PACKING YOUR ANDROID

THE HITCHHACKERS GUIDE TO ANDROID UNBOXING

Presenter: Swapnil Deshmukh

Event: BSides Philly 2017

Date: 08/12/2017

DISCLAIMER!

The views and opinions expressed in this *presentation* are those of the author's and do not necessarily represent official policy or position of any other individual or company.

$WHOAMI

- Co-authored Hacker's Handbook series
- Lead Security Team for Emerging Technology
- Malware Researcher

- twitter - @sec2_0
- Poppy Seed project @github - https://github.com/sdswapz
- Malware Repo - https://malware.swapnil.me

WHY ARE WE HERE

WHY ARE WE HERE

- Easy Target
- Millions are getting Impacted
- Sophos report - Android Malware Sample Growth
  - 31% YOY growth
  - One in three new apps released has Malicious Genome

CHALLENGES WITH HARDENED ANDROID APPLICATION

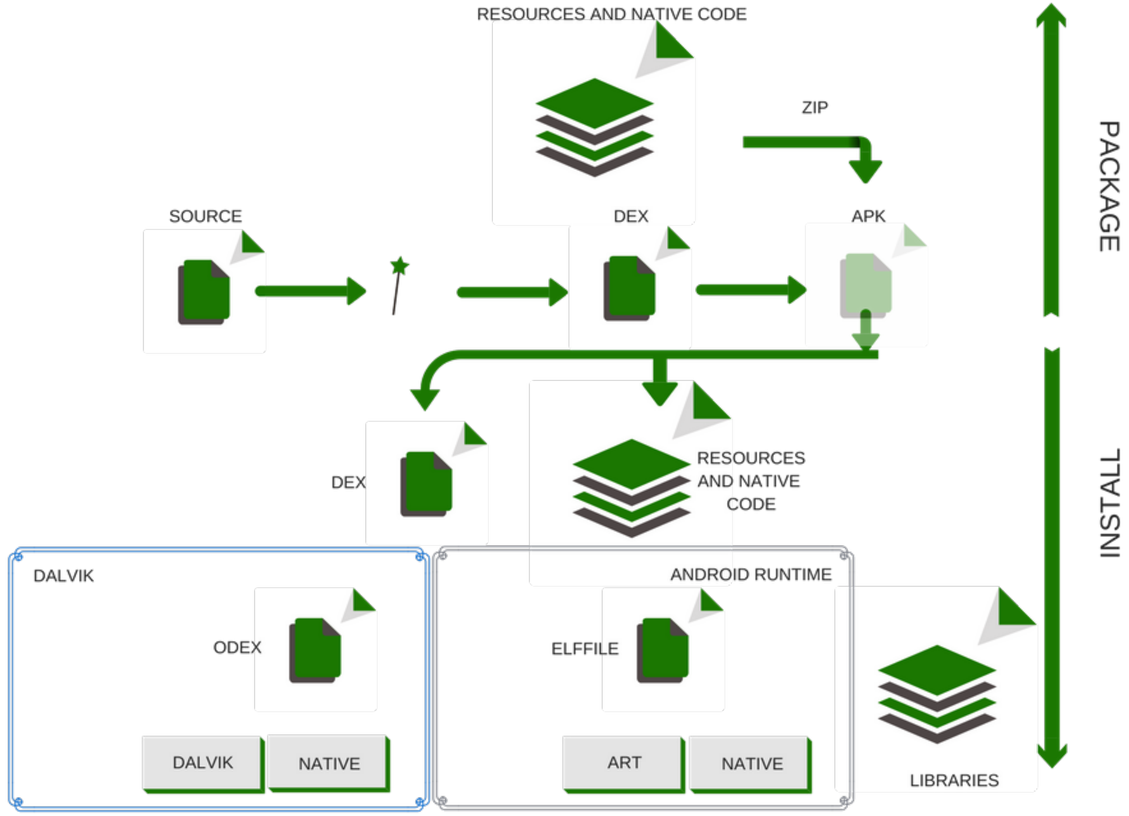
- Obfuscation
- Root Detection Checks and Anti-Debugging
- Anti-Tampering
- Anti-Emulation
- White Box Cryptography

