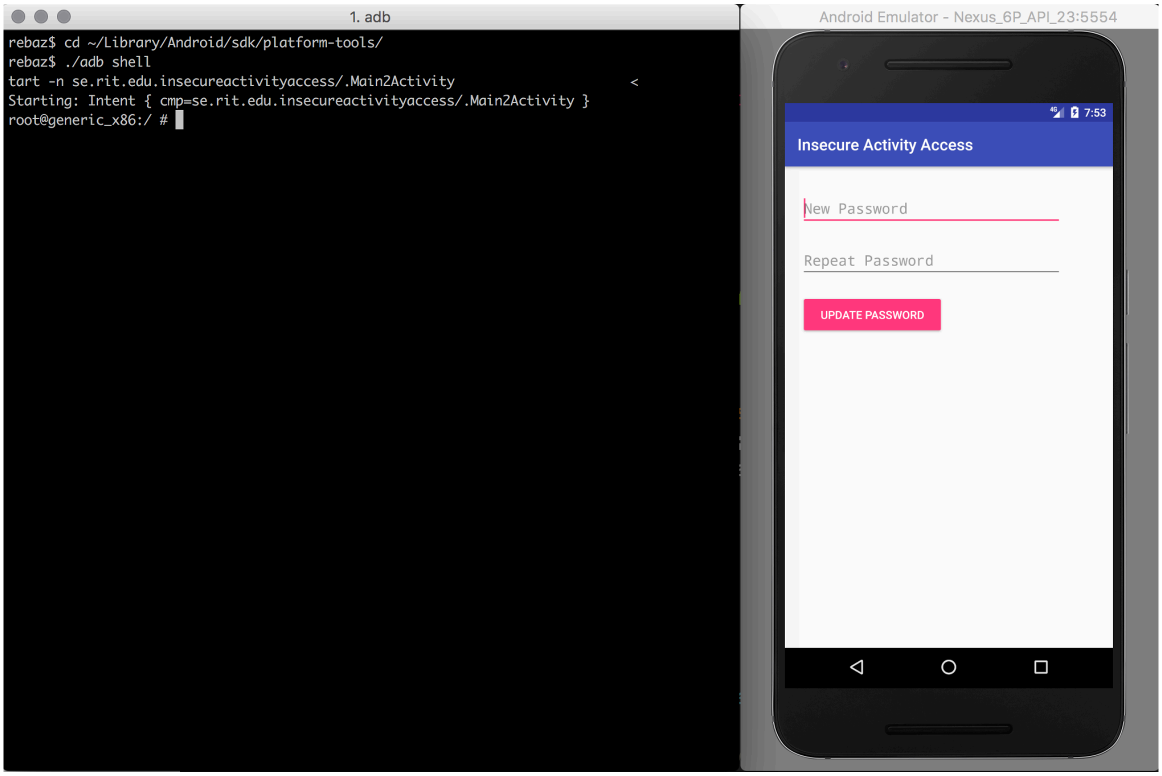
     - ii. From here, it doesn't matter what platform you are running this on. We simply needed to find the Android/sdk/platform-tools directory.

       ```
       ./adb shell
       ```

     - iii. `am start -n package_name/.ActivityName`
3. Once you have executed the commands above, you will be sent to the activity that is supposed to be after login only. We will be able to change the password without login.



```
rebaz$ cd ~/Library/Android/sdk/platform-tools/
rebaz$ ./adb shell
tart -n se.rit.edu.insecureactivityaccess/.Main2Activity                    <
```



```
rebaz$ cd ~/Library/Android/sdk/platform-tools/
rebaz$ ./adb shell
tart -n se.rit.edu.insecureactivityaccess/.Main2Activity                    <
Starting: Intent { cmp=se.rit.edu.insecureactivityaccess/.Main2Activity }
root@generic_x86:/ #
```

Defense

To fix this problem, we will send the key associated with the value over the intent to change password activity. In the second activity, we will then read the key and make sure the value is correct. If it is correct, we can start the password-changing activity. Otherwise, we will dismiss the activity. Then, when we run the "am" command without the key to open Main2Activity, it will not open.

