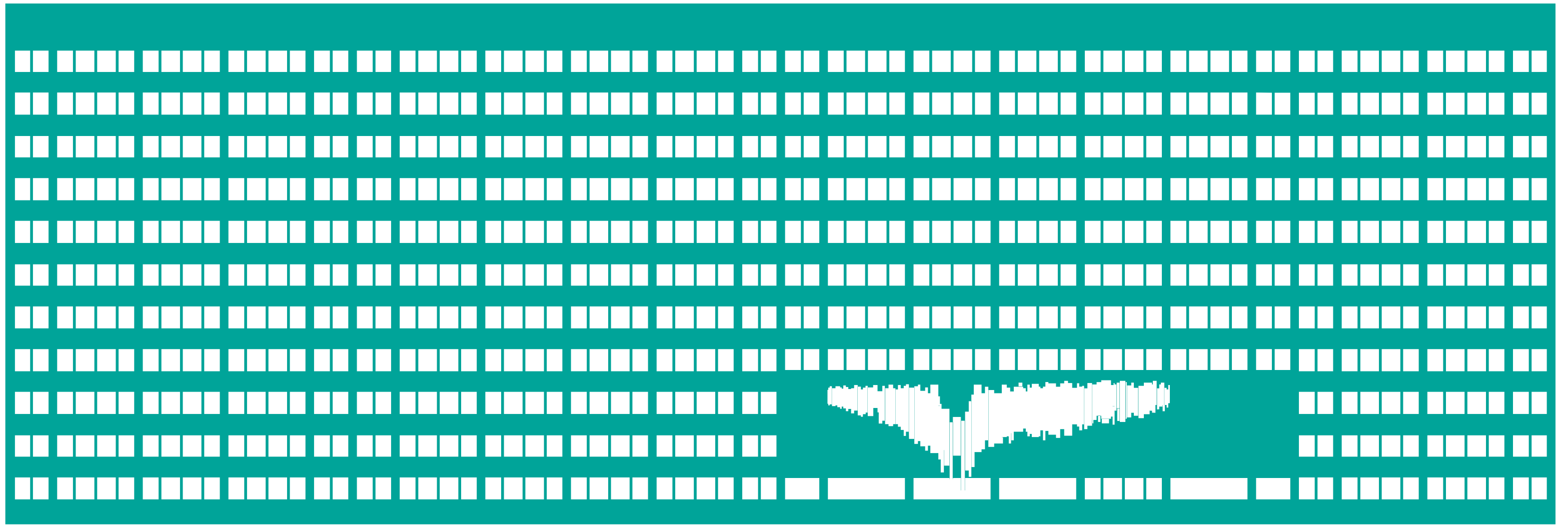
VŠB TECHNICKÁ UNIVERZITA OSTRAVA

VSB TECHNICAL UNIVERSITY OF OSTRAVA



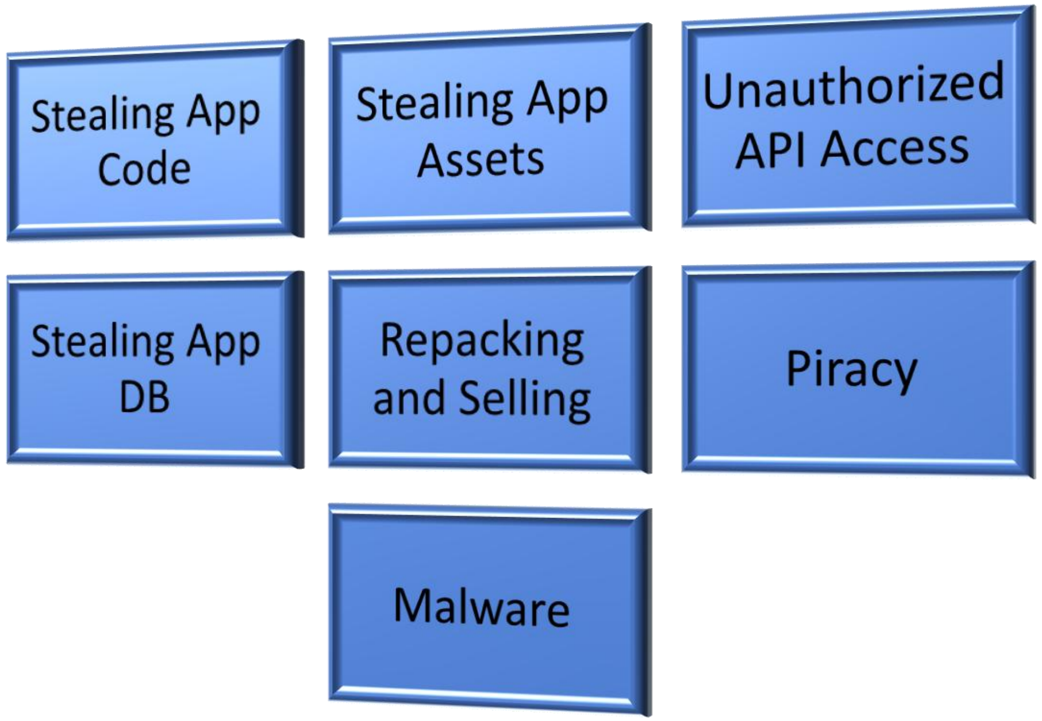
www.vsb.cz

Reverse Engineering Protection

Michal Krumnikl

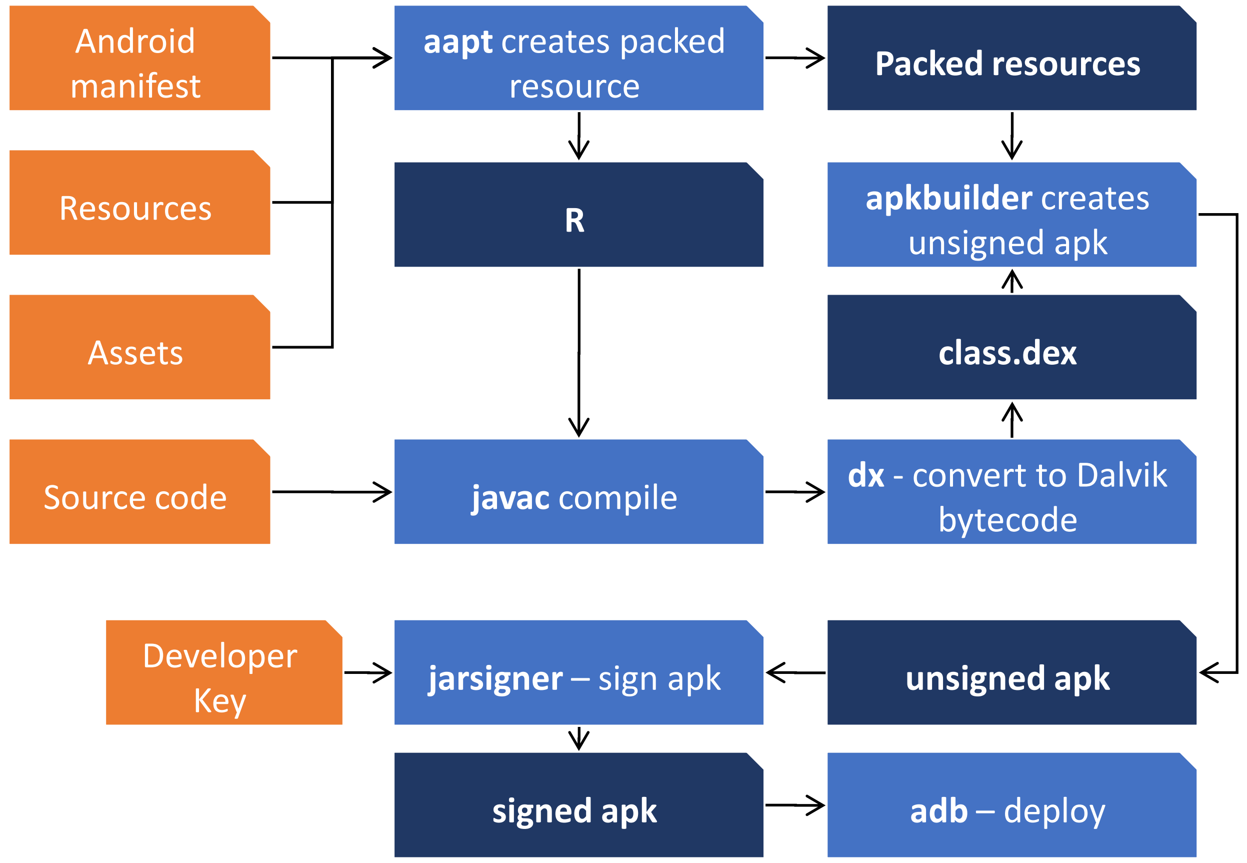
Current Threats

- **APK packages are easily downloadable**
- **APK to JAR conversion is easy**
- Java is partially compiled and then interpreted
    - **Few instructions**
    - **Opcodes are fixed**
