APK TESTING REPORT

Mobile Ecosystem Security

APK INFORMATION

APK Name :- Assignment.apk

Package Name :- com.intrepidusgroup.learner

Description :- Assignment.apk has eight challenges which are designed as:-

TESTING PLATFORM

Host Machine :- Microsoft Windows

Virtual Machine :- Kali Linux

Assignment.apk has been installed in Android Virtual Environment(GenyMotion)

TOOLS

Adb :- Android Debug Bridge is a tool used to communicate with the phone

Apk Tool :- A tool for reverse engineering 3rd party, closed, binary Android apps. It can decode resources to nearly original form and rebuild them after making some modifications.

MobSF:- Mobile Security Framework (MobSF) is an automated, all-in-one mobile application (Android/iOS/Windows) pen-testing, malware analysis and security assessment framework capable of performing static and dynamic analysis.

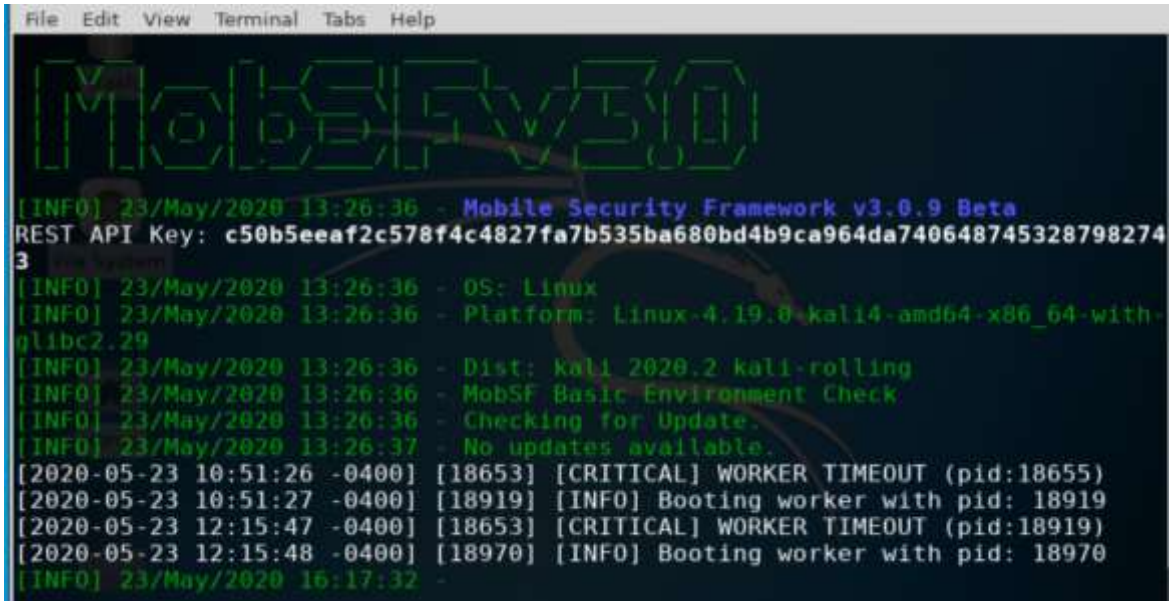
Drozer :- Drozer (formerly Mercury) is the leading security testing framework for Android.

