I have been practicing bug bounty in HackerOne and BugCroud sites. Currently there are few applications in bugCroud that takes testers for new joining hackers. Therefore I have tried an Android application in HackerOne.

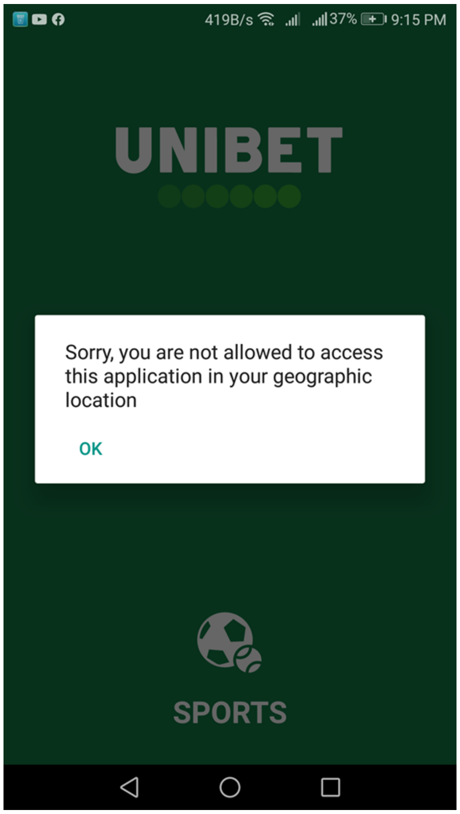
Details on the bugs found

Program: Kindred Group   https://www.kindredgroup.com

The mobile application is a live sports betting application. I tried to run the application in two emulators

1. Android emulator
2. Genymotion emulator

But was unsuccessful as the application did not even install. Then tried in my own phone and the application successfully installed but did not run as it checks for the geographical location the app is run. I got a message that the application doesn't run in my current geographical area.



Sorry, you are not allowed to access this application in your geographic location

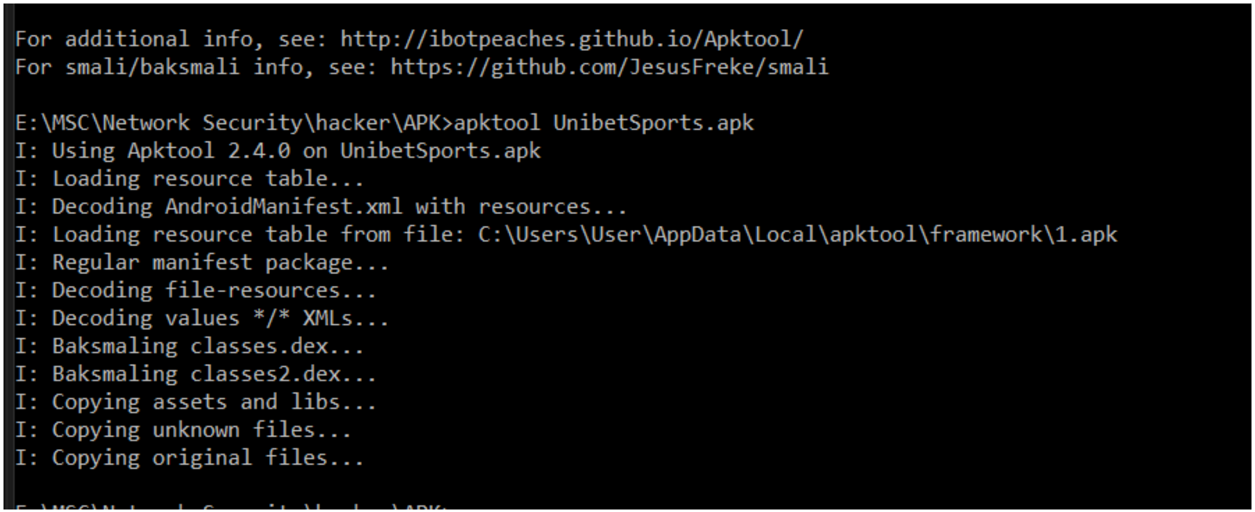
OK

Bug – 01

Vulnerability : Backup mode enabled

How it was found :