WHAT IS ALREADY OUT THERE

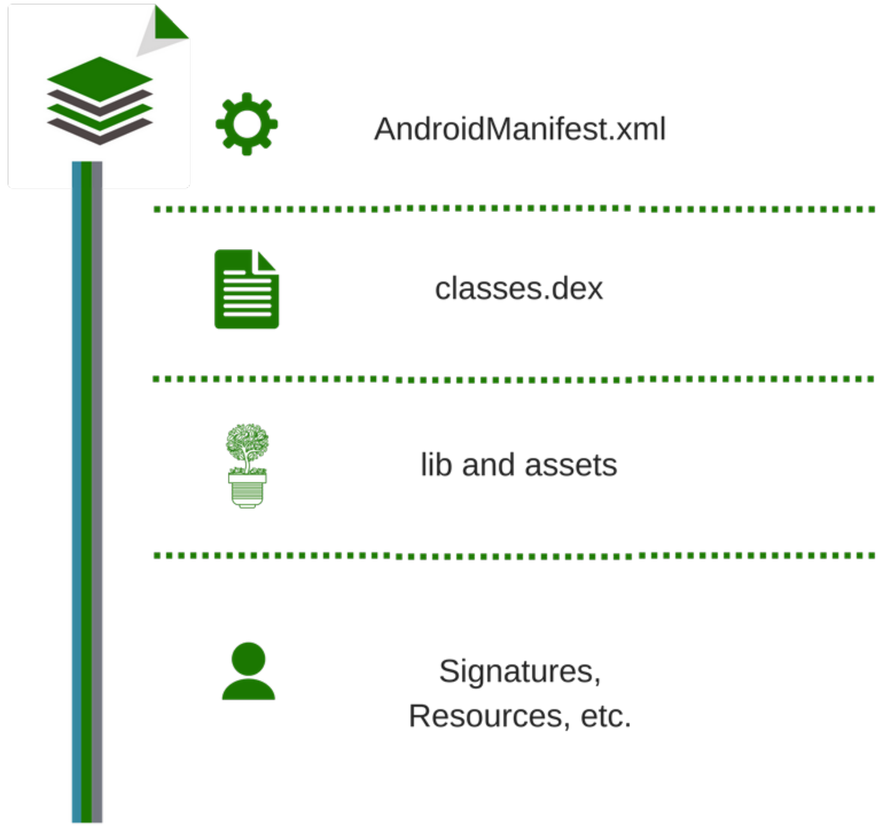
- [Defcon 22] Android Hacker Protection Level 0
- [HITCON 2015] The Terminator to Android Hardening Services
- [HITCON 2016] Android Compiler Fingerprinting
- … and many more

LIFE OF AN APK

LIFE OF AN APK



UNDER THE HOOD – APK



OBFUSCATES

OBFUSCATORS AND OPTIMIZERS

- Scrambles Data
  - Class, Method and Variable Names
  - Control Flow Obfuscation/ Reordering
  - String Encryption
- Java Reflect

PROGUARD

```
@Override public CharSequence onDisableRequested(Context context, Intent intent) {
    String string = Integer.toString(1008);
    this.getManager(context).lockNow();
    this.getManager(context).resetPassword(string, 0);
    return super.onDisableRequested(context, intent);
}

@Override public void onEnabled(Context context, Intent intent) {
    Class class0;
    MyAdmin myAdmin = this;
    Context context = context;
    Intent intent = intent;
    String string = Integer.toString(1008);
    Intent intent = null;
    Intent intent = null;
    Context context = context;
    try {
        class0 = Class.forName("com.h.s");
    }
    catch(ClassNotFoundException classNotFoundException0) {
        throw new NoClassDefFoundError(classNotFoundException0.getMessage());
    }

    super(context, class0);
    intent.setFlags(268435456);
    context.startService(intent);
    myAdmin.getManager(context).resetPassword(string, 0);
    super.onEnabled(context, intent);
}
```

DEXPROTECTOR

```
    }
    public void onCreate() {
        //Before the first component of the application starts
        super.onCreate()
        final Method declareMethod = Class.forname("dexprotector.name").getDeclaredMethod("run",hash);
        declaredMethod.setAccessible(true);
        declaredMethod.invoke(this, "com.dailyworkout.tizi.activities.ActivitySplash")
```

```
    public void onCreate() {
        super.onCreate();
        final Method declaredMethod = Class.forName(onCreate("\ua5ac\uc28c\u6b09\u67c8\ub589\u1543\u6148\uaf99\u4643\ud5f8\u8d09\udbe4\uf4ae\u1090\uca52\ub6a3\uc47c
        declaredMethod.setAccessible(true);
        declaredMethod.invoke(this, onCreate("\u6805\u4fbd\uc65f\u132f\u8502\u6413\u1db1\u3da9\uedee\u4c79\ud2e3\u6bf3\u1e81\u0559\u6ba2\uee61"));
    }
}
```

