# Module 1 – Understanding `getIntent()`

You're going to learn to be best friends with `getIntent()`

Before we begin, there is one concept that you must always remember

# The application has to be in the foreground before it can execute "`startActivity()`"

Keep this memorized throughout the remainder of this course

# Understanding `getIntent()`

- When a component is called in Android, an “Intent” is created
- To pass additional data to the component, an Intent can contain extra data
  - This is similar to POST data in a web request
- The Java function `` `getIntent()` `` can be used by the called component to read the incoming Intent data
- Understanding how this Intent data is processed can be the key to exploiting a poorly programmed application

```
private void startActivityIntent(){
    Intent intent = new Intent(); // create intent
    intent.setComponent(new ComponentName( pkg: "target.package",  cls: "targetActivity")); // target component/activity
    intent.putExtra( name: "string_extra",  value: "yay_string_value_yay"); // add string extra
    intent.putExtra( name: "int_extra",  value: 1234); // add int extra
    intent.putExtra( name: "boolean_extra",  value: true); // add boolean extra
    startActivity(intent); // start target activity
}
```

Example function creating an Intent

MALICIOUS ERECTION LLC

# Understanding `getIntent()`

- When `getIntent()` reads the incoming Intent, the data can be stored into a Java object `android.content.intent`
- Most of the time, when `getIntent()` is executed, the application intends to process the extra data stored in the Intent
  - This is again similar to how a web application would read incoming data from a POST request

```
private void readIntent(){
    Intent intent = getIntent(); // retrieve incoming intent
    String yaystringyay = intent.getStringExtra( name: "string_extra"); // retrieve string extra
    int yayintyay = intent.getIntExtra( name: "int_extra",  defaultValue: 0); // retrieve int extra
    boolean yayboolyay = intent.getBooleanExtra( name: "boolean_extra",  defaultValue: false); // retrieve boolean extra
    processData(yaystringyay, yayintyay, yayboolyay); // send retrieved data to another method
}
```

Example function which uses `getIntent()` to read the Intent data

# Understanding `getIntent()`

- Intents can come with extra data and be bundled with other data
- There are many different “extra” types that can be stored in an Intent
  - String (`java.lang.String`)
  - Integer (`int`)
  - Boolean (`java.lang.Boolean`)
  - Byte (`byte`)
  - Parcelable (`android.os.Parcelable`)
  - And many more…
- Java objects can also be bundled in an Intent
  - Uri (`android.net.Uri`)
  - Action (`java.lang.String`)
  - Scheme (`java.lang.String`)
  - NFC data (more on this later in the course)


Android Studio showing different Intent Extras that can be added to an Intent

# Understanding `getIntent()`

- Manually creating an Intent with “extra” data is easy
  - Can be done in Java and/or the ADB shell

```
Intent intent = new Intent();
intent.setComponent(new ComponentName( pkg: "yayPackageYay",  cls: "yayActivityYay"));
intent.putExtra( name: "extraString",  value: "yaystringyay");
intent.putExtra( name: "extraInt",  value: 1234);
startActivity(intent);
```

Java code creating a new Intent with extras and launching the target activity

```
$ am start -n yayPackageYay/yayActivityYay --es extraString yaystringyay --ei extraInt 1234
```

`adb` command creating a new Intent with extras and launching the target activity

MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

- We will now use Axolotl to better demonstrate how `` `getIntent()` `` works
- On Axolotl's main menu, tap:
  - "Exercise Modules"
  - "Understanding getIntent"
- A blank activity will appear with some text
  - The launched activity is programmed via the Java class `com.maliciouserection.axolotl.example.activity.intents.getIntent`


MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

- Use your favorite Android application decompiler to decompile Axolotl
  - For this course, JADX-GUI will be used for demonstration purposes
- Open the `manifest.xml` file and confirm that the Activity `com.maliciouserection.axolotl.example.activity.intents.getIntent` is exported
  - It needs to be exported since we will be launching it from our Example Exploit application


```
<activity android:name="com.maliciouserection.axolotl.example.webViewFileAccessOptions" android:exported="true"/>
<activity android:name="com.maliciouserection.axolotl.example.activity.intents.getIntent" android:exported="true"/>
<activity android:name="com.maliciouserection.axolotl.example.activity.intents.browsableIntent" android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.VIEW"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <category android:name="android.intent.category.BROWSABLE"/>
```

Axolotl's manifest.xml with the exported Activity highlighted

MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

- In JADX, navigate to `com.maliciouserection.axolotl.example.activity.intents.getIntent`
- The function `` `getIntent()` `` is found on various lines of the decompiled Activity, including in the method `` `theMainMethod()` ``
  - The method `` `theMainMethod()` `` checks if the calling Intent's extra String value "yay" is not null
  - If “yay” is not null, a TextView will be filled with whatever value was defined by the Extra String value "yay"
  - Also, if the Extra Integer value "yay" is greater than 0, then the method `` `aSecondMethod(Intent)` `` is executed

```
public class getIntent extends Activity {
    TextView text;
    TextView title;

    @Override // android.app.Activity
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_example_activity);
        TextView textView = (TextView) findViewById(R.id.yayexampleactivityyay_title);
        this.title = textView;
        textView.setText("Example - Understanding getIntent()");
        TextView textView2 = (TextView) findViewById(R.id.yayexampleactivityyay_text);
        this.text = textView2;
        textView2.setText("Analyze this activity and learn about getIntent()");
        theMainMethod();
    }

    private void theMainMethod() {
        if (getIntent().getStringExtra("yay") != null) {
            String yaystringyay = getIntent().getStringExtra("yay");
            this.text.setText("theMainMethod - getIntent().getStringExtra(\"yay\"): " + yaystringyay);
        }
        int yayintyay = getIntent().getIntExtra("yay", 0);
        if (yayintyay > 0) {
            aSecondMethod(getIntent());
        }
    }

    private void aSecondMethod(Intent yayintentyay) {
        Toast.makeText(getApplicationContext(), "getIntent(): " + yayintentyay, 1).show();
    }
```

Decompiled example Activity `getIntent`

MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

- When opening the example Activity, we can use Frida to analyze the Intent that gets retrieved by `` `theMainMethod()` ``
- Make a Frida script which hooks into the Activity `` `getIntent` ``, method `` `theMainMethod()` ``
- The script should output “called” in the Frida console whenever the method `` `theMainMethod()` `` is executed
- After making the script, use Frida to hook into Axolotl and run the script
  - `frida -U -l ./frida-script.js Axolotl`

```
Java.perform(function() {
    var yayclass1yay = Java.use('com.maliciouserection.
    axolotl.example.activity.intents.getIntent');
    yayclass1yay.theMainMethod.overload().
        implementation = function() {
        console.log("called");
        var ret_value = this.theMainMethod();
        return ret_value;
    }
});
```

Frida script that hooks into the example Activity

MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

- After using Frida to hook into Axolotl, launch the Activity `` `getIntent` ``
  - You can launch `` `getIntent` `` via `` `adb` `` or Example Exploit via Java code
  - Example commands/code are below

- Remember, an application must be in the foreground before it can execute `` `startActivity()` ``

```
Intent intent = new Intent();
intent.setComponent(new ComponentName(
        pkg: "com.maliciouserection.axolotl",
        cls: "com.maliciouserection.axolotl.example.activity.intents.getIntent"));
startActivity(intent);
```

Java code creating a new Intent and launching the example Activity

```
emulator64_x86_64_arm64:/ $ am start -n \
> com.maliciouserection.axolotl/.example.activity.intents.getIntent
```

`` `adb` `` command creating a new Intent and launching the example Activity

MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

```
emulator64_x86_64_arm64:/ $ am start -n \
> com.maliciouserection.axolotl/.example.activity.intents.getIntent
Starting: Intent { cmp=com.maliciouserection.axolotl/.example.activity.intents.getIntent }
```