- Main stream commercial packers, protectors and obfuscators
    - Most anti-decompilation/analysis tricks fixed in mainstream tools
    - baksmali, dex2jar, IDA Pro, radar



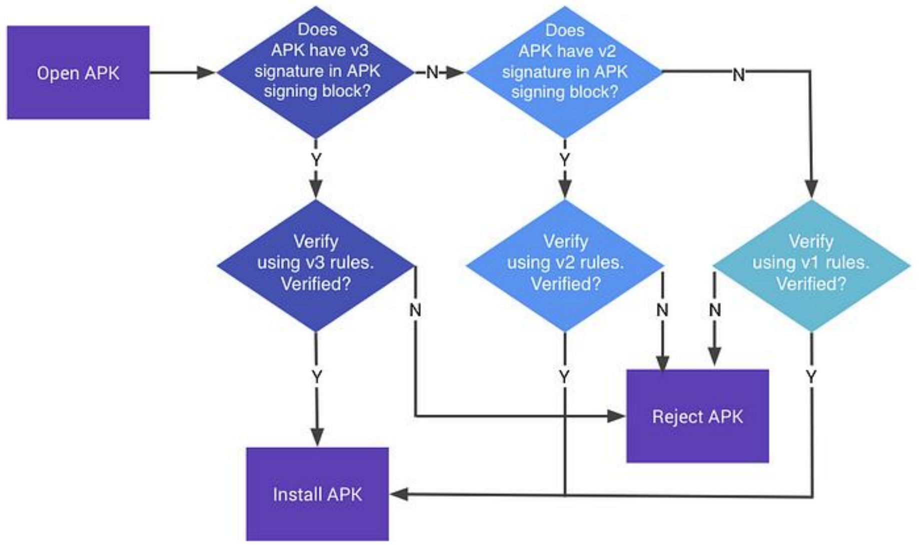
APK Creation vs. Disassembly

- **Disassembling** to smali
  - Similar to Jasmin syntax (Java assembler code)
  - **Apktool**
    - **Correct smali code**
- **Decompiling** to Java
  - **Dex2Jar + Java Decompiler**
    - **Sometimes incorrect Java code**



APK File Internals

- **Simple ZIP** file, renamed to “APK” extension
  - App resources (/res)
  - Signature (/META-INF)
  - Manifest (binary XML)
- **APK validation flow**

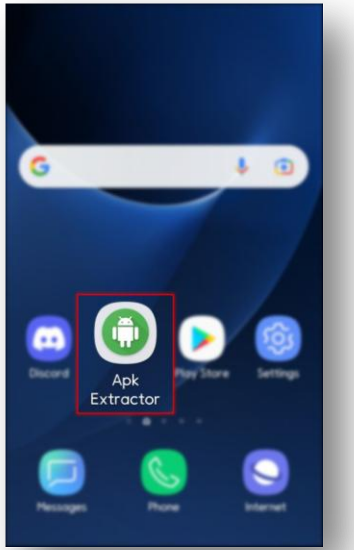

Open APK
Does APK have v3 signature in APK signing block?
N
Does APK have v2 signature in APK signing block?
N
Y
Y
Verify using v3 rules. Verified?
Verify using v2 rules. Verified?
Verify using v1 rules. Verified?
N
N
N
Y
Y
Y
Reject APK
Install APK