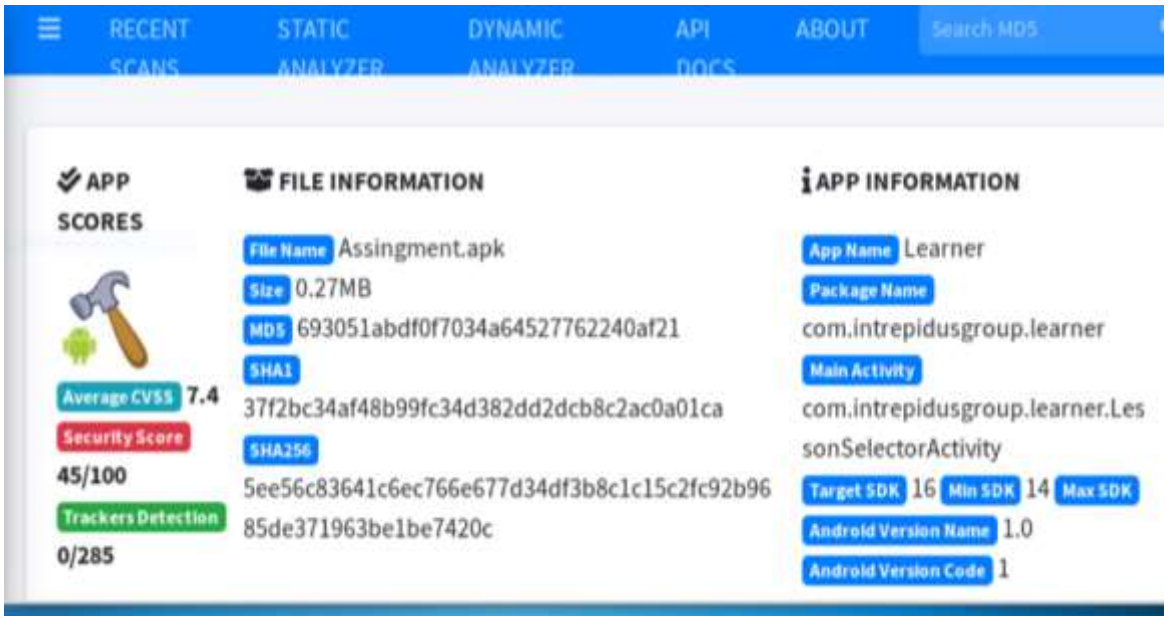
Burpe Suite :- Intercept tool by Portswigger

Assignment.apk Analysis Using MobSF

Mobile Security Framework (MobSF) is an automated, all-in-one mobile application (Android/iOS/Windows) pen-testing, malware analysis and security assessment framework capable of performing static and dynamic analysis.

MobSF is used to analyse the Java Code of the Application.





Connect to Phone

To connect to phone and install the apk we will use the following commands as shown below :-

```
root@kali:~/Downloads/platform-tools# adb connect 192.168.42.102
connected to 192.168.42.102:5555
root@kali:~/Downloads/platform-tools# adb devices
List of devices attached
192.168.42.102:5555     device
```

192.168.42.102 = IP of Mobile Phone

```
# adb install Assignment.apk
```

Lesson 1: Android Logging Secret

Analysed the java code of com/intrepidusgroup/learner/Lesson1Activity.java and the instruction given in lession 1 we get to know that create challenge function is followed by fillLogwithGarbage( ) function which is responsible for creating logs appending with “LEARNER”

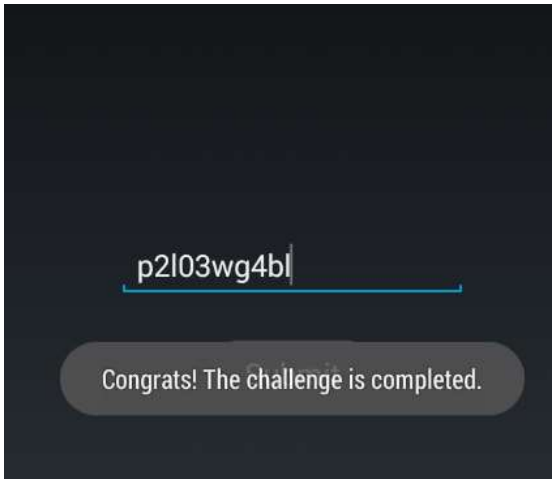
```
    }

    private void fillLogWithGarbage() {
        int nonSecureRandomInt = new Random().nextInt(100);
        for (int i = 0; i < 100; i++) {
            if (i == nonSecureRandomInt) {
                Log.d("LEARNER",
String.valueOf(getResources().getString(R.string.challengeString)) + this.challenge);
            } else {
                Log.d("LEARNER",
String.valueOf(getResources().getString(R.string.garbageLogString)) +
generateChallenge());
            }
        }
    }
```