ARXAN ENSUREIT

```
import android.app.Activity;
import hhhhhh.whhwwh;

public class TestActivity extends Activity {
    public static int f3b04440444044404440444 = 70;
    public static int f4b0413041304130413 = 0;
    public static int f5b0413 = 2;
    public static boolean f6b041304130413 = false;
    public static int f7b04440444 = 1;
    private whhwwh f8b0413041304130413;

    static {
        try {
            int b0444ϕϕϕ04440444 = m3b044404440444() + f7b04440444;
            if (((f3b044404440444044404440444 + f7b04440444) * f3b044404440444044404440444) % f5b0413 != f4b0413041304130413) {
                f3b044404440444044404440444 = m3b044404440444();
                f4b0413041304130413 = m3b044404440444();
            }
            try {
                if ((b0444ϕϕϕ04440444 * m3b044404440444()) % f5b0413 != f4b0413041304130413) {
                    f3b04440444044404440444 = 71;
                    f4b0413041304130413 = m3b044404440444();
                }
            } catch (Exception e) {
                throw e;
            }
        } catch (Exception e2) {
            throw e2;
        }
    }
```

ARXAN ENSUREIT

```
public static int m3b044404440444() {
    return 61;
}

public void onCreate(android.os.Bundle r5) {
    /*
    r4 = this;
L_0x0000:
    r0 = 0;
    switch(r0) {
        case 0: goto L_0x0009;
        case 1: goto L_0x0000;
        default: goto L_0x0004;
    };
L_0x0004:
    r0 = 1;
    switch(r0) {
        case 0: goto L_0x0000;
        case 1: goto L_0x0009;
        default: goto L_0x0008;
    };
L_0x0008:
    goto L_0x0004;
L_0x0009:
    super.onCreate(r5);  Catch:{ Exception -> 0x0068 }
    hhhhhh.hwwwwh.m30b043E043E043E(r4);  Catch:{ Exception -> 0x0068 }
    r0 = 2130903040; // 0x7f030000 float:1.7412887E38 double:1.0528059867E-314;
```

ARXAN ENSUREIT

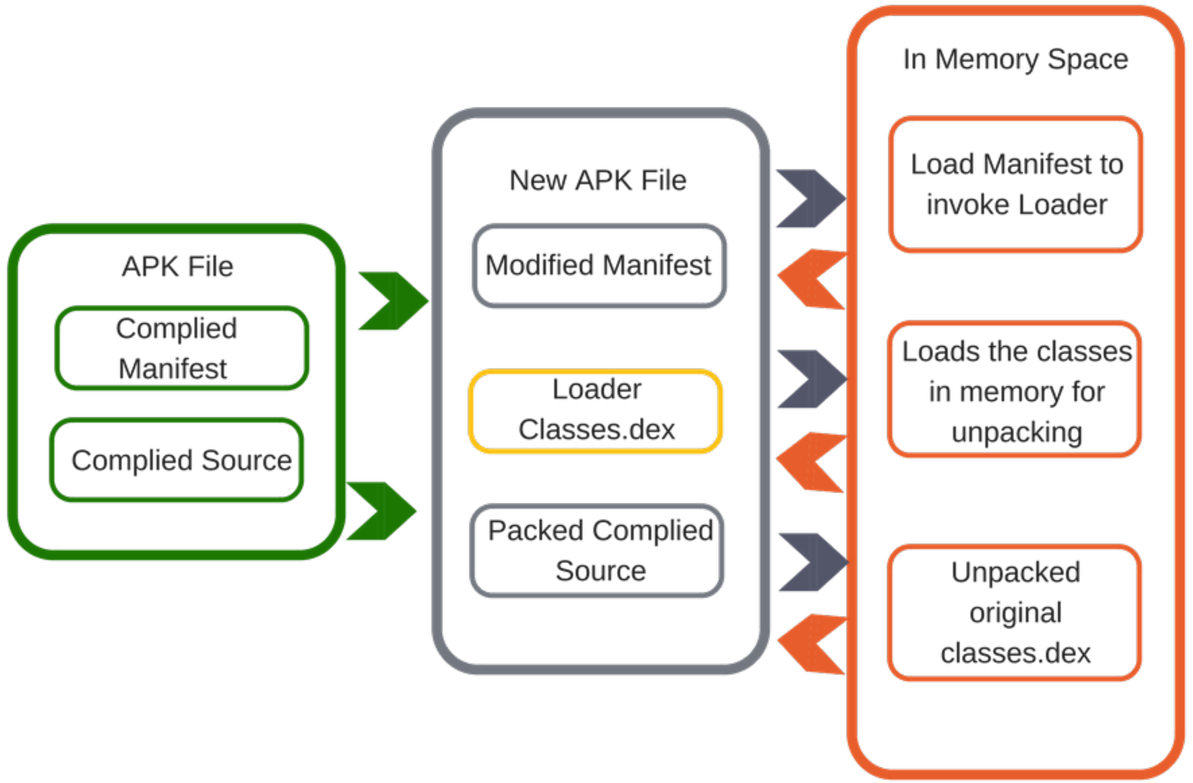
```
public static java.lang.String m38b043E(java.lang.String r3, char r4, char r5) {
    /*
    r2 = 0;
    r0 = f64b04440444;   Catch:{ Exception -> 0x0067 }
    if (r0 == 0) goto L_0x000b;
L_0x0005:
    m35b043E();  Catch:{ Exception -> 0x0067 }
    r0 = 0;
    f64b04440444 = r0;   Catch:{ Exception -> 0x0067 }
L_0x000b:
    r0 = f70b;   Catch:{ Exception -> 0x005c }
    r0 = r0.get(r5);     Catch:{ Exception -> 0x005c }
    r0 = (hhhhhh.hwhhhh) r0;     Catch:{ Exception -> 0x005c }
L_0x0013:
    r1 = 1;
    switch(r1) {
        case 0: goto L_0x0013;
        case 1: goto L_0x001b;
        default: goto L_0x0017;
    };
L_0x0017:
    switch(r2) {
        case 0: goto L_0x001b;
        case 1: goto L_0x0013;
        default: goto L_0x001a;
    };
L_0x001a:
    goto L_0x0017;
L_0x001b:
    r1 = f69b0444;
    r2 = f65b04440444;
    r1 = r1 + r2;
    r2 = f69b0444;
    r1 = r1 * r2;
```