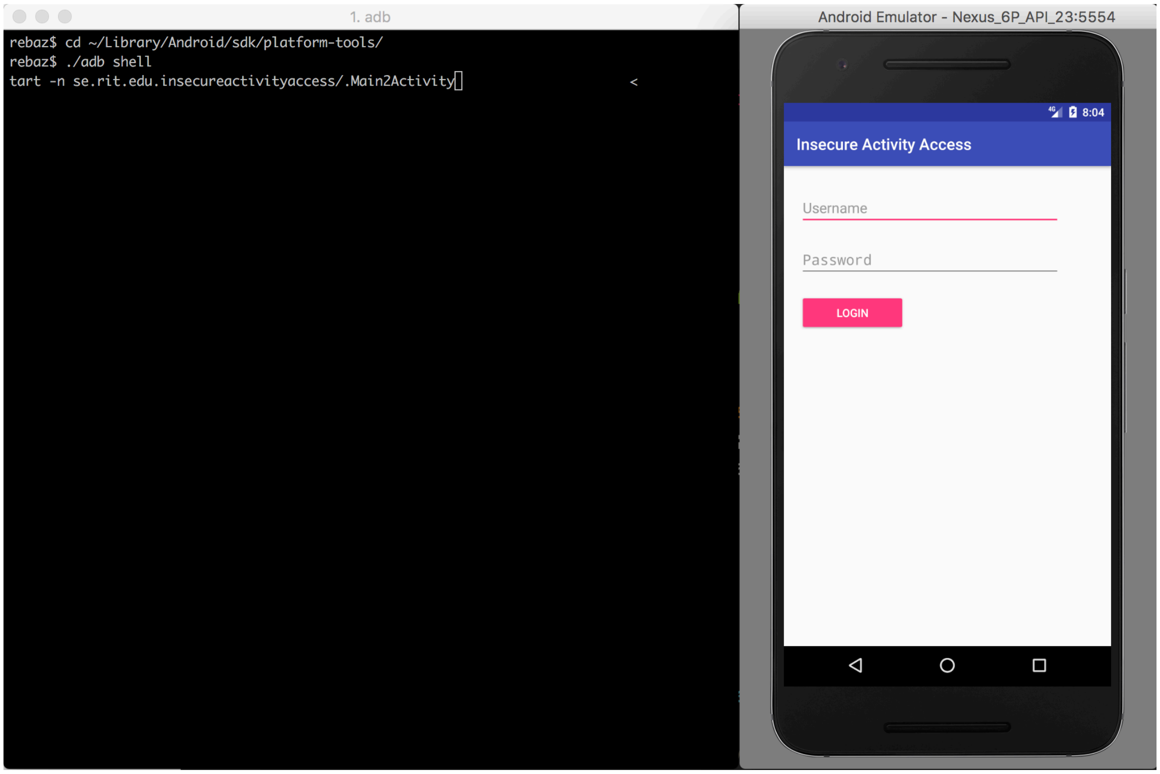
1. Code
   The **onCreate** method of the MainActivity class is very like the one we see in the previous part. The only thing that has changed is the highlighted line: we include a key-value pair with the intent we are passing to the startActivity function.

```
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // username and password hardcoded for testing purpose
        final String USERNAME = "admin";
        final String PASSWORD = "admin";


        // initialize  user name  instance with the real input in xml
        final EditText etUsername = (EditText)findViewById(R.id.etUsername);
        // initialize  password  instance with the real input in xml
        final EditText etPassword = (EditText)findViewById(R.id.etPassword);

        // initialize login button instance
        final Button btnLogin = (Button)findViewById(R.id.btnLogin);
        btnLogin.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {

                // collect user's username input
                String username = etUsername.getText().toString();

                // collect user's password input
                String password = etPassword.getText().toString();

                // compare values
                if(USERNAME.equals(username) && PASSWORD.equals(password))
                {
                    Toast.makeText( MainActivity.this,
                            "You are logged in successfully",
                            Toast.LENGTH_LONG).show();

                    Intent intent = new Intent(getApplicationContext(),
Main2Activity.class);
                    intent.putExtra("key", 3433);
                    startActivity(intent);
                }

                else {
                    Toast.makeText( MainActivity.this,
                            "Invalid credentials",
                            Toast.LENGTH_LONG).show();
                }
            }
        });
```