Analyzing the log with Learner as a tag

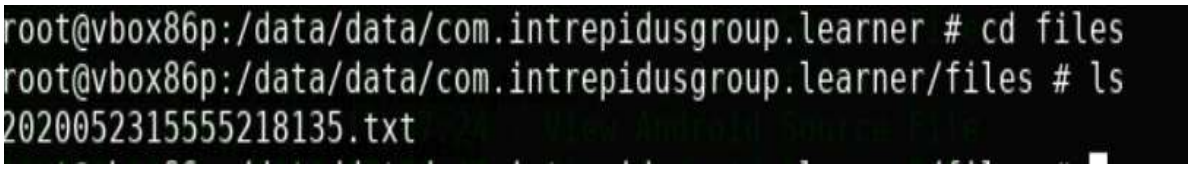
```
root@kali:~/Downloads/platform-tools# adb logcat LEARNER
```


D/LEARNER (2746): This is garbage. The challenge value is:6beid6g74n
D/LEARNER (2746): This is garbage. The challenge value is:4v1snalou1
D/LEARNER (2746): This is garbage. The challenge value is:1bcevb71ww
D/LEARNER (2746): This is garbage. The challenge value is:7mx7n666mc
D/LEARNER (2746): This is not garbage. The challenge value is:p2l03wg4bl
D/LEARNER (2746): This is garbage. The challenge value is:mtujs8q8h3
D/LEARNER (2746): This is garbage. The challenge value is:o4aay1jmbe
D/LEARNER (2746): This is garbage. The challenge value is:csidpr92kg
D/LEARNER (2746): This is garbage. The challenge value is:in4e7jxa5u
D/LEARNER (2746): This is garbage. The challenge value is:77l61n0k45

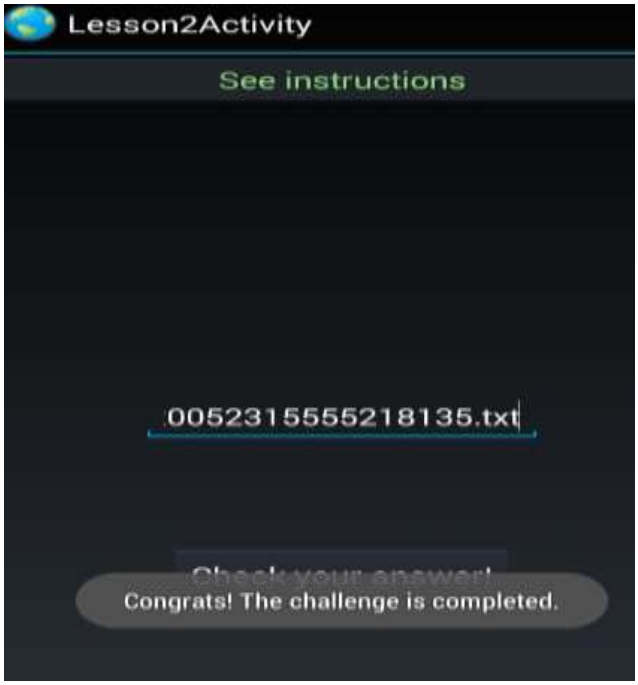

p2l03wg4bl
Congrats! The challenge is completed.

Lesson 2: Screwy File Permission

In java source of file com/intrepidusgroup/learner/Lesson2Activity.java the code for generating the file name is visible according to which the file name is current date with phone number and extension as .txt But, Here the need is to find the file in the apk package following the hint



