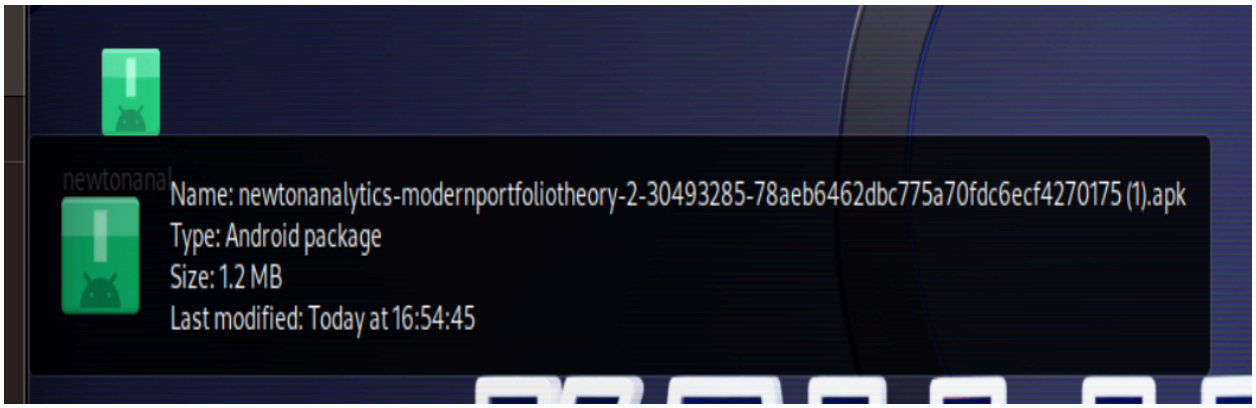
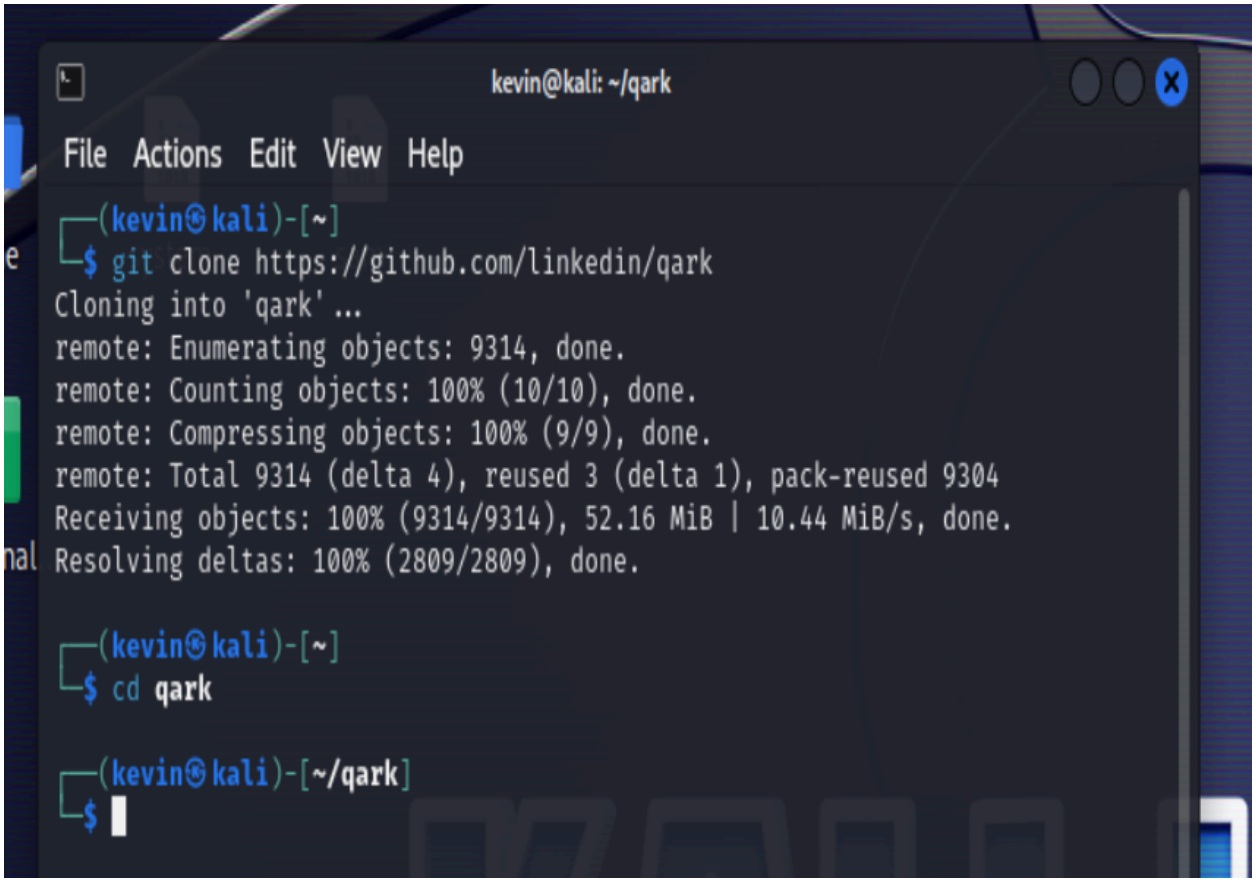
Lab 10 - Mobile Security.pdf

Task 1 - Static Analysis

1) Downloaded the APK for 'Modern Portfolio' from the canvas link instead of apkcombo.com and dragged it to Kali.