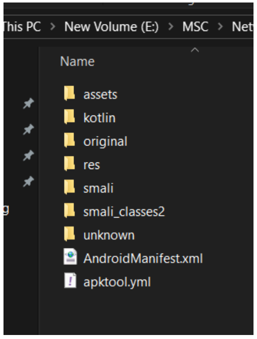
1. App was downloaded from HackerOne site
2. Decompiled using the APKTool



```
For additional info, see: http://ibotpeaches.github.io/Apktool/
For smali/baksmali info, see: https://github.com/JesusFreke/smali

E:\MSC\Network Security\hacker\APK>apktool UnibetSports.apk
I: Using Apktool 2.4.0 on UnibetSports.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:\Users\User\AppData\Local\apktool\framework\1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...

E:\MSC\Network Security\hacker\APK>
```



his PC › New Volume (E:) › MSC › Net

Name

assets
kotlin
original
res
smali
smali_classes2
unknown
AndroidManifest.xml
apktool.yml

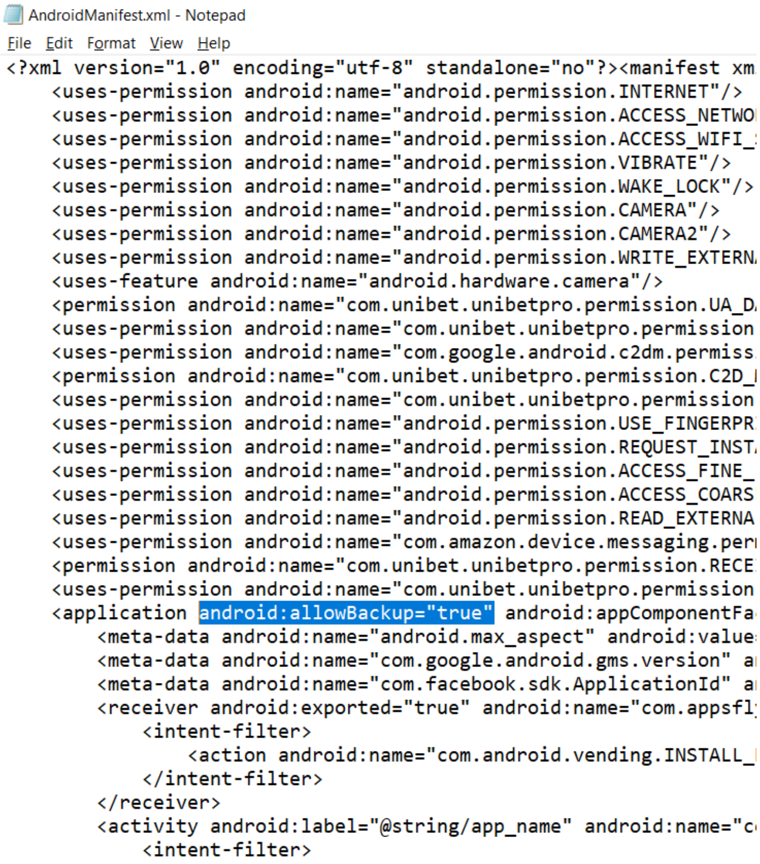
3. Opened the Manifest file

   Manifest file is located at the root of the application. The manifest file describes essential information about your app to the Android build tools, the Android operating system, and Google Play.

4. Searched for the work “backup” and found that the application provides permission to take backups by setting the value true.

The application allows to take backup “application android:allowBackup="true”” .

This allows attackers to obtain sensitive cleartext information via an "adb backup '-f APP_name" command. By using “adb backup” command an attacker can back up the entire device. It's very useful to collect all the data the application is handling. It can back up all APKs and their data, your personal files, downloads, music, pics, etc. It's up to you which parameters you set for the degree of backup.



