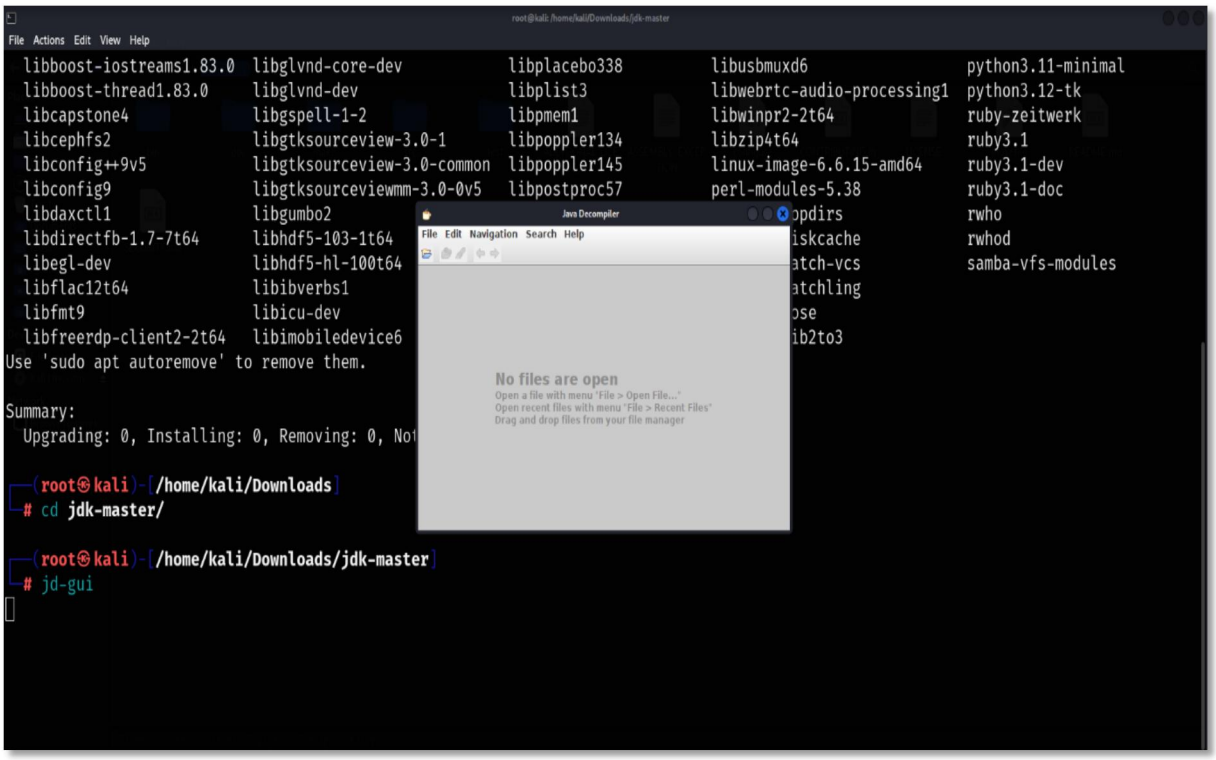
LAB – 12

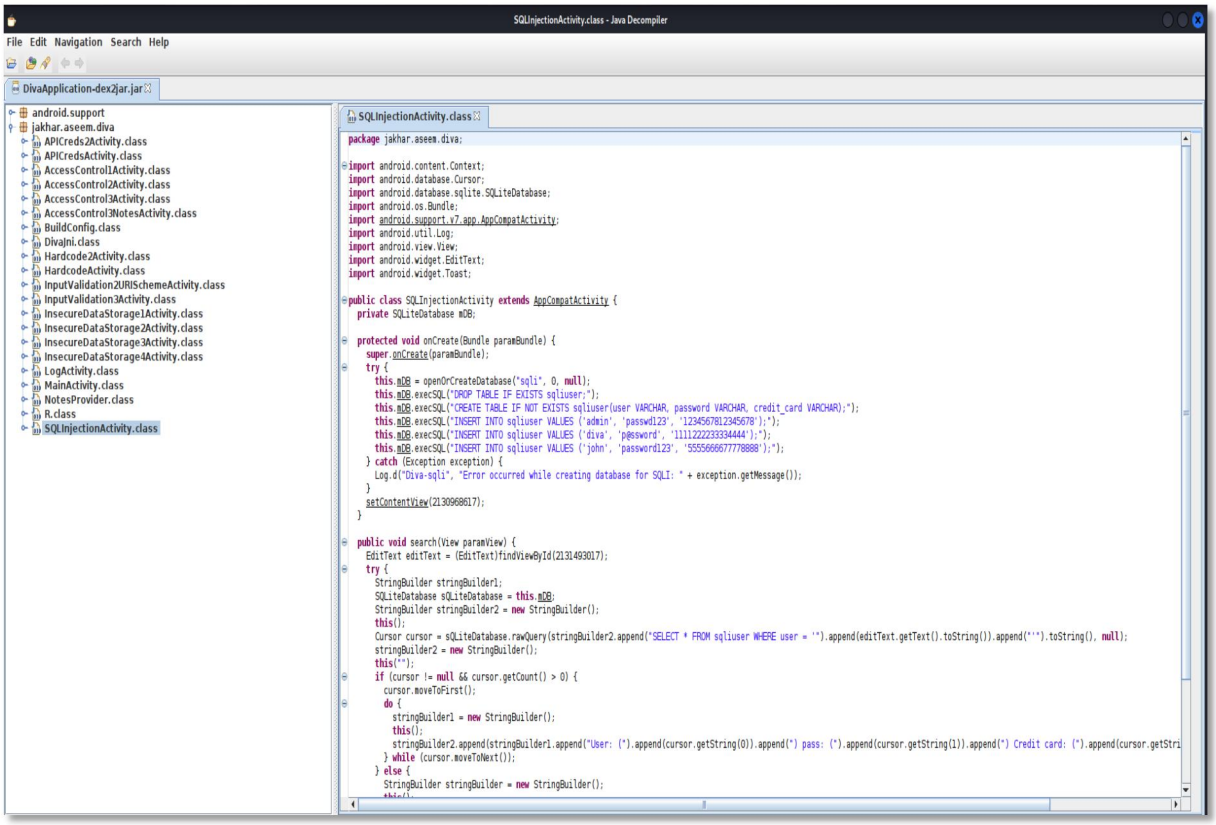