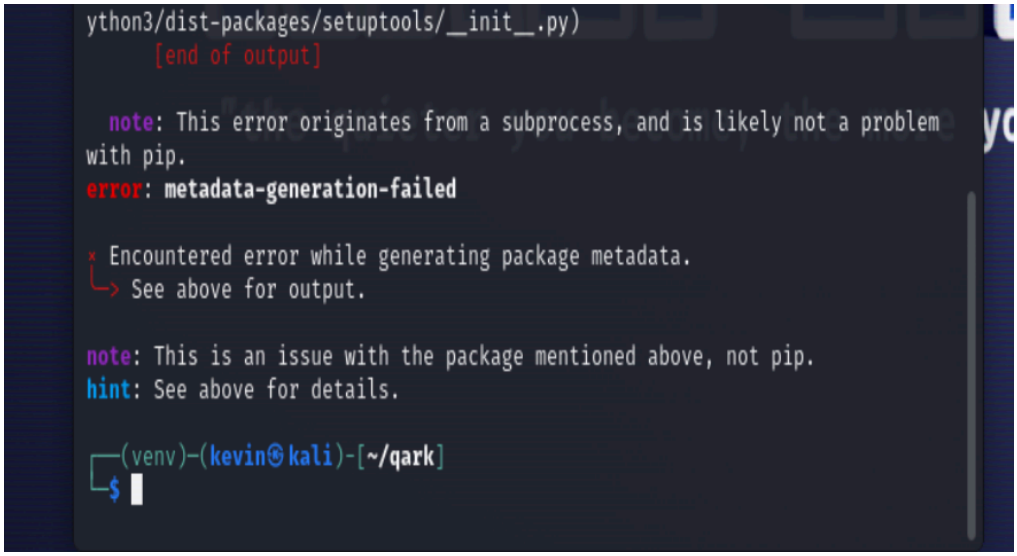


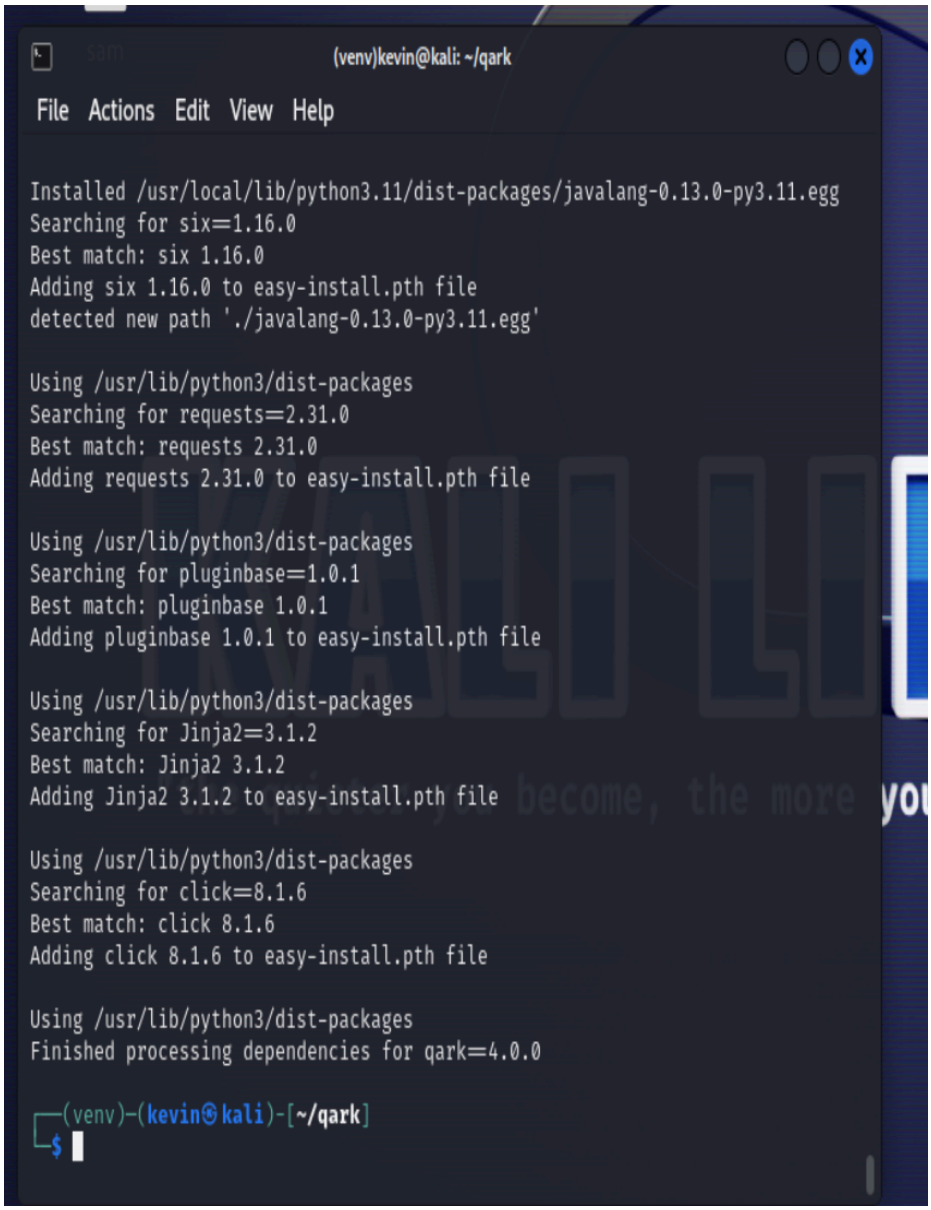
2) Installed and ran Qark in my Kali VM's virtual environment. Encountered some setup errors (requirement.txt installation in the virtual environment), but managed to complete the installation.

