Title Goes Here

## Background

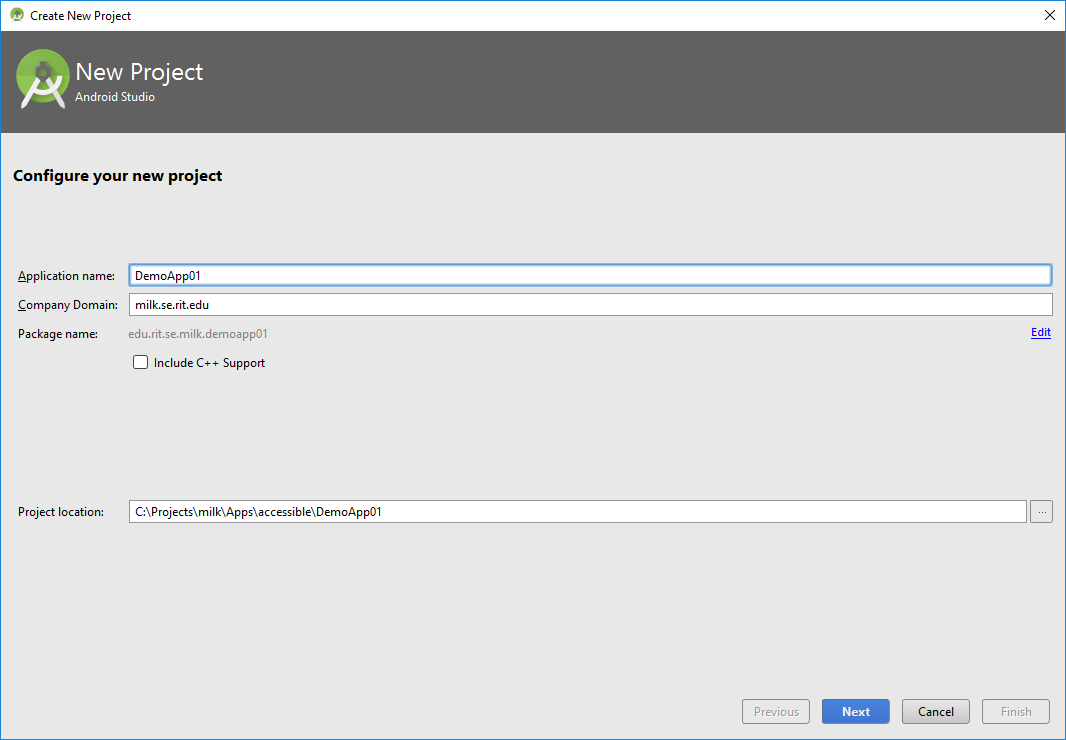
Background goes here. Paragraph about the Android components behind the security problem.

Paragraph about the security problem, and its implications.