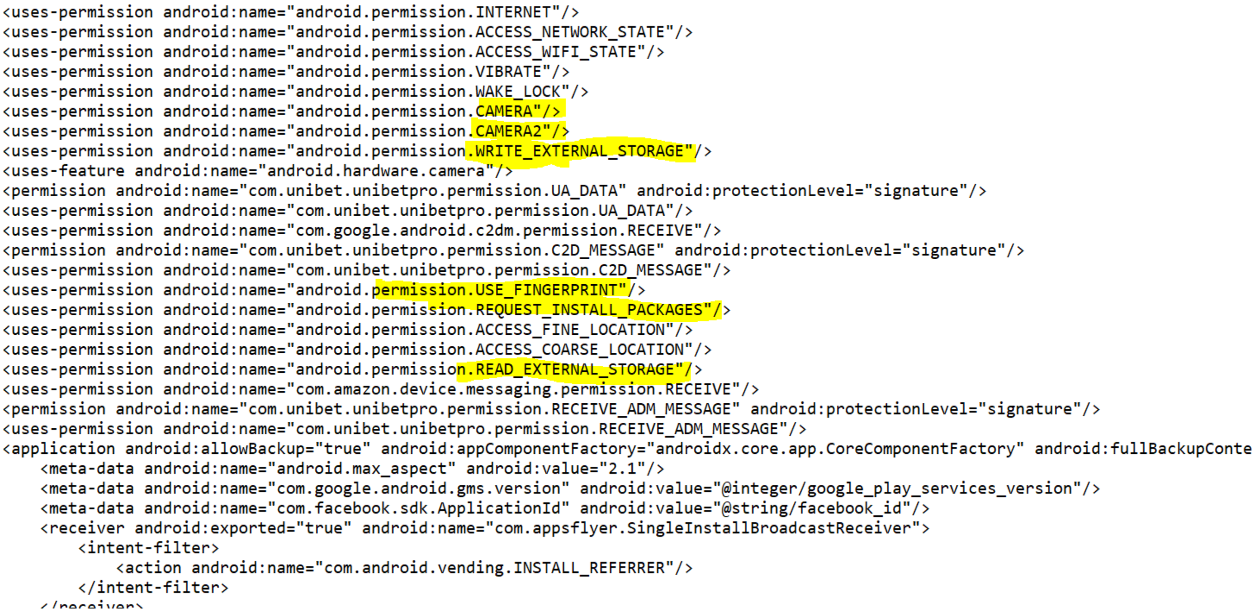
```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xm
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWO
    <uses-permission android:name="android.permission.ACCESS_WIFI_
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.CAMERA2"/>
    <uses-permission android:name="android.permission.WRITE_EXTERN
    <uses-feature android:name="android.hardware.camera"/>
    <permission android:name="com.unibet.unibetpro.permission.UA_D
    <uses-permission android:name="com.unibet.unibetpro.permission
    <uses-permission android:name="com.google.android.c2dm.permiss
    <permission android:name="com.unibet.unibetpro.permission.C2D_
    <uses-permission android:name="com.unibet.unibetpro.permission
    <uses-permission android:name="android.permission.USE_FINGERPR
    <uses-permission android:name="android.permission.REQUEST_INST
    <uses-permission android:name="android.permission.ACCESS_FINE_
    <uses-permission android:name="android.permission.ACCESS_COARS
    <uses-permission android:name="android.permission.READ_EXTERNA
    <uses-permission android:name="com.amazon.device.messaging.per
    <permission android:name="com.unibet.unibetpro.permission.RECE
    <uses-permission android:name="com.unibet.unibetpro.permission
    <application android:allowBackup="true" android:appComponentFa
        <meta-data android:name="android.max_aspect" android:value
        <meta-data android:name="com.google.android.gms.version" a
        <meta-data android:name="com.facebook.sdk.ApplicationId" a
        <receiver android:exported="true" android:name="com.appsfl
            <intent-filter>
                <action android:name="com.android.vending.INSTALL_
            </intent-filter>
        </receiver>
        <activity android:label="@string/app_name" android:name="c
            <intent-filter>
```

Further investigations are required on

- If there are any backup rules are defined and if so need to check if any sensitive details are allowed to backup.
- Need to check with burp if plaintext data are being sent. (Have an issue in running the app due to GPS check on location)
- Can use logcat and check if any logs are being maintained.

Bug – 02

The application has many unused permissions allowed. The developers must be able to provide justifications for using these permissions. Or else these must be removed.



```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.CAMERA2"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-feature android:name="android.hardware.camera"/>
<permission android:name="com.unibet.unibetpro.permission.UA_DATA" android:protectionLevel="signature"/>
<uses-permission android:name="com.unibet.unibetpro.permission.UA_DATA"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<permission android:name="com.unibet.unibetpro.permission.C2D_MESSAGE" android:protectionLevel="signature"/>
<uses-permission android:name="com.unibet.unibetpro.permission.C2D_MESSAGE"/>
<uses-permission android:name="android.permission.USE_FINGERPRINT"/>
<uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="com.amazon.device.messaging.permission.RECEIVE"/>
<permission android:name="com.unibet.unibetpro.permission.RECEIVE_ADM_MESSAGE" android:protectionLevel="signature"/>
<uses-permission android:name="com.unibet.unibetpro.permission.RECEIVE_ADM_MESSAGE"/>
<application android:allowBackup="true" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:fullBackupConte
    <meta-data android:name="android.max_aspect" android:value="2.1"/>
    <meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version"/>
    <meta-data android:name="com.facebook.sdk.ApplicationId" android:value="@string/facebook_id"/>
    <receiver android:exported="true" android:name="com.appsflyer.SingleInstallBroadcastReceiver">
        <intent-filter>
            <action android:name="com.android.vending.INSTALL_REFERRER"/>
        </intent-filter>
    </receiver>
```