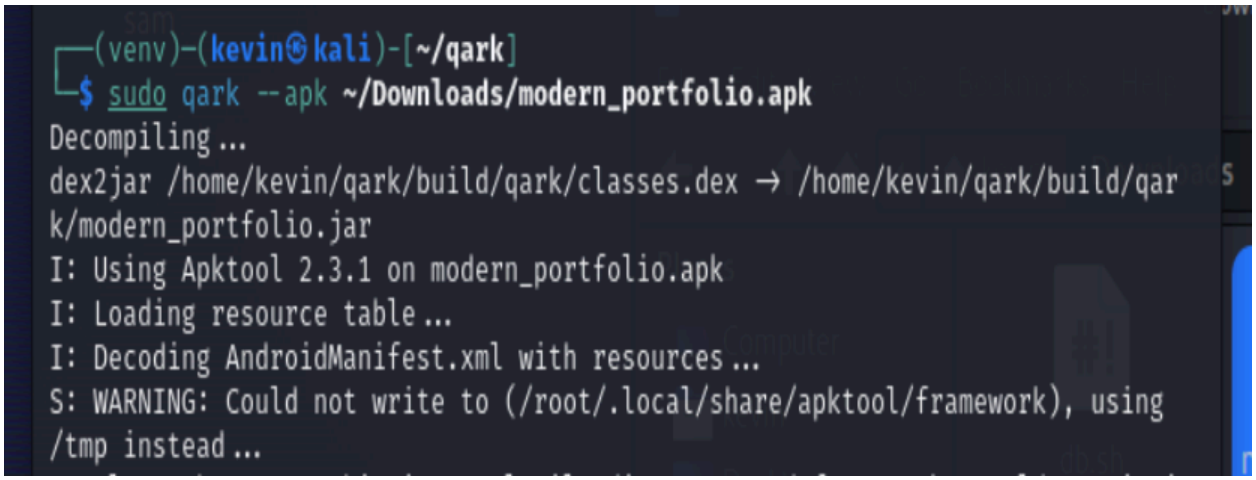

kevin@kali: ~/qark
File Actions Edit View Help
(kevin㉿kali)-[~]
$ git clone https://github.com/linkedin/qark
Cloning into 'qark' ...
remote: Enumerating objects: 9314, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 9314 (delta 4), reused 3 (delta 1), pack-reused 9304
Receiving objects: 100% (9314/9314), 52.16 MiB | 10.44 MiB/s, done.
Resolving deltas: 100% (2809/2809), done.
(kevin㉿kali)-[~]
$ cd qark
(kevin㉿kali)-[~/qark]
$


(kevin㉿kali)-[~/qark]
$ virtualenv -p python3 venv
created virtual environment CPython3.11.6.final.0-64 in 1195ms
creator CPython3Posix(dest=/home/kevin/qark/venv, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/kevin/.local/share/virtualenv)
added seed packages: pip=23.3, setuptools=68.1.2, wheel=0.41.2
activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator


(kevin㉿kali)-[~/qark]
$ source venv/bin/activate


ython3/dist-packages/setuptools/__init__.py)
[end of output]
note: This error originates from a subprocess, and is likely not a problem with pip.
error: metadata-generation-failed
× Encountered error while generating package metadata.
└> See above for output.
note: This is an issue with the package mentioned above, not pip.
hint: See above for details.
┌──(venv)-(kevin㉿kali)-[~/qark]
└─$


(venv)kevin@kali: ~/qark
File Actions Edit View Help
Installed /usr/local/lib/python3.11/dist-packages/javalang-0.13.0-py3.11.egg
Searching for six=1.16.0
Best match: six 1.16.0
Adding six 1.16.0 to easy-install.pth file
detected new path './javalang-0.13.0-py3.11.egg'
Using /usr/lib/python3/dist-packages
Searching for requests=2.31.0
Best match: requests 2.31.0
Adding requests 2.31.0 to easy-install.pth file
Using /usr/lib/python3/dist-packages
Searching for pluginbase=1.0.1
Best match: pluginbase 1.0.1
Adding pluginbase 1.0.1 to easy-install.pth file
Using /usr/lib/python3/dist-packages
Searching for Jinja2=3.1.2
Best match: Jinja2 3.1.2
Adding Jinja2 3.1.2 to easy-install.pth file
Using /usr/lib/python3/dist-packages
Searching for click=8.1.6
Best match: click 8.1.6
Adding click 8.1.6 to easy-install.pth file
Using /usr/lib/python3/dist-packages
Finished processing dependencies for qark=4.0.0
(venv)-(kevin㉿kali)-[~/qark]
$


(venv)-(kevin㉿kali)-[~/qark]
$ sudo qark --apk ~/Downloads/modern_portfolio.apk
Decompiling...
dex2jar /home/kevin/qark/build/qark/classes.dex → /home/kevin/qark/build/qark/modern_portfolio.jar
I: Using Apktool 2.3.1 on modern_portfolio.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
S: WARNING: Could not write to (/root/.local/share/apktool/framework), using /tmp instead...