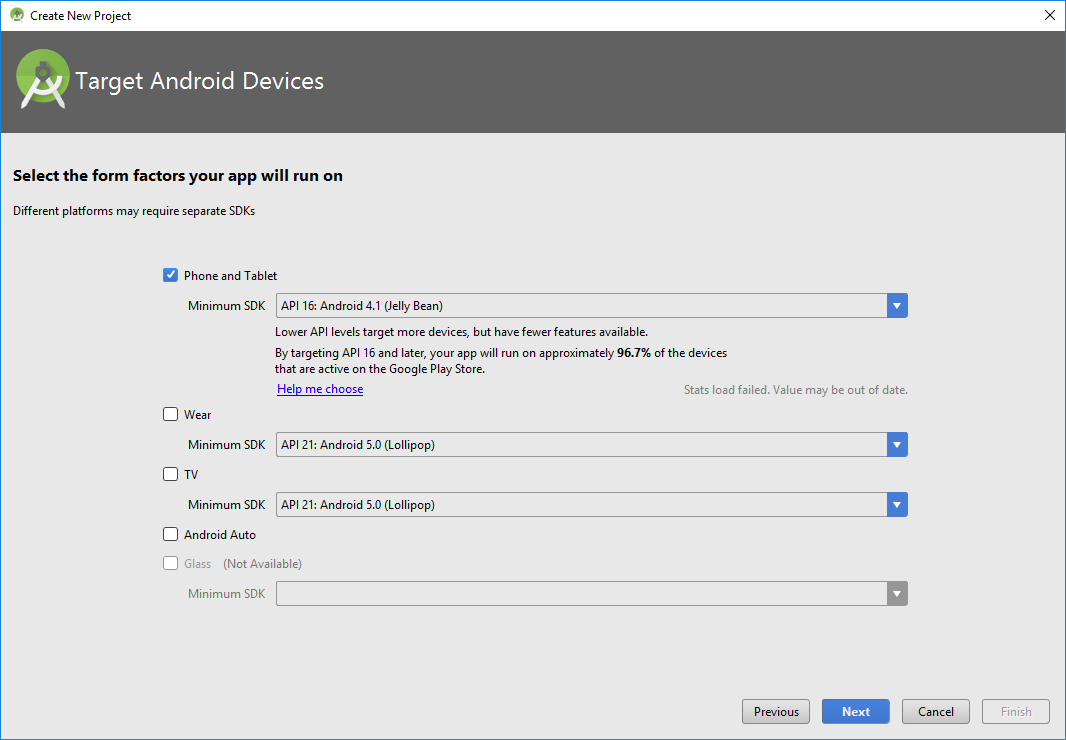
## Activity Instructions

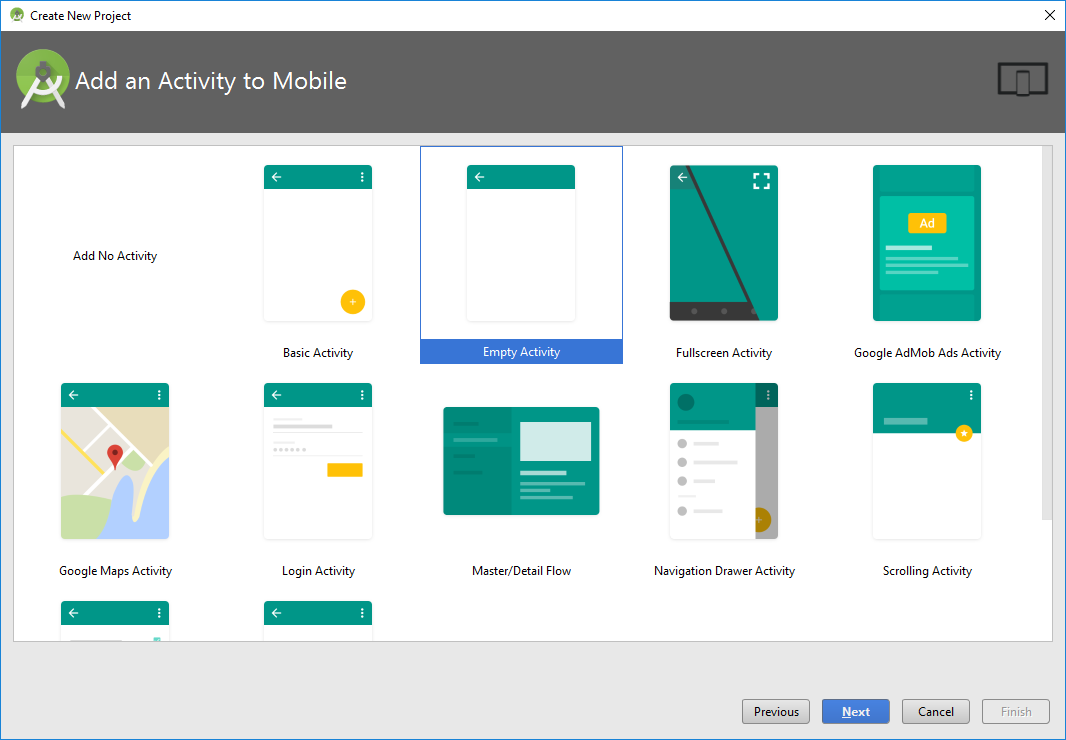
Activity illustrates the problem by doing these things.