PACKER

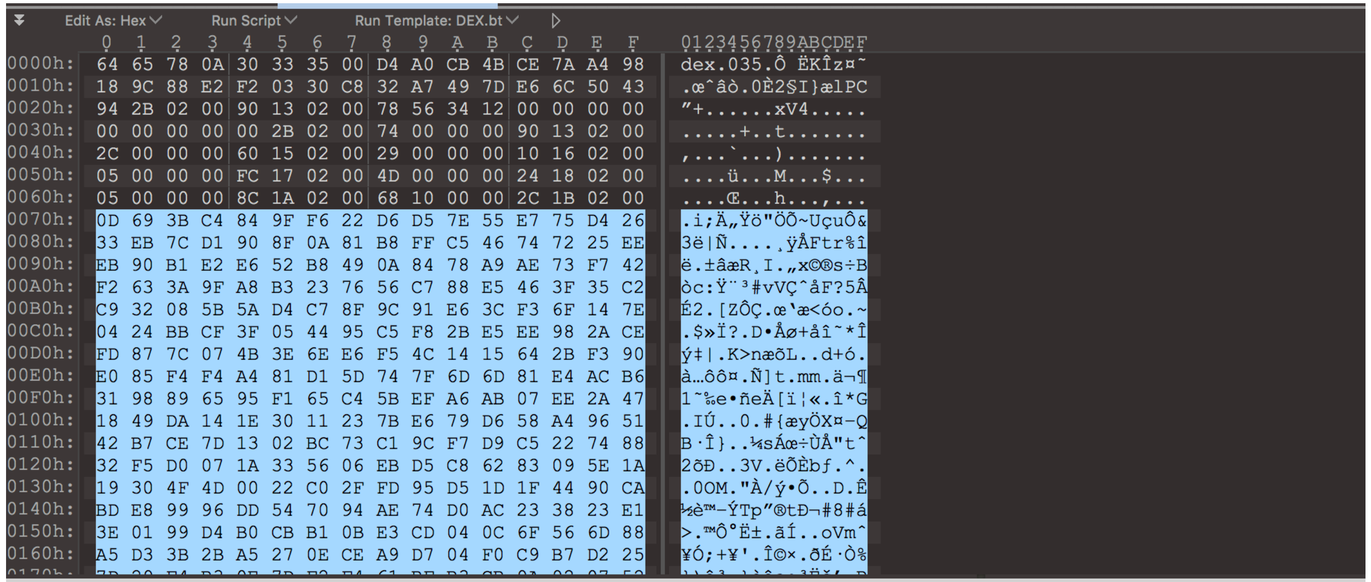
PACKERS

- Dynamic Code Modification
- Dynamic Loading
- In File Loading

PACKER – UNDER THE HOOD



'JMPJ2' PACKER (*POC PACKER)