3) Analyzed the contents of the Modern Portfolio app.

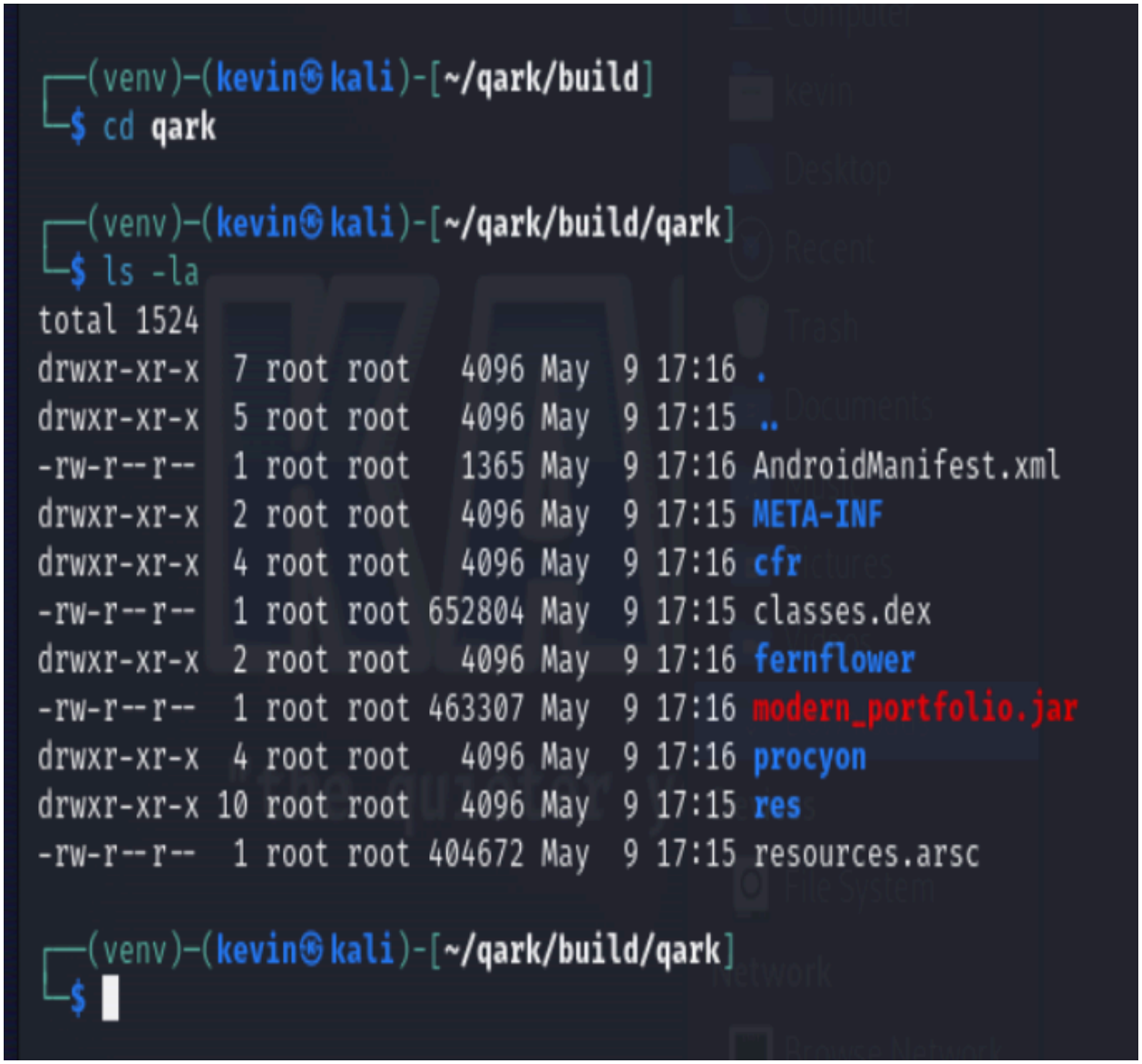
Using cat, I can see the following:
Some app details like the version, the Android platform version, some permissions, SDK version, and what are likely app components (like Main Activity)

I then looked through the disassembled source code and looked at possible problematic spots that could be vulnerable to SQL injection:

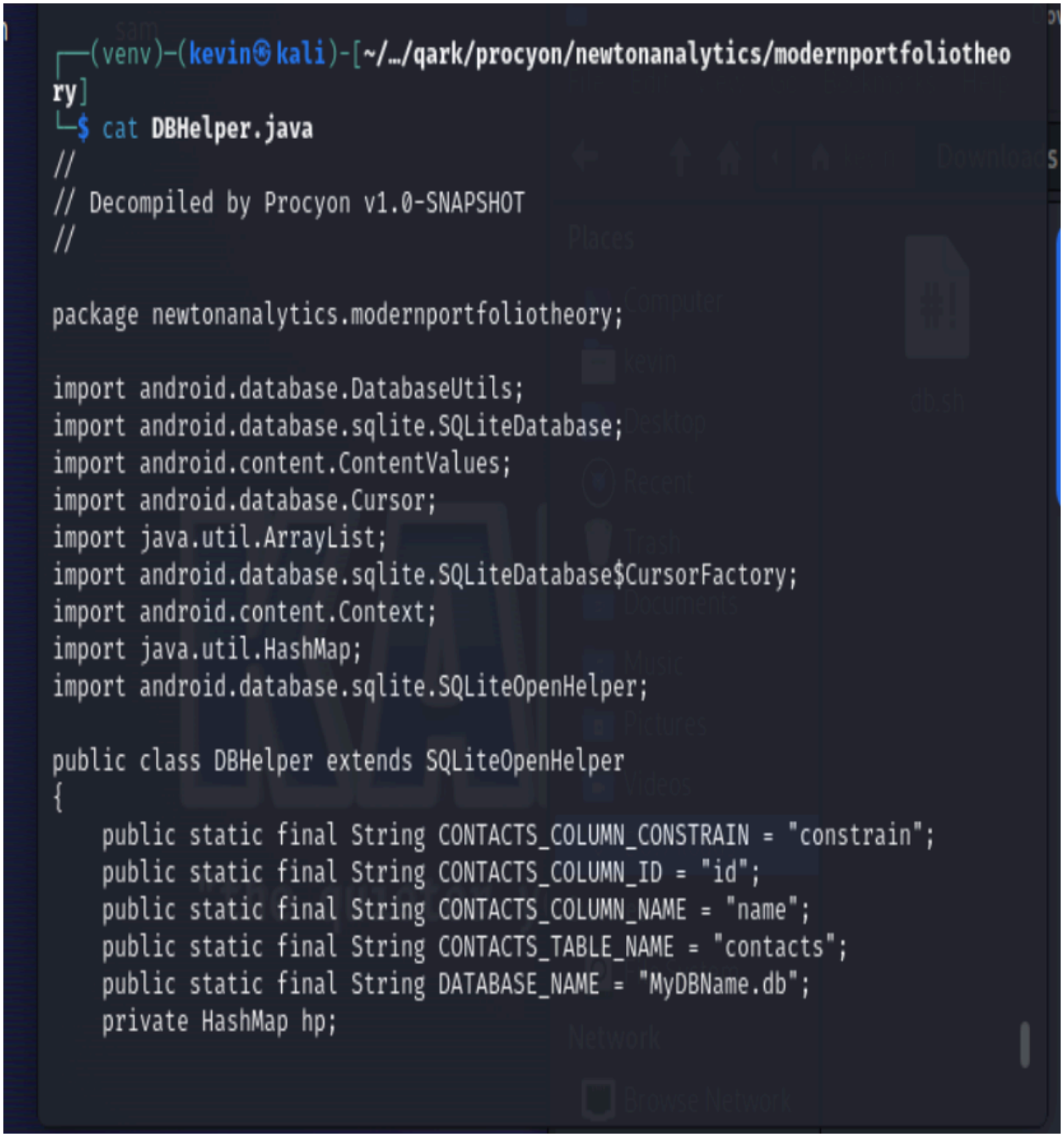
```
public boolean checkTicker(final String str) {
   return this.getReadableDatabase().rawQuery("select * from contacts where name='" +
str + "'", (String[])null).getCount() > 0;
}
```

This block of code uses raw SQL queries made from concatenating strings, making it vulnerable to injection.

Instead of using string concatenation, we could use placeholdesr in place of the parameters, and bind user input to those. By sanitizing the data by separating the code from it, we can prevent SQL injection attacks.