Aim :- Opening jar file in jdx GUI and find hardcore strings.

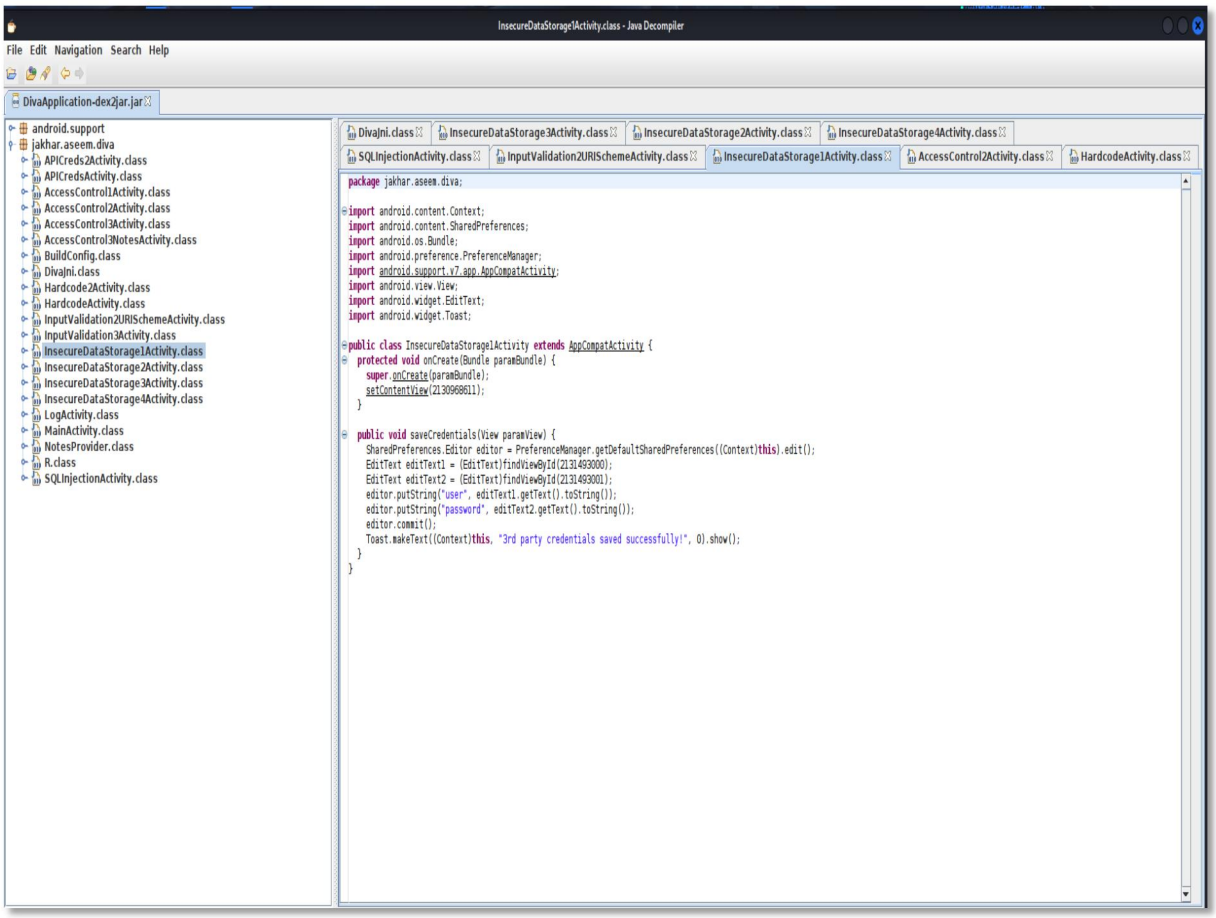
Definition:

Opening a JAR file in JD-GUI and finding hardcoded strings refers to the process of analyzing a Java archive file (usually converted from an APK) in JD-GUI, a Java decompiler. JD-GUI allows you to view the decompiled Java source code inside a .jar file. By exploring the code, you can search for hardcoded strings — like API keys, passwords, URLs, tokens, or any static values embedded directly in the source code — which can pose security risks if not handled properly.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install jd-gui
[sudo] password for kali:
The following packages were automatically installed and are no longer required:
  crackmapexec              libfreerdp2-2t64          libiniparser1          libpython3.12t64          python3-mistune0
  firebird3.0-common        libgdal34t64              libjim0.82t64          libqt5sensors5            python3-pathspec
  firebird3.0-common-doc    libgeos3.12.1t64          libjsoncpp25           libqt5webkit5             python3-pendulum
  fonts-liberation2         libgeos3.13.0             liblbfgsb0             librados2                 python3-pluggy
  freerdp2-x11              libgfapi0                 libmbedcrypto7t64      librdmacm1t64             python3-pytzdata
  hydra-gtk                 libgfrpc0                 libmfx1                libre2-10                 python3-rsa
  ibverbs-providers         libgfxdr0                 libmimalloc3           libroc0.3                 python3-setproctitle
  icu-devtools              libgl1-mesa-dev           libmsgraph-0-1         libsuperlu6               python3-setuptools-scm
  libarmadillo12            libglapi-mesa             libndctl6              libtag1v5                 python3-time-machine
  libassuan0                libgles-dev               libnetcdf19t64         libtag1v5-vanilla         python3-trove-classifiers
  libavfilter9              libgles1                  libpaper1              libtagc0                  python3.11
  libbfio1                  libglusterfs0             libperl5.38t64         libu2f-udev               python3.11-dev
  libboost-iostreams1.83.0  libglvnd-core-dev         libplacebo338          libusbmuxd6               python3.11-minimal
  libboost-thread1.83.0     libglvnd-dev              libplist3              libwebrtc-audio-processing1  python3.12-tk
  libcapstone4              libgspell-1-2             libpmem1               libwinpr2-2t64            ruby-zeitwerk
  libcephfs2                libgtksourceview-3.0-1    libpoppler134          libzip4t64                ruby3.1
  libconfig++9v5            libgtksourceview-3.0-common  libpoppler145       linux-image-6.6.15-amd64  ruby3.1-dev
  libconfig9                libgtksourceviewmm-3.0-0v5  libpostproc57        perl-modules-5.38         ruby3.1-doc
  libdaxctl1                libgumbo2                 libpython3.11-dev      python3-appdirs           rwho
  libdirectfb-1.7-7t64      libhdf5-103-1t64          libpython3.11-minimal  python3-diskcache         rwhod
  libegl-dev                libhdf5-hl-100t64         libpython3.11-stdlib   python3-hatch-vcs         samba-vfs-modules
  libflac12t64              libibverbs1               libpython3.11t64       python3-hatchling
  libfmt9                   libicu-dev                libpython3.12-minimal  python3-jose
  libfreerdp-client2-2t64   libimobiledevice6        libpython3.12-stdlib   python3-lib2to3
Use 'sudo apt autoremove' to remove them.

Installing:
  jd-gui

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 233
  Download size: 1287 kB
  Space needed: 1500 kB / 9760 MB available

Get:1 http://kali.download/kali kali-rolling/main amd64 jd-gui all 1.6.6-0kali1 [1287 kB]
Fetched 1287 kB in 5s (277 kB/s)
Selecting previously unselected package jd-gui.
(Reading database ... 463891 files and directories currently installed.)
Preparing to unpack .../jd-gui_1.6.6-0kali1_all.deb ...
Unpacking jd-gui (1.6.6-0kali1) ...
Setting up jd-gui (1.6.6-0kali1) ...
Processing triggers for kali-menu (2025.1.1) ...