Bug – 03

Application source code is not obfuscated

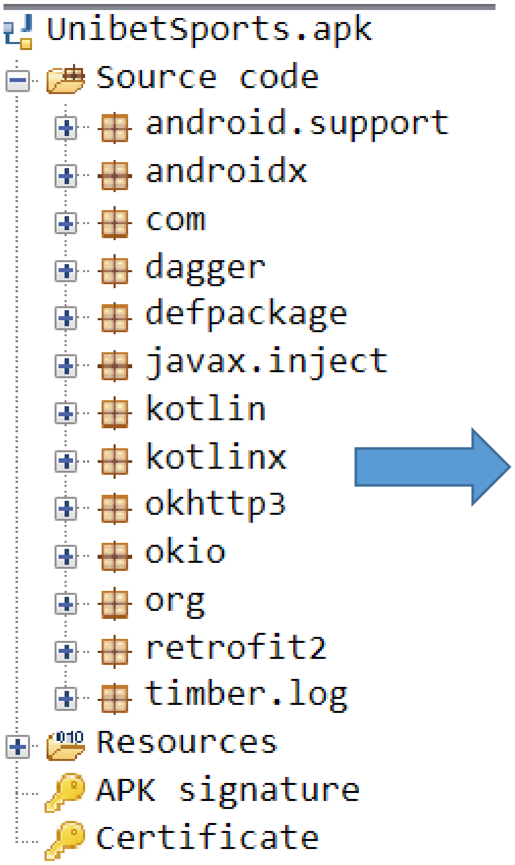
Used tools : jadx-gui-1.0.0

Jadex tool is used to convert the Dex codes to Decompiled java files .(Human readable)

By opening the application it was seen that the file has not been obfuscated, therefore all the files in the app are human readable making it more vulnerable to attacks. In software development, obfuscation is the deliberate act of creating source or machine code that is difficult for humans to understand. Like obfuscation in natural language, it may use needlessly roundabout expressions to compose statements

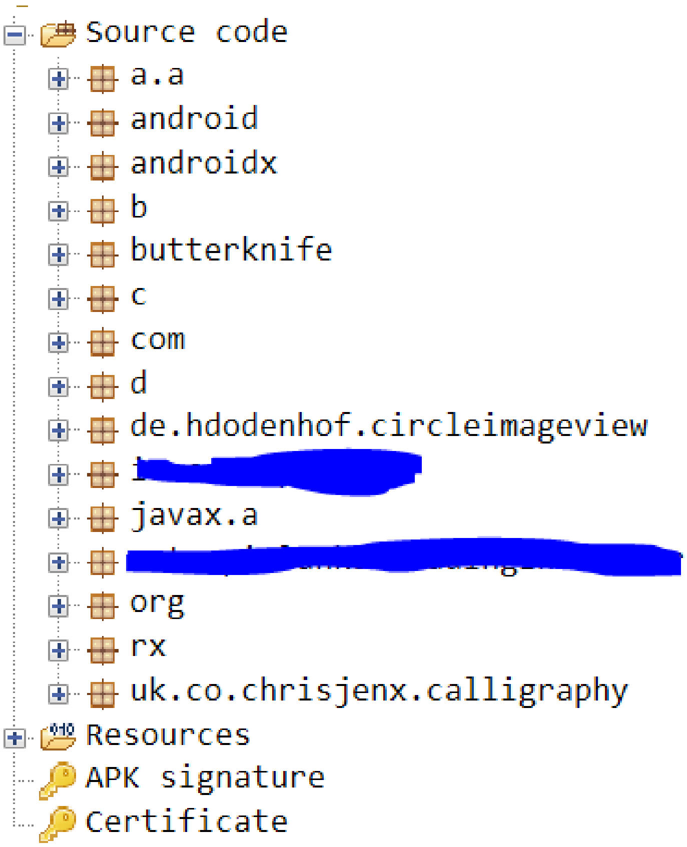
Basic obfuscations can be done by configuring android studio or else can purchase obfuscating tools for this.