(venv)-(kevin㉿kali)-[~/qark/build]
$ cd qark
(venv)-(kevin㉿kali)-[~/qark/build/qark]
$ ls -la
total 1524
drwxr-xr-x 7 root root   4096 May 9 17:16 .
drwxr-xr-x 5 root root   4096 May 9 17:15 ..
-rw-r--r-- 1 root root   1365 May 9 17:16 AndroidManifest.xml
drwxr-xr-x 2 root root   4096 May 9 17:15 META-INF
drwxr-xr-x 4 root root   4096 May 9 17:16 cfr
-rw-r--r-- 1 root root 652804 May 9 17:15 classes.dex
drwxr-xr-x 2 root root   4096 May 9 17:16 fernflower
-rw-r--r-- 1 root root 463307 May 9 17:16 modern_portfolio.jar
drwxr-xr-x 4 root root   4096 May 9 17:16 procyon
drwxr-xr-x 10 root root   4096 May 9 17:15 res
-rw-r--r-- 1 root root 404672 May 9 17:15 resources.arsc
(venv)-(kevin㉿kali)-[~/qark/build/qark]
$

```
<?xml version="1.0" encoding="utf-8"?>
<manifest android:versionCode="2" android:versionName="1.1" package="newtonan
alytics.modernportfoliotheory" platformBuildVersionCode="18" platformBuildVer
sionName="4.3.1-1425645"
  xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-sdk android:minSdkVersion="14" android:targetSdkVersion="18" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application android:theme="@style/AppTheme" android:label="@string/app_n
ame" android:icon="@drawable/ic_launcher" android:allowBackup="true">
        <activity android:label="@string/app_name" android:name="newtonanalyt
ics.modernportfoliotheory.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:label="@string/title_activity_display_contact" andr
oid:name="newtonanalytics.modernportfoliotheory.DisplayContact" />
        <activity android:label="Your Settings" android:name="newtonanalytics
.modernportfoliotheory.UserSettingActivity" />
        <activity android:label="@string/title_activity_run" android:name="ne
wtonanalytics.modernportfoliotheory.Run" android:screenOrientation="portrait"
 />
    </application>
</manifest>
```



```
┌──(venv)-(kevin㉿kali)-[~/…/qark/procyon/newtonanalytics/modernportfoliotheo
ry]
└─$ cat DBHelper.java
//
// Decompiled by Procyon v1.0-SNAPSHOT
//

package newtonanalytics.modernportfoliotheory;

import android.database.DatabaseUtils;
import android.database.sqlite.SQLiteDatabase;
import android.content.ContentValues;
import android.database.Cursor;
import java.util.ArrayList;
import android.database.sqlite.SQLiteDatabase$CursorFactory;
import android.content.Context;
import java.util.HashMap;
import android.database.sqlite.SQLiteOpenHelper;

public class DBHelper extends SQLiteOpenHelper
{
    public static final String CONTACTS_COLUMN_CONSTRAIN = "constrain";
    public static final String CONTACTS_COLUMN_ID = "id";
    public static final String CONTACTS_COLUMN_NAME = "name";
    public static final String CONTACTS_TABLE_NAME = "contacts";
    public static final String DATABASE_NAME = "MyDBName.db";
    private HashMap hp;
```

4) Observed the html report and the Logging vulnerabilities. I found 2 Other Unique Vulnerabilities:

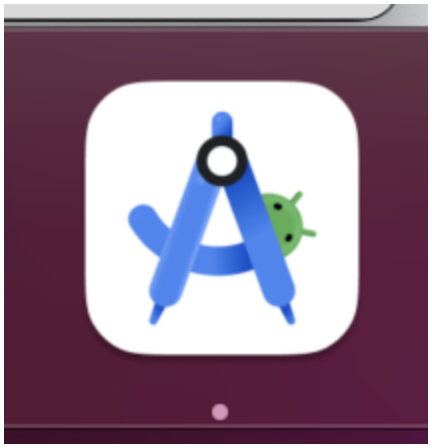
1) Vulnerability:
   This vulnerability occurs when a WebView is configured to allow JavaScript running in a file scheme context (local files) to access content from any origin.
2) Where the vulnerability exists & it's source code:
   webView.getSettings().setAllowUniversalAccessFromFileURLs(true);

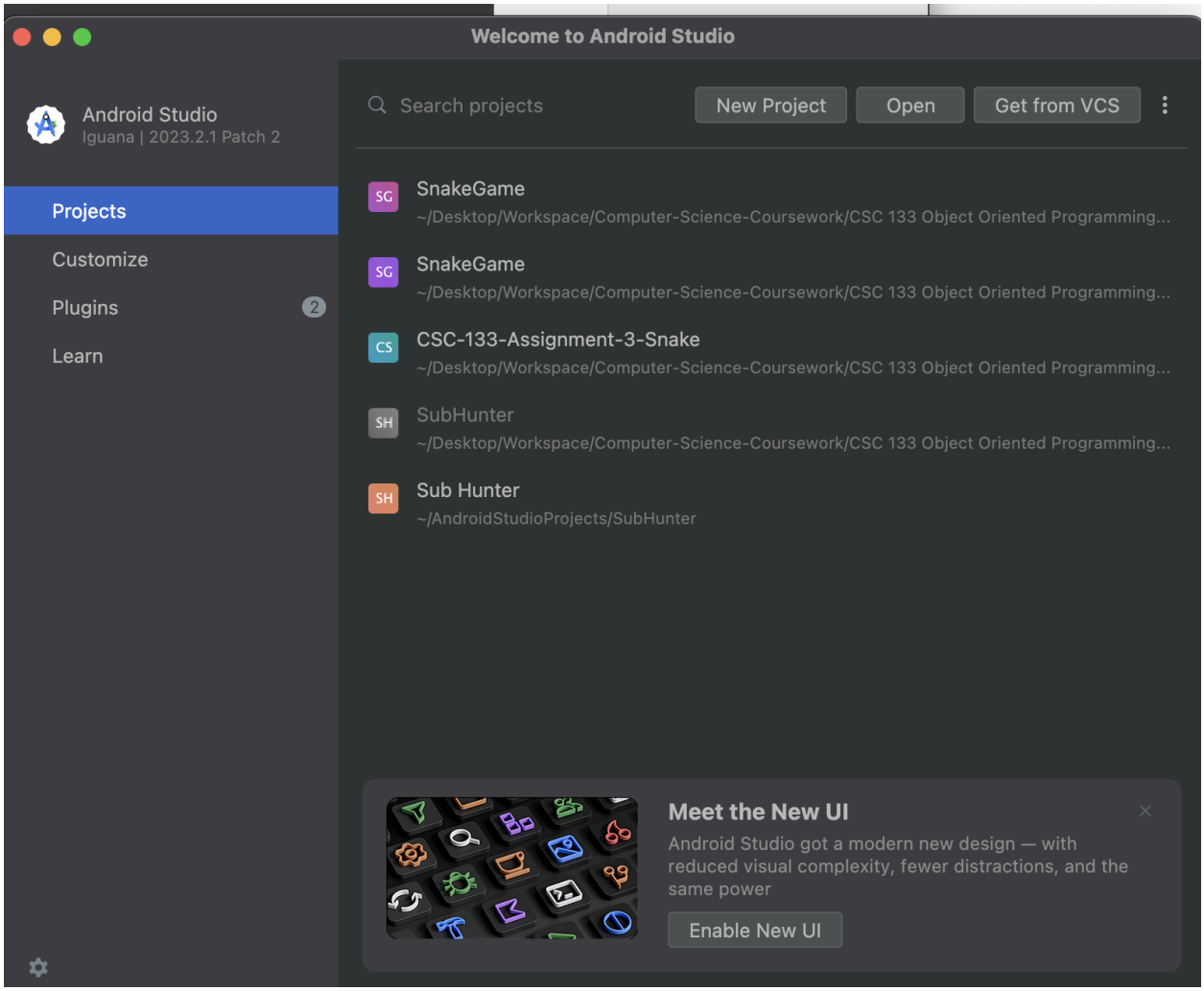
3) Severity, impact:
   High severity due to the potential for an attacker to exploit XSS vulnerabilities or access sensitive local file data. This could lead to data theft, session hijacking, and other malicious activities.
4) How to mitigate:
   Set setAllowUniversalAccessFromFileURLs to false unless absolutely necessary.
   Use input validation and sanitation to prevent XSS.