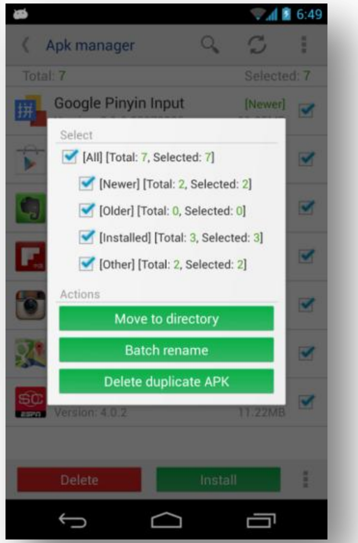
```
[kru13@localhost snake]$ unzip Snake2.zip
Archive:  Snake2.zip
 extracting: res/drawable/greenstar.png
 extracting: res/drawable/redstar.png
 extracting: res/drawable/yellowstar.png
  inflating: res/layout/snake_layout.xml
  inflating: AndroidManifest.xml
 extracting: resources.arsc
  inflating: classes.dex
  inflating: META-INF/MANIFEST.MF
  inflating: META-INF/CERT.SF
  inflating: META-INF/CERT.RSA
```

Source: https://medium.com/@vlad.iftimie88/some-points-on-android-apk-files-part-3-a562858dc92f

Getting APK from Phone

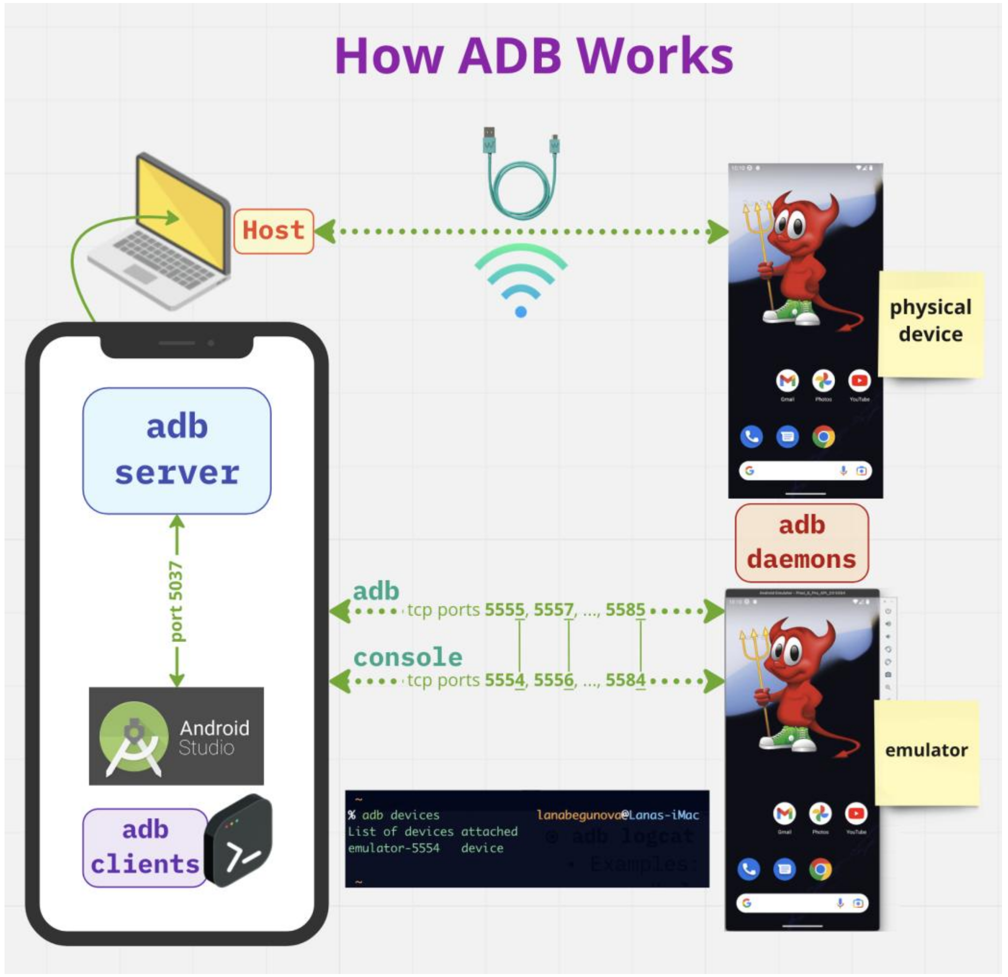
- **APKOptic**
- **Astro file manager**
- **APK File Manager**
- **APK Extractor**