How it looks originally



```
UnibetSports.apk
  Source code
    android.support
    androidx
    com
    dagger
    defpackage
    javax.inject
    kotlin
    kotlinx
    okhttp3
    okio
    org
    retrofit2
    timber.log
  Resources
  APK signature
  Certificate
```

How it should look after obfuscation

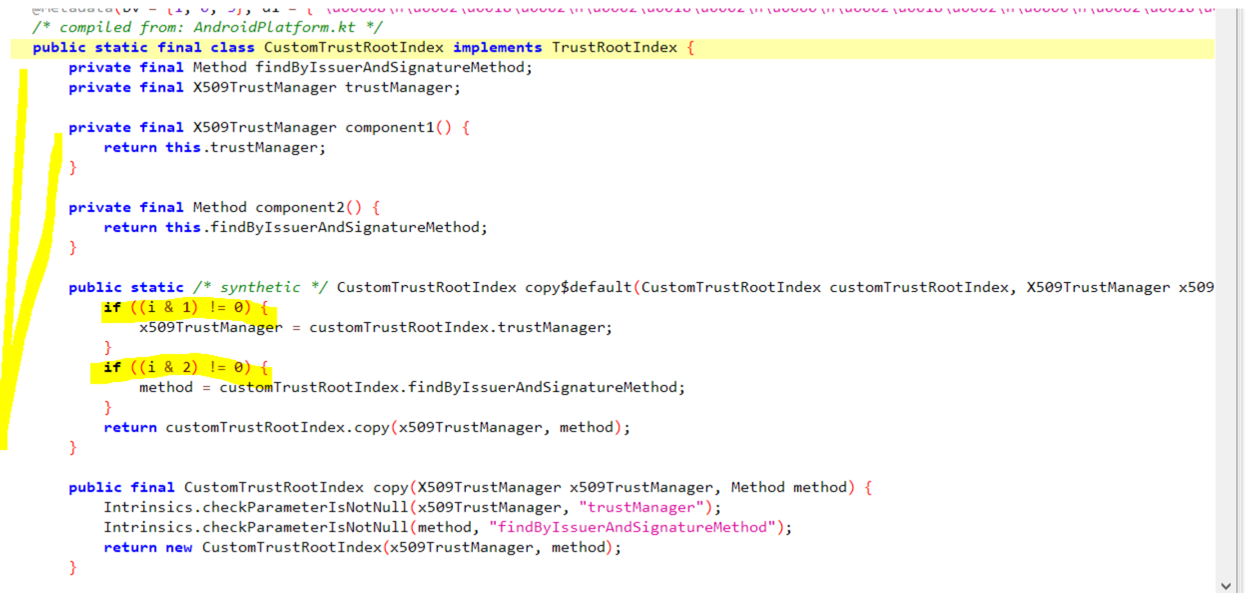


```
Source code
  a.a
  android
  androidx
  b
  butterknife
  c
  com
  d
  de.hdodenhof.circleimageview
  [redacted]
  javax.a
  [redacted]
  org
  rx
  uk.co.chrisjenx.calligraphy
Resources
APK signature
Certificate
```

Bug – 04

Possibility to bypass root access of the application

The if condition can be bypassed and hopefully root check can be bypassed. Need to do more source code review and try this.



```
/* compiled from: AndroidPlatform.kt */
public static final class CustomTrustRootIndex implements TrustRootIndex {
    private final Method findByIssuerAndSignatureMethod;
    private final X509TrustManager trustManager;

    private final X509TrustManager component1() {
        return this.trustManager;
    }

    private final Method component2() {
        return this.findByIssuerAndSignatureMethod;
    }

    public static /* synthetic */ CustomTrustRootIndex copy$default(CustomTrustRootIndex customTrustRootIndex, X509TrustManager x509
        if ((i & 1) != 0) {
            x509TrustManager = customTrustRootIndex.trustManager;
        }
        if ((i & 2) != 0) {
            method = customTrustRootIndex.findByIssuerAndSignatureMethod;
        }
        return customTrustRootIndex.copy(x509TrustManager, method);
    }

    public final CustomTrustRootIndex copy(X509TrustManager x509TrustManager, Method method) {
        Intrinsics.checkParameterIsNotNull(x509TrustManager, "trustManager");
        Intrinsics.checkParameterIsNotNull(method, "findByIssuerAndSignatureMethod");
        return new CustomTrustRootIndex(x509TrustManager, method);
    }
```