`adb` command launching the example Activity

- After launching `getIntent`, your Frida console should show the output “called”

```
c:\>frida -U -l .\frida-script.js Axolotl
     ____
    / _  |   Frida 16.1.7 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to Android Emulator 5554 (id=emulator-5554)

[Android Emulator 5554::Axolotl ]-> called
```

Frida output

MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

- Lets modify the Frida script so that when the example Activity is opened, the Java code `` `getIntent()` `` is executed and logged to the Frida console
  - Yes you can execute Java functions from Frida!
  - Don't forget to convert the output of `` `getIntent()` `` into a String object via `` `toString()` ``
  - The console log only supports String objects
  - `` `getIntent()` `` returns an Intent object, not a String object
- Adjust your Frida script and re-launch the example Activity

```
Java.perform(function() {
    var yayclass1yay = Java.use('com.maliciouserection.
    axolotl.example.activity.intents.getIntent');
    yayclass1yay.theMainMethod.overload().
        implementation = function() {
        console.log("called");
        console.log(this.getIntent().toString());
        var ret_value = this.theMainMethod();
        return ret_value;
    }
});
```

Frida script that hooks into the example Activity

MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

- After launching `getIntent`, your Frida console should show the output “called” as well as the contents of your Intent

```
emulator64_x86_64_arm64:/ $ am start -n \
> com.maliciouserection.axolotl/.example.activity.intents.getIntent
Starting: Intent { cmp=com.maliciouserection.axolotl/.example.activity.intents.getIntent }
```

`adb` command launching the example Activity

```
c:\>frida -U -l .\frida-script.js Axolotl
     ____
    / _  |   Frida 16.1.7 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to Android Emulator 5554 (id=emulator-5554)

[Android Emulator 5554::Axolotl ]-> called
Intent { flg=0x10000000 cmp=com.maliciouserection.axolotl/.example.activity.intents.getIntent }
```

Frida output

MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

- As a final step, lets specifically log the Intent string extra "yay"
- Whichever method you are using to launch the Activity `getIntent`, you will need to add the String Extra “yay” to the launching Intent
- Below are code examples that have the String extra “yay” added to the launching Intent

```
Intent intent = new Intent();
intent.setComponent(new ComponentName(
        pkg: "com.maliciouserection.axolotl",
        cls: "com.maliciouserection.axolotl.example.activity.intents.getIntent"));
intent.putExtra( name: "yay",  value: "test input");
startActivity(intent);
```

Java code creating a new Intent with a String extra and launching the example Activity

```
emulator64_x86_64_arm64:/ $ am start -n \
> com.maliciouserection.axolotl/.example.activity.intents.getIntent \
> --es "yay" "test input"
```

`adb` command creating a new Intent and launching the example Activity

MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

- Then make one more adjustment to the Frida script
- We will execute the Java code `` `getIntent()` `` followed by `` `getStringExtra()` `` and output the extra to the console log
- Since `` `getStringExtra()` `` returns a String object, we don't need to add `` `toString()` `` before passing the output to console log
  - Reference: https://developer.android.com/reference/android/content/Intent#getStringExtra(java.lang.String)

```
Java.perform(function() {
    var yayclass1yay = Java.use('com.maliciouserection.
    axolotl.example.activity.intents.getIntent');
    yayclass1yay.theMainMethod.overload().
        implementation = function() {
        console.log("called");
        console.log(this.getIntent().toString());
        console.log(this.getIntent().getStringExtra("
            yay"));
        var ret_value = this.theMainMethod();
        return ret_value;
    }
});
```

Frida script that hooks into the example Activity

MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

```
emulator64_x86_64_arm64:/ $ am start -n \
> com.maliciouserection.axolotl/.example.activity.intents.getIntent \
> --es yay "test input"
Starting: Intent { cmp=com.maliciouserection.axolotl/.example.activity.intents.getIntent (has extras) }
```