1) Vulnerability:
   Hardcoding HTTP URLs in the source code, especially for data transfer, exposes the application to man-in-the-middle (MITM) attacks because HTTP does not encrypt data transmission.
2) Where the vulnerability exists & it's source code:
   String url = "http://www.newtonanalytics.com/mobile/mpt.control.php?obs=";
3) Severity, impact:
   High severity as attackers could intercept, read, and potentially alter the data transmitted between the client and the server.
4) How to mitigate:
   Replace HTTP with HTTPS to ensure data is encrypted during transit.
   Avoid hardcoding URLs; instead, retrieve them from more secure, configurable sources.

Task 2 - Dynamic Analysis

1) I actually already had Android Studio installed on my Mac since I wanted to try mobile app development! It's also needed for Ali Ataya's 133 class.





2) Took a lot of debugging (was missing some files, Mac problems maybe?), but i launched the emulator from the Android SDK emulator directory and confirmed the list of AVDs.

```
home@Kevins-MacBook-Pro-8 emulator % ls
LICENSE                 emulator                nimble_bridge
NOTICE.csv              emulator-check          package.xml
NOTICE.txt              include                 qemu
android-info.txt        lib                     qemu-img
bin64                   lib64                   qsn
crashpad_handler        mksdcard                resources
crashreport             netsimd                 source.properties
home@Kevins-MacBook-Pro-8 emulator % ./emulator -list-avds

Pixel_3a_API_34_extension_level_7_x86_64
pixel_5_-_api_33
home@Kevins-MacBook-Pro-8 emulator %
```

```
home@Kevins-MacBook-Pro-8 emulator % ls $ANDROID_SDK_ROOT/system-images/androi
33/google_apis/x86_64

VerifiedBootParams.textproto    package.xml
advancedFeatures.ini            ramdisk.img
build.prop                      source.properties
data                            system.img
encryptionkey.img               userdata.img
kernel-ranchu                   vendor.img
home@Kevins-MacBook-Pro-8 emulator % ./emulator -avd pixel_5_-_api_33

INFO    | Android emulator version 33.1.24.0 (build_id 11237101) (CL:N/A)
INFO    | Found systemPath /Users/home/Library/Android/sdk/system-images/andro
-33/google_apis/x86_64/
INFO    | Storing crashdata in: , detection is enabled for process: 2145
INFO    | Duplicate loglines will be removed, if you wish to see each individua
 line launch with the -log-nofilter flag.
INFO    | Changing default hw.initialOrientation to portrait
INFO    | Increasing RAM size to 2048MB
library_mode host gpu mode host
I0509 18:14:37.675312 735a7c0 HealthMonitor.cpp:279] HealthMonitor disabled.
I0509 18:14:37.973433 735a7c0 FrameBuffer.cpp:486] Graphics Adapter Vendor Goog
e (ATI Technologies Inc.)
```


Android Emulator - pixel_5_-_api_33:5554

3) Downloaded the same 'Modern Portfolio' APK to my host computer as used in the previous task.