Location of file

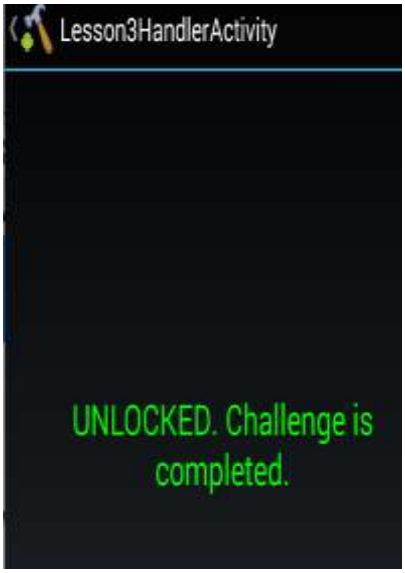


Lesson 3: URI Handler Craziness

Just By analyzing the file Just by com/intrepidusgroup/learner/Lesson3HandlerActivity.java we will get the answer for this challenge

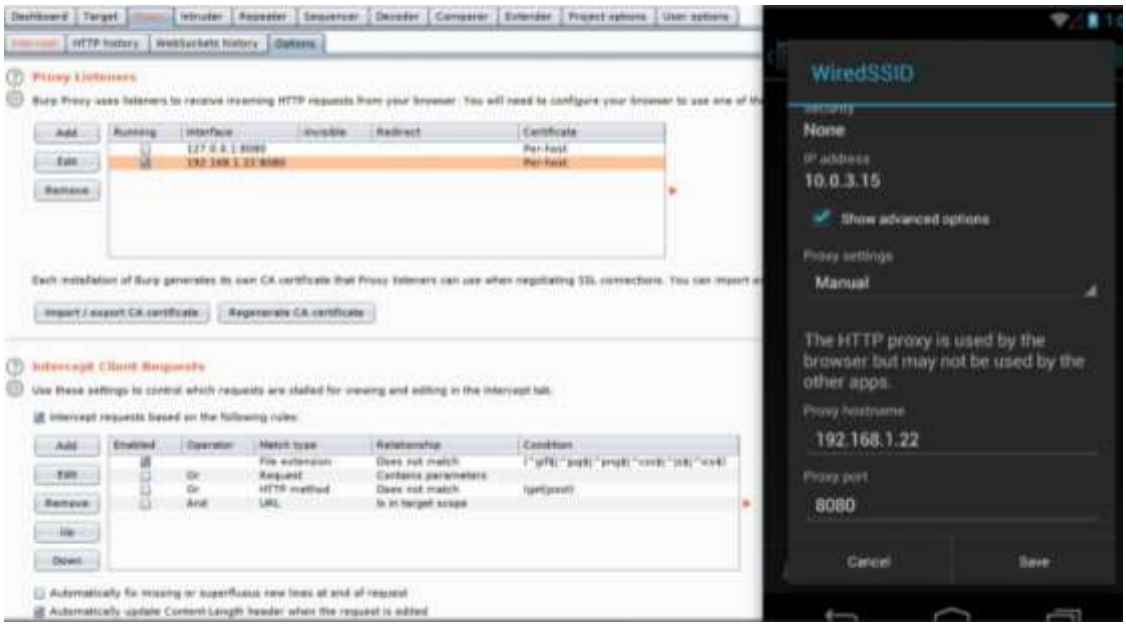
```
public class Lesson3HandlerActivity extends Activity {
    final String unlockkey = "crazyurihandler";

    /* access modifiers changed from: protected */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lesson3_handler);
        getActionBar().setDisplayHomeAsUpEnabled(true);
        Intent uriReqIntent = getIntent();
        String uri = uriReqIntent.getScheme();
        String path = uriReqIntent.getData().getPath();
        Log.d("LEARNER", "Received request for URI " + uri + " at path: " + path);
        TextView lockedStatus = (TextView) findViewById(R.id.textViewLocked);
        if (path.contains("crazyurihandler")) {
```

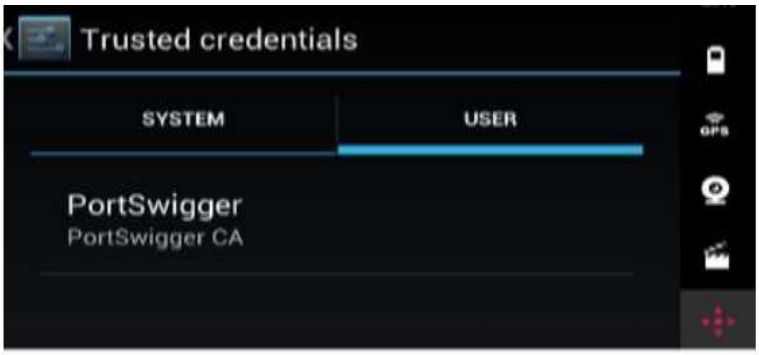


Lesson 4: SSL Man in the Middle Attack

This challenge requires Burp Suite to be configured with the custom phone. Download the burp certificate from http://burp from phone's browser as cert.der



After adding the PortSwigger Certificate as a trusted one in System < Security < Trusted Credentials, Open the Learner application and click on resend code while keeping burp suite intercept as on. The Secret Token Header is the key for this level, i.e. i6ju7n2zx7



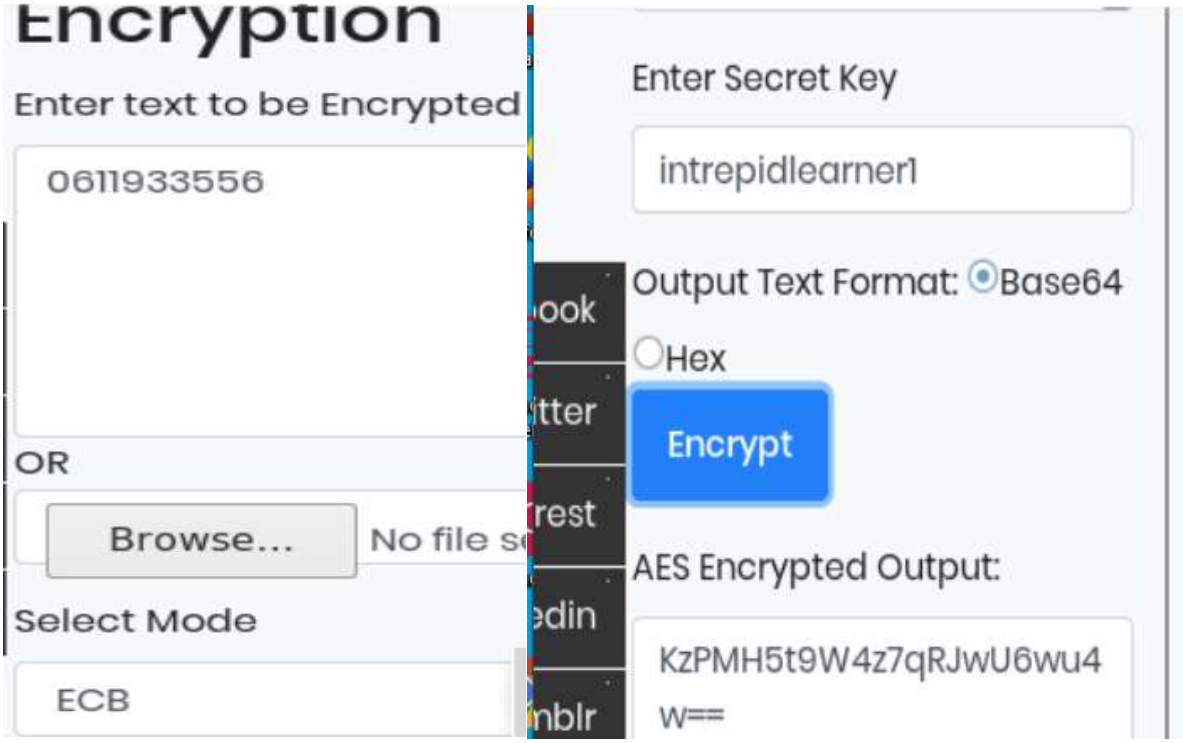
Lesson 6: Encryption vs Encraption

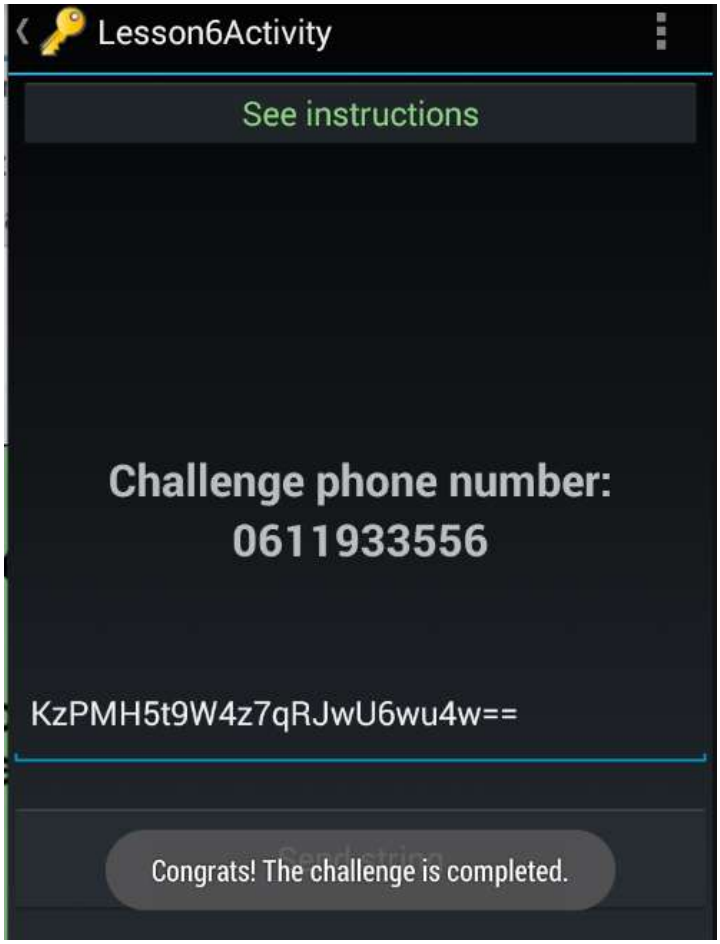
Analysing the com/intrepidusgroup/learner/Lesson6Activity.java code we get to know the AES Encryption is used with key “intrepidlearner1”

```
    private String encryptNumberWithAES(String number) throws
NoSuchAlgorithmException, NoSuchPaddingException, InvalidKeyException,
IllegalBlockSizeException, BadPaddingException {
        byte[] numberToEncryptBytes = number.getBytes();
        byte[] key = "intrepidlearner1".getBytes();
        Cipher c = Cipher.getInstance("AES");
        c.init(1, new SecretKeySpec(key, "AES"));
        return Base64.encodeToString(c.doFinal(numberToEncryptBytes), 0);
    }

    public void lesson6OnSubmitClick(View v) {
        if (checkSecretToken(((TextView)
```

We had used online tool to decrypt the phone number that is provided to us.





Lesson 7: Shared with the World

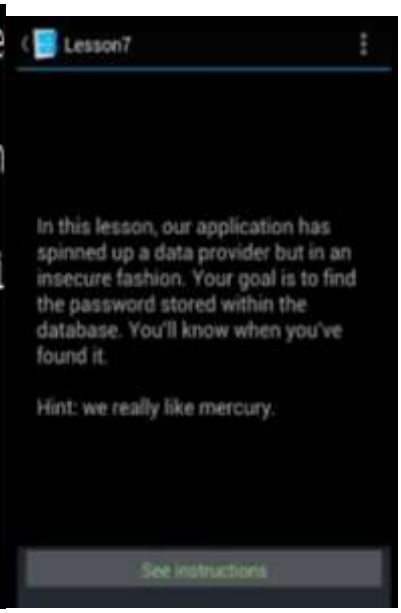
Drozer is an open source software, a framework for Android security assessments. “Drozer allows you to assume the role of an Android app, and to interact with other apps, through Android’s Inter-Process Communication (IPC) mechanism, and the underlying operating system.”

```
public class Lesson7ContentProvider extends ContentProvider {
    private static final String AUTHORITY = "com.intrepidusgroup.learner.contentprovider";
    private static final String BASE_PATH = "iglearnerdb";
    public static final String CONTENT_ITEM_TYPE = "vnd.android.cursor.item/user";
    public static final String CONTENT_TYPE = "vnd.android.cursor.dir/users";
    public static final Uri CONTENT_URI = Uri.parse("content://com.intrepidusgroup.learner.contentprovider/iglearnerdb");
    private static final int USERS = 10;
    private static final int USER_ID = 20;
    private static final UriMatcher sURIMatcher = new UriMatcher(-1);
    private Lesson7DatabaseHelper database;
```

Download the drozer.apk and install in the phone and forward the tcp ports.

```
root@kali:~/Downloads/platform-tools# adb forward tcp:31415 tcp:31415
root@kali:~/Downloads/platform-tools# drozer console connect
```

```
dz> run app.provider.query content://com.intrepidusgroup.learner.ctentprovide
r/iglearnerdb
Could not get a ContentProviderClient for content://com.intrepidusgroup.learn
er.ctentprovider/iglearnerdb.
dz> run app.provider.query content://com.intrepidusgroup.learner.contentprovi
der/iglearnerdb
| _id | user    | password                    | description
|
| 1   | johndoe | you found the password:zod4w3tnox | John Doe's description
|
```



Lesson 8: Malicious intent

Analysing the Manifest.xml file will help in this case .The file includes nodes of each of the files including Activities, Services ,Content Providers and Broadcast Receiver that makes an application and intent filters Permission determines how they coordinate with each of the other Application.

```
    /* access modifiers changed from: protected */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lesson8_aux);
        getActionBar().setDisplayHomeAsUpEnabled(true);
        TextView lockedStatus = (TextView) findViewById(R.id.lesson8TextViewLocked);
        Intent intent = getIntent();
        if
(!intent.getAction().equals(getResources().getString(R.string.lesson8IntentActionStrin
g)) || !intent.getExtras().containsKey(getSecret())) {
            lockedStatus.setText(R.string.wrongIntentString);
            lockedStatus.setTextColor(-65536);
            return;
        }
        lockedStatus.setText(R.string.debugString);
        lockedStatus.setTextColor(-16711936);
    }
```

In Lesson8AuxActivity.java we can see that class Lesson8AuxActivity calls for action with an extra string defined in function getSecret() and the secret is current date.

```
    private String getSecret() {
        return getDate();
    }

    private String getDate() {
        DateFormat dateFormat = new SimpleDateFormat("yyyyMMdd");
        Date currDate = new Date();
        Log.d("LEARNER", "Current date is " + dateFormat.format(currDate));
        return dateFormat.format(currDate);
    }
}
```

Drozer is being used

Following are the steps to call the action “com.intrepidusgroup.learner.custom.intent.action.SEND” :

```
root@kali:~/Downloads/platform-tools# adb forward tcp:31415 tcp:31415
root@kali:~/Downloads/platform-tools# drozer console connect
```

```
For more information on how to formulate an Intent, type 'help intents'.

Last Modified: 2012-11-06
Credit: MWR InfoSecurity (@mwrlabs)
License: BSD (3 clause)

optional arguments:
  -h, --help
  --action ACTION       specify the action to include in the Intent
  --category CATEGORY [CATEGORY ...]
                        specify the category to include in the Intent
  --component PACKAGE COMPONENT
                        specify the component name to include in the Intent
  --data-uri DATA_URI   specify a Uri to attach as data in the Intent
  --extra TYPE KEY VALUE
                        add an field to the Intent's extras bundle
  --flags FLAGS [FLAGS ...]
                        specify one-or-more flags to include in the Intent
  --mimetype MIMETYPE   specify the MIME type to send in the Intent
dz>
```

Command

```
dz> run app.activity.start --component com.intrepidusgroup.learner com.intrep
idusgroup.learner.Lesson8AuxActivity --action com.intrepidusgroup.learner.cus
tom.intent.SEND --extra string 20200524 intent
dz>
```


Lesson8AuxActivity
DEBUG SCREEN!