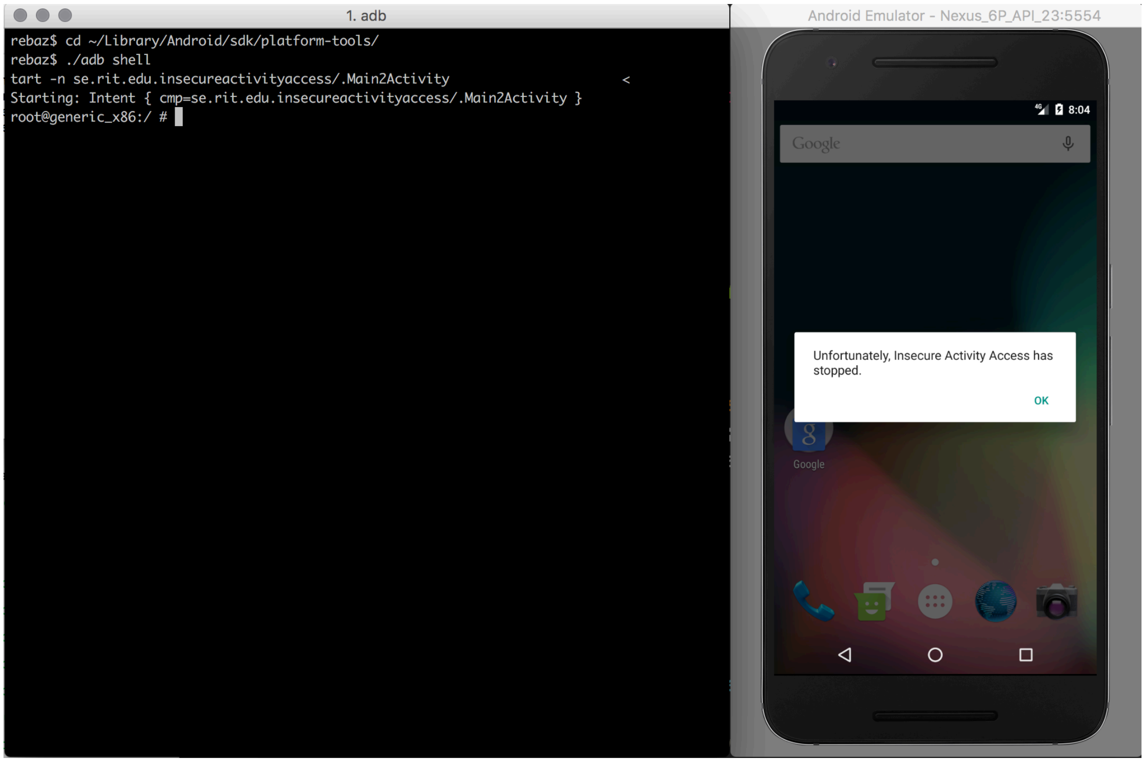
Now, we change the **onCreate** method of the Main2Activity class to check the intent it is passed for the key-value pair. If it is incorrect, or non-existent, then the Main2Activity class will simply not start.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);

    Bundle b = getIntent().getExtras();

    int key = b.getInt("key");
    if (key != 3433)
        finish();
}
```

2. When we run the same commands in adb again, we will get the following screens:



```
rebaz$ cd ~/Library/Android/sdk/platform-tools/
rebaz$ ./adb shell
tart -n se.rit.edu.insecureactivityaccess/.Main2Activity                     <
```



```
rebaz$ cd ~/Library/Android/sdk/platform-tools/
rebaz$ ./adb shell
tart -n se.rit.edu.insecureactivityaccess/.Main2Activity                     <
Starting: Intent { cmp=se.rit.edu.insecureactivityaccess/.Main2Activity }
root@generic_x86:/ #
```

Unfortunately, Insecure Activity Access has stopped.

As we can see, an attacker would no longer be able to access the “change password” screen without the key-value pair.