┌──(kali㉿kali)-[~]
└─$
```


root@kali: /home/kali/Downloads/jdk-master
File Actions Edit View Help
libboost-iostreams1.83.0 libglvnd-core-dev libplacebo338 libusbmuxd6 python3.11-minimal
libboost-thread1.83.0 libglvnd-dev libplist3 libwebrtc-audio-processing1 python3.12-tk
libcapstone4 libgspell-1-2 libpmem1 libwinpr2-2t64 ruby-zeitwerk
libcephfs2 libgtksourceview-3.0-1 libpoppler134 libzip4t64 ruby3.1
libconfig++9v5 libgtksourceview-3.0-common libpoppler145 linux-image-6.6.15-amd64 ruby3.1-dev
libconfig9 libgtksourceviewmm-3.0-0v5 libpostproc57 perl-modules-5.38 ruby3.1-doc
libdaxctl1 libgumbo2 opdirs rwho
libdirectfb-1.7-7t64 libhdf5-103-1t64 iskcache rwhod
libegl-dev libhdf5-hl-100t64 atch-vcs samba-vfs-modules
libflac12t64 libibverbs1 atchling
libfmt9 libicu-dev ose
libfreerdp-client2-2t64 libimobiledevice6 ib2to3
Use 'sudo apt autoremove' to remove them.
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not
Java Decompiler
File Edit Navigation Search Help
No files are open
Open a file with menu 'File > Open File...'
Open recent files with menu 'File > Recent Files'
Drag and drop files from your file manager
root kali /home/kali/Downloads
cd jdk-master/
root kali /home/kali/Downloads/jdk-master
jd-gui


SQLInjectionActivity.class - Java Decompiler
File Edit Navigation Search Help
DivaApplication-dex2jar.jar
android.support
jakhar.aseem.diva
APICreds2Activity.class
APICredsActivity.class
AccessControl1Activity.class
AccessControl2Activity.class
AccessControl3Activity.class
AccessControl3NotesActivity.class
BuildConfig.class
DivaJni.class
Hardcode2Activity.class
HardcodeActivity.class
InputValidation2URISchemeActivity.class
InputValidation3Activity.class
InsecureDataStorage1Activity.class
InsecureDataStorage2Activity.class
InsecureDataStorage3Activity.class
InsecureDataStorage4Activity.class
LogActivity.class
MainActivity.class
NotesProvider.class
R.class
SQLInjectionActivity.class
SQLInjectionActivity.class
package jakhar.aseem.diva;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
public class SQLInjectionActivity extends AppCompatActivity {
private SQLiteDatabase mDB;
protected void onCreate(Bundle paramBundle) {
super.onCreate(paramBundle);
try {
this.mDB = openOrCreateDatabase("sqli", 0, null);
this.mDB.execSQL("DROP TABLE IF EXISTS sqliuser;");
this.mDB.execSQL("CREATE TABLE IF NOT EXISTS sqliuser(user VARCHAR, password VARCHAR, credit_card VARCHAR);");
this.mDB.execSQL("INSERT INTO sqliuser VALUES ('admin', 'passwd123', '1234567812345678');");
this.mDB.execSQL("INSERT INTO sqliuser VALUES ('diva', 'p@ssword', '1111222233334444');");
this.mDB.execSQL("INSERT INTO sqliuser VALUES ('john', 'password123', '5555666677778888');");
} catch (Exception exception) {
Log.d("Diva-sqli", "Error occurred while creating database for SQLI: " + exception.getMessage());
}
setContentView(2130968617);
}
public void search(View paramView) {
EditText editText = (EditText)findViewById(2131493017);
try {
StringBuilder stringBuilder1;
SQLiteDatabase sQLiteDatabase = this.mDB;
StringBuilder stringBuilder2 = new StringBuilder();
this();
Cursor cursor = sQLiteDatabase.rawQuery(stringBuilder2.append("SELECT * FROM sqliuser WHERE user = '").append(editText.getText().toString()).append("'").toString(), null);
stringBuilder2 = new StringBuilder();
this("");
if (cursor != null && cursor.getCount() > 0) {
cursor.moveToFirst();
do {
stringBuilder1 = new StringBuilder();
this();
stringBuilder2.append(stringBuilder1.append("User: (").append(cursor.getString(0)).append(") pass: (").append(cursor.getString(1)).append(") Credit card: (").append(cursor.getStri
} while (cursor.moveToNext());
} else {
StringBuilder stringBuilder = new StringBuilder();



Conclusion :-

Using JD-GUI to open a JAR file and find hardcoded strings is a valuable technique for reverse engineering and security analysis. It helps identify sensitive data (like API keys, tokens, or credentials) that may have been unintentionally hardcoded in the app's source code. This process is essential for penetration testers, ethical hackers, or developers performing code audits. However, it should only be used on your own apps or with permission to ensure ethical and legal compliance.