`adb` command launching the example Activity

- After launching `getIntent`, your Frida console should show:
  - The output “called”
  - The contents of your Intent
  - The Extra String “yay” value

```
c:\>frida -U -l .\frida-script.js Axolotl
     ____
    / _  |   Frida 16.1.7 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to Android Emulator 5554 (id=emulator-5554)

[Android Emulator 5554::Axolotl ]-> called
Intent { flg=0x10000000 cmp=com.maliciouserection.axolotl/.example.activity.intents.getIntent (has extras) }
test input
```

MALICIOUS ERECTION LLC

# Understanding `getIntent()` - Example

- If you look at the application, you should notice that the text changed
- It will now show the text “test input“
- If you look at the source code for the Activity `getIntent`, you will see that the second `TextView` box will change based on the received Intent’s Extra String “yay” value

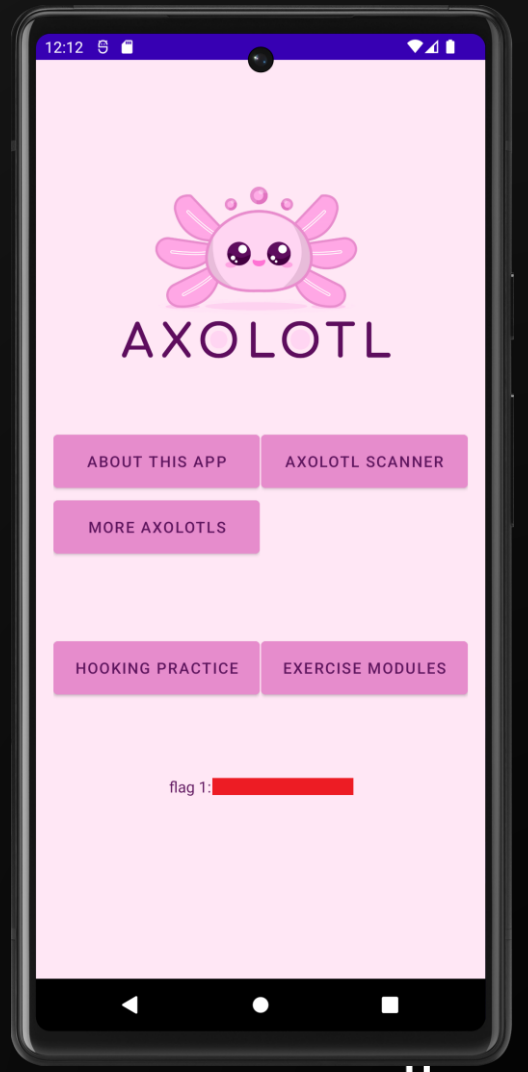
```
private void theMainMethod() {
    if (getIntent().getStringExtra("yay") != null) {
        String yaystringyay = getIntent().getStringExtra("yay");
        this.text.setText("theMainMethod - getIntent().getStringExtra(\"yay\"): " + yaystringyay);
    }
    int yayintyay = getIntent().getIntExtra("yay", 0);
    if (yayintyay > 0) {
        aSecondMethod(getIntent());
    }
}
```

Decompiled example Activity


MALICIOUS ERECTION LLC

# Module 1 Exercise

- Earlier, it was mentioned that the activity `com.maliciouserection.axolotl.example.activity.intents.getIntent` had the method `aSecondMethod(Intent)`
  - Make an Intent which calls this method
- Capture The Flag - The activity `com.maliciouserection.axolotl.MainActivity` contains the method `showFlag1()`
  - This method will output Flag 1 on Axolotl's `MainActivity` if the Activity receives an Intent with specific Intent Extra values
  - Craft an Intent which will show the flag on Axolotl's `MainActivity`
    - Hint: https://developer.android.com/reference/android/content/Intent is your friend
    - Another hint: Keep in mind how you used Frida to execute the Java function `getIntent()`




MALICIOUS ERECTION LLC