SECONDHAND ? [PACKER]

```
<uses-permission android:name="en.cold.jewel.th.lines.permission.C2D_MESSAGE"/>
<application android:allowBackup="true" android:icon="@drawable/ic_launcher" android:label="@string/app_name" android:name="android.support.multidex.
MultiDexApplication" android:supportsRtl="true">
    <activity android:configChanges="keyboardHidden|orientation|screenSize" android:label="@string/app_name" android:name="com.neurondigital.JewelMiner.MainGame"
    android:windowSoftInputMode="stateHidden|adjustPan">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
```

```
package android.support.multidex;

import android.app.Application;
import android.content.Context;
import infiltration.outcry;

public class MultiDexApplication extends Application {
    protected void attachBaseContext(Context base) {
        super.attachBaseContext(base);
        outcry infiltration_outcry = new outcry(this, "NzZmBCp6IFcbCSQPb2NWa2ViBWNKekMGew==");
        MultiDex.install(this);
    }
}
```

SECONDHAND PACKER

outcry.java | MultiDexApplication.java

```
1   package infiltration;
2
3   import android.content.Context;
4
5   public class outcry {
6       public outcry(Context context, String str) {
7           try {
8               init(context, str);
9           } catch (Exception e) {
10              e.printStackTrace();
11          }
12      }
13
14      private native void init(Context context, String str);
15
16      static {
17          System.loadLibrary("secondhand");
18      }
19  }
20
```

VKEY

```
public class VGApplication extends Application {
    protected void attachBaseContext(Context context) {
        super.attachBaseContext(context);
        Util.loadNativeLib(this, "libelfhook.so");
        Util.loadNativeLib(this, "libwrapper.so");
        Util.loadDex(this);
        Util.bindApplication(this);
    }

    public void onCreate() {
        super.onCreate();
        Util.bindApplication(this);
        if (Util.app != null) {
            Util.app.onCreate();
        }
    }
}
```

VKEY

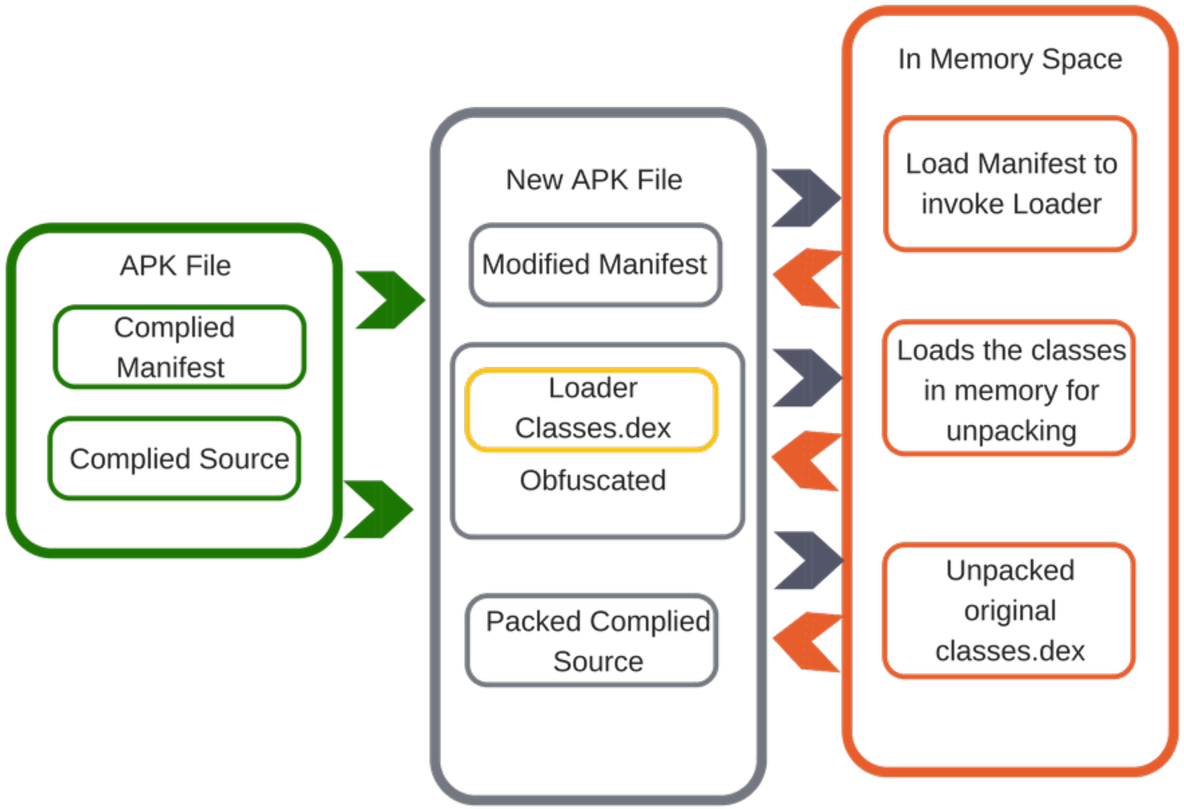
```
private static native void injectDexClassLoader(ClassLoader classLoader, AssetManager assetManager);

public static void loadDex(Context context) {
    try {
        ClassLoader classLoader = (ClassLoader) RefInvoke.getFieldOjbect(((WeakReference) ((Map) RefInvoke.getFieldOjbect(
        AssetManager assets = context.getResources().getAssets();
        startVOS(assets);
        injectDexClassLoader(classLoader, assets);
    } catch (Exception e) {
        Log.e(TAG, "loadDex: " + e.getMessage());
    }
}

public static void loadNativeLib(Context context, String str) {
    try {
        File file = new File(new StringBuilder(String.valueOf(context.getApplicationInfo().dataDir)).append("/files/").app
        if (file.exists()) {
            file.delete();
        }
        InputStream open = context.getAssets().open(str);
        OutputStream openFileOutput = context.openFileOutput(str, 0);
        if (open != null) {
            byte[] bArr = new byte[1024];
            while (true) {
                int read = open.read(bArr);
                if (read <= 0) {
                    break;
```