4) Navigated to the APK directories and installed the APK app w/ Android Debugger; **some commands may look a little different due to being on a Mac.**

Exploited a vulnerable intent in the 'Modern Portfolio' app, demonstrating how any app could launch MainActivity due to the lack of explicit intents or permissions.

```
home@Kevins-MacBook-Pro-8 ~ % cd ~/Library/Android/sdk/platform-tools/
```

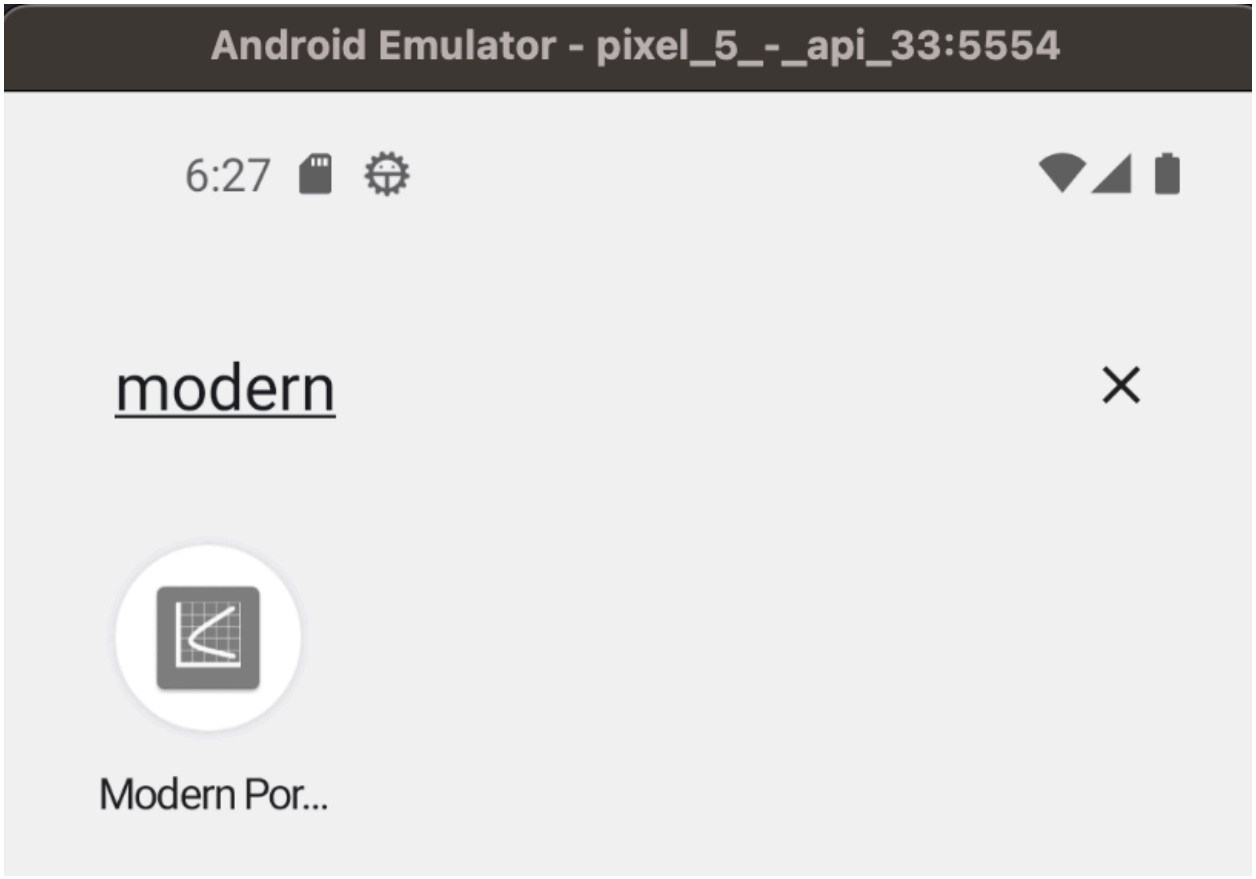
```
va:4313)
        at com.android.server.pm.PackageManag
nsact(PackageManagerService.java:5948)
        at android.os.Binder.%
home@Kevins-MacBook-Pro-8 platform-tools % ad
lio.apk

Performing Streamed Install
Success
home@Kevins-MacBook-Pro-8 platform-tools %
```

```
home@Kevins-MacBook-Pro-8 platform-tools % adb shell

emu64xa:/ $
```

```
package:com.android.cts.ctsshim
package:com.google.android.apps.photos
package:com.android.bluetooth
package:com.google.android.markup
package:com.android.emulator.radio.con
package:com.android.internal.display.c
package:com.google.android.gms
package:com.android.storagemanager
package:com.google.android.sdksetup
package:com.android.printspooler
package:com.android.systemui.auto_gene
package:com.android.providers.partnerb
package:com.android.soundpicker
package:com.google.mainline.telemetry
package:com.android.dynsystem
package:com.android.providers.telephon
package:com.google.android.connectivit
package:com.android.bips.auto_generate
package:com.google.android.youtube
package:com.android.simappdialog.auto_
package:com.android.externalstorage
package:com.android.server.telecom
emu64xa:/ $
```


Android Emulator - pixel_5_-_api_33:5554
6:27
modern
Modern Por...

```
emu64xa:/ $ am start -n newtonanalytics.modernportfoliotheory/.MainActivity
Starting: Intent { cmp=newtonanalytics.modernportfoliotheory/.MainActivity }
emu64xa:/ $
```