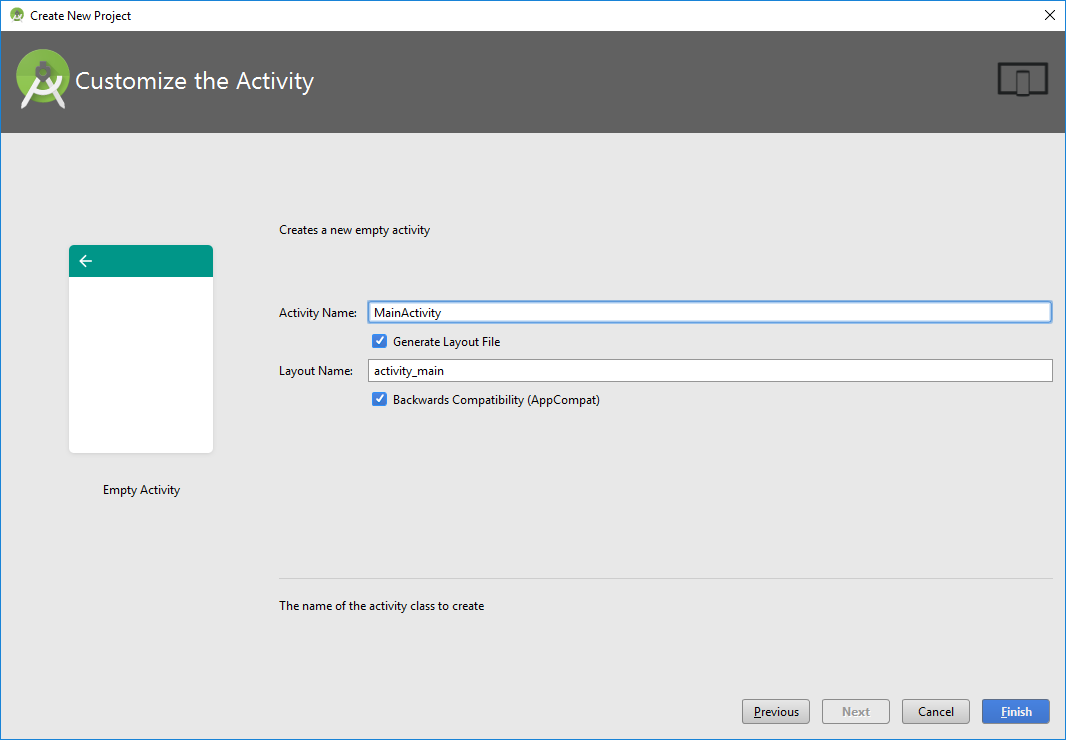
1. Create problematic app.
2. Exploit it.
3. Explain how to avoid it or modify your code to avoid.
4. Project Creation
   1. Follow the screens below to create a new project:



Name the project “TitleGoesHere”.







1. Construct User Interface
   1. From the Palette tool window, add the following UI controls into the screen layout. Access the screen layout by going to “app/res/layout”.
      1. Update the following properties of the existing Relative Layout:
         * layout\_width="match\_parent"
         * layout\_height="match\_parent"
      2. Within the existing Relative Layout add:
         1. **Switch**
            * Update the following properties:

text="Alternate Rendering"

layout\_width="match\_parent"

layout\_height="wrap\_content"

id="@+id/switchAccessibility"

focusable="false"

* + - 1. **Relative Layout**:
         * Update the following properties:

layout\_width="match\_parent"

layout\_height="wrap\_content"

id="@+id/layoutContents"

layout\_weight="100"

* + - * + Add the following UI controls:

**Button**:

Update the following properties:

text="&#60;&#60;--"

layout\_width="wrap\_content"

layout\_height="wrap\_content"

layout\_alignParentTop="true"

layout\_alignParentLeft="true"

layout\_alignParentStart="true"

layout\_marginTop="146dp"

id="@+id/buttonLeft"

background="@android:drawable/btn\_default"

gravity="center"

layout\_gravity="left|center"

layout\_marginLeft="50dp"

**TextView**

* Update the following properties:

text="Tap the Back or Next button to proceed"

layout\_width="match\_parent"

layout\_height="wrap\_content"

id="@+id/textView"

textAppearance="@android:style/TextAppearance.DeviceDefault.Medium"

layout\_above="@+id/buttonRight"

layout\_marginBottom="36dp"

textStyle="normal|bold"

textAlignment="center"

layout\_alignParentLeft="false"

layout\_alignParentStart="false"

layout\_alignParentRight="false"

layout\_alignParentEnd="false"

gravity="center\_horizontal"

**Button**

Update the following properties:

text="--&#62;&#62;"

layout\_width="wrap\_content"

layout\_height="wrap\_content"

id="@+id/buttonRight"

layout\_gravity="right"

background="@android:drawable/btn\_default"

gravity="center"

layout\_alignTop="@+id/buttonLeft"

layout\_alignParentRight="true"

layout\_alignParentEnd="true"

layout\_marginRight="50dp"

* + - 1. **Relative** **Layout**
         * Update the following properties:

layout\_width="match\_parent"

layout\_height="wrap\_content"

layout\_weight="1"

* + - * + Add the following UI controls:

**TextView**:

Update the following properties:

layout\_width="match\_parent"

layout\_height="wrap\_content"

id="@+id/textViewStatus"

layout\_alignParentTop="true"

layout\_alignParentLeft="true"

layout\_alignParentStart="true"

textAlignment="center"

textStyle="normal|bold"

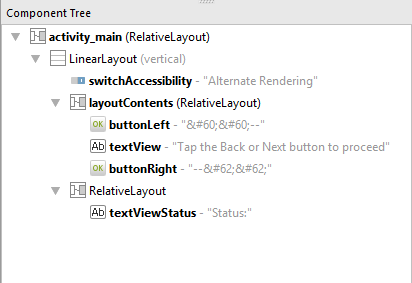
layout\_alignParentRight="true"

layout\_alignParentEnd="true"

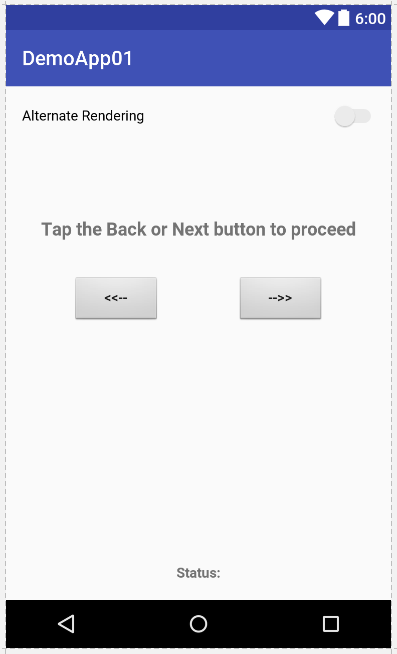
text="Status:"

gravity="bottom"

Following is the hierarchical layout of the controls on the screen:



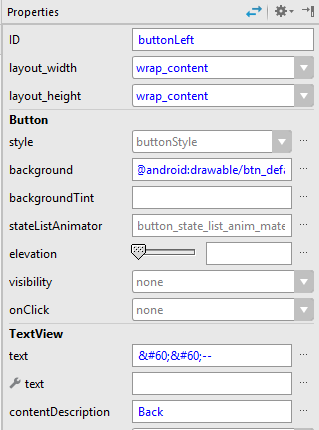
Following is the rendering of controls on the screen:



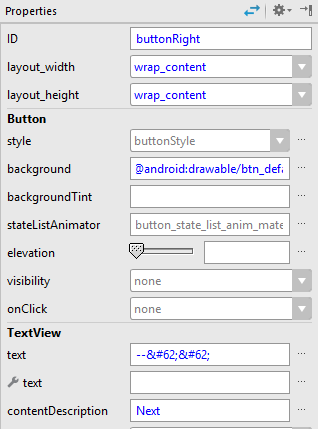
1. Set Content Description

The “contentDescription” property of a control is utilized for accessibility purposes. The text associated with this property defines the content of this control and is used by Android’s TalkBack feature.

Select the Button with id “buttonLeft”. Set contentDescription to “Back”



Select the Button with id “buttonRight”. Set contentDescription to “Next”



1. Code

Open MainActivity.java, found under “app/java/your\_package\_name” and add the following code:

* 1. Declare the following variables:

|  |
| --- |
| **public class** MainActivity **extends** AppCompatActivity {   **public static final** String ***MY\_PREFERENCES*** = **"MyPrefLogin"** ;  **public static final** String ***USERNAME*** = **"UsernameKey"** ;  **public static final** String ***USER\_PASSWORD*** = **"PasswordKey"** ;  SharedPreferences **mSharedPreferences**;  EditText **mUsernameEditText**, **mPasswordEditText**;  Button **mStoreButton**; |

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



* 1. Add the following imports to the file.

|  |
| --- |
| **import** android.content.Context; **import** android.content.SharedPreferences; **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; |

The above code achieves the following:

1. Handles the onCheckedChange event of the Switch control to:
   1. Set the color of all the controls to Black so that none of the controls are visible when the switch is checked (i.e. set to “On”)
   2. Sets the colors of all the controls to their original colors when the switch is unchecked (i.e. set to “Off”)
   3. ‘

## Exploitation Instructions

Activity illustrates the problem by doing these things.

1. Run the app. Enter your login credentials.
2. Using adb shell, view the saved preferences file.
   1. Open Terminal or Command line.
   2. Run the following commands.
   3. ./adb shell

## Defense

Describe how defending against it would work, in theory.

If relevant, include instructions on how to edit the code to show how the defense would work in action.