PROTECTOR

PROTECTORS

- Packer + Obfuscators
- Cat and Mouse game
- JNI Interface
- Dead Code Injection

PROTECTOR - UNDER THE HOOD



DEXPROTECTOR

```
13    <uses-permission android:name="android.location.GPS_ENABLED_CHANGE" />
14    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
15    <uses-permission android:name="android.permission.READ_SMS" />
16    <uses-permission android:name="android.permission.READ_CONTACTS" />
17    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
18    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
19    <application android:theme="@style/AppTheme" android:label="@string/p" android:icon="@mipmap/p" android:name="com.google.
      android.gms.common.api.Application" android:allowBackup="true">
20        <activity android:label="@string/p" android:name="com.dailyworkout.tizi.activities.ActivitySplash" android:
          screenOrientation="portrait" android:noHistory="true" />
21        <activity android:label="@string/p" android:name="com.dailyworkout.tizi.activities.ActivityHome" android:launchMode="
          singleTop" android:screenOrientation="portrait">
22            <intent-filter>
23                <action android:name="android.intent.action.MAIN" />
24                <category android:name="android.intent.category.LAUNCHER" />
25            </intent-filter>
```

DEXPROTECTOR

```
    }
    public void onCreate() {
        //Before the first component of the application starts
        super.onCreate()
        final Method declareMethod = Class.forname("dexprotector.name").getDeclaredMethod("run",hash);
        declaredMethod.setAccessible(true);
        declaredMethod.invoke(this, "com.dailyworkout.tizi.activities.ActivitySplash")
```

```
    public void onCreate() {
        super.onCreate();
        final Method declaredMethod = Class.forName(onCreate("\ua5ac\uc28c\u6b09\u67c8\ub589\u1543\u6148\uaf99\u4643\ud5f8\u8d09\udbe4\uf4ae\u1090\uca52\ub6a3\uc47c
        declaredMethod.setAccessible(true);
        declaredMethod.invoke(this, onCreate("\u6805\u4fbd\uc65f\u132f\u8502\u6413\u1db1\u3da9\uedee\u4c79\ud2e3\u6bf3\u1e81\u0559\u6ba2\uee61"));
    }
}
```

DEMO

WHAT'S NEXT?

WHAT'S NEXT – FROM COMMUNITY BACK TO COMMUNITY

- Create Android Malware Genome
  - Samples
  - Build Malware Genome KillChain
- Static Analysis Engines
- Dynamic Scanning Engine
- Scan PlayStore

QUESTIONS?

THANK YOU
FOR ATTENDING