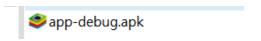
It's my secret

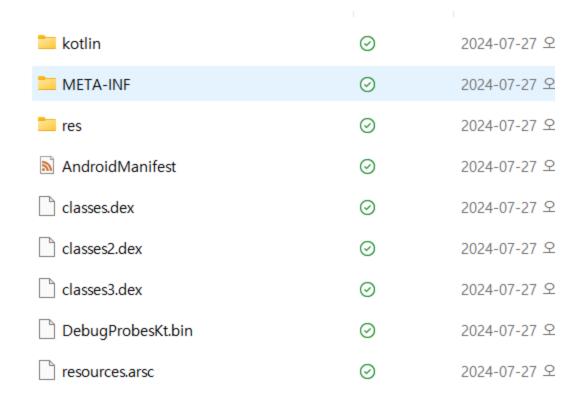


우선 문제 제작에 NOx emulator를 사용했다고 쓰여 있었는데, kusis 강의를 수강하기 위해 환경설정을 하다 내 pc에서 nox가 정상적으로 작동하지 않는다는 것을 발견했다.

가능하다면 블루 스택스로 대체할 예정이다.



apk 파일; 안드로이드용 설치 파일이라는 듯.

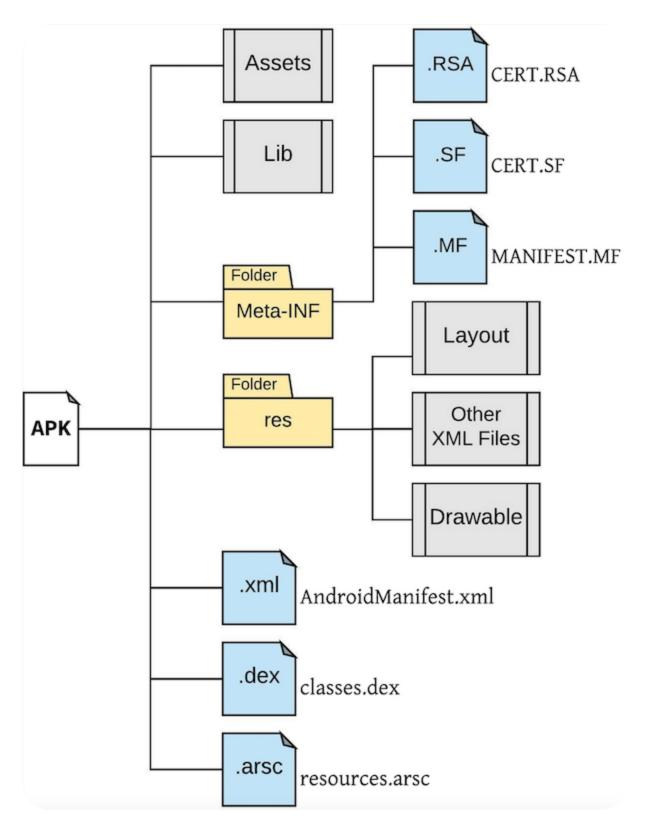


ctf 당일날 다운로드 받은 zip파일엔 해체 버전이 들어 있다.



AndroidMenifest.xml

보통은 xml 파일에 정보값이 들어있는데... 이 모양이니 참고하긴 쉽지 않을 듯.



<u>https://tech-carrot.tistory.com/entry/안드로이드-APK-구조</u>

위의 참고자료의 도움을 받아 manifest.xml을 디코딩 해 보기로 했다.



java -jar AXMLPrinter2.jar AndroidManifest.xml > AndroidManifest2.xml

→ 디코딩.



```
cmanifest
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:versionCode="1"
   android:versionName="1.0"
   android:compileSdkVersion="34"
   android:compileSdkVersionCodename="14"
   package="com.example.securityfactorial"
   platformBuildVersionCode="34"
   platformBuildVersionName="14"
   <uses-sdk
       android:minSdkVersion="24"
       android:targetSdkVersion="34"
   </uses-sdk>
   <permission</pre>
       android:name="com.example.securityfactorial.DYNAMIC RECEI
       android:protectionLevel="0x000000002"
   </permission>
   <uses-permission
       android:name="com.example.securityfactorial.DYNAMIC RECEI
   </uses-permission>
   <application</a>
       android:theme="@7F100272"
       android:label="@7F0F001C"
       android:icon="@7F0D0000"
       android:debuggable="true"
```

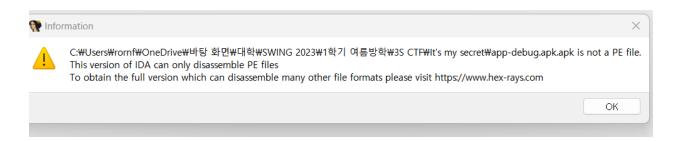
성공적으로 복호화되었다.

전체적으로 훑어봤는데 딱히 플래그와 관계된 내용은 보이지 않음.

분류가 안드로이드/리버싱이었으니 이걸 추적해야 하는 걸까?

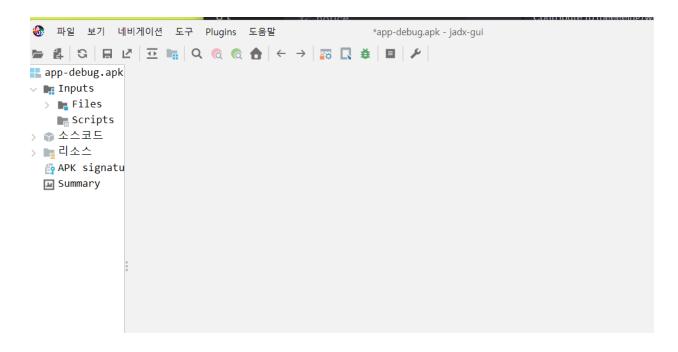
```
<action
    android:name="android.intent.action.MAIN"
    >
```

안드로이도 + 리버싱을 검색해 몇 가지 방법을 찾아봤다.



아이다는 파일 형식 문제로 지원하지 않음.

- 1. APK 추출: APK 파일을 디바이스나 에뮬레이터에서 추출합니다.
- 2. **DEX 파일 분리:** 추출한 APK 파일에서 DEX 파일을 분리합니다.
- 3. JADX 실행: jadx 도구를 사용하여 DEX 파일을 디컴파일합니다.
- 4. Java 소스 코드 획득: 디컴파일된 코드를 확인하고 필요한 Java 소스 코드를 얻습니다.



jadx 툴을 설치했다.

텍스트 검색 :	
Q* main	
정의 검색 :	션 검색:ㅡ
□ 클래스 □ 메소드 □ 필드 ☑ 코드 □ 리소스 □ 주석 ☑	대소문지
노드	
@ androidx.annotation.MainThread	public
androidx.arch.core.executor.TaskExecutor.executeOnl	public
androidx.arch.core.executor.TaskExecutor.executeOnl	if (is <mark></mark>
androidx.arch.core.executor.TaskExecutor.executeOnl	postTo <mark>l</mark>
androidx.arch.core.executor.TaskExecutor.isMainThr	public
androidx.arch.core.executor.TaskExecutor.postToMai	public
androidx.constraintlayout.widget.R.dimen	public
androidx.constraintlayout.widget.R.id	public
androidx.constraintlayout.widget.R.id	public
androidx.core.app.ActivityRecreator	private
androidx.core.app.ActivityRecreator	protect
androidx.core.app.ActivityRecreator.getMainThreadF	private Field n
androidx.core.app.ActivityRecreator.getMainThreadF	main Thr
<pre>androidx.core.app.ActivityRecreator.getMainThreadF: androidx.core.app.ActivityRecreator.getMainThreadF:</pre>	return
androidx.core.app.ActivityRecreator.getMainThreadFi androidx.core.app.ActivityRecreator.queueOnStopIfN	final (
androidx.core.app.ActivityRecreator.recreate(Act	if (tok
androidx.core.app.ActivityRecreator.AnonymousCla	mainHar
androidx.core.app.ActivityRecreator.AnonymousCla	<mark>main</mark> Har
androidx.core.app.ActivityRecreator.AnonymousCla	<mark>main</mark> Har
androidx.core.app.ActivityRecreator.AnonymousCla	<mark>main</mark> Har
androidx.core.app.AppLaunchChecker.onActivityCreat	if (!sp
androidx.core.app.NavUtils.getParentActivityIntent	return
androidx.core.app.NavUtils.getParentActivityIntent	Intent
<pre>androidx.core.app.NavUtils.getParentActivityIntent</pre>	Intent
	Ť
모두 로드 더보기 정지 50+개 찾음	

main을 검색하자 50개가 넘는 결과가 나왔는데, 일단 하나씩 살펴보자.

```
@Override // android.view.View.OnClickListener
public void onClick(View v) {
    Toast.makeText(SecretActivity.this, "SF{It's_your_first_step_toward_Android_hacking}", 0).show()
};
```

Com 폴더의 secretactivity에 있었음.

플래그

SF{It's_your_first_step_toward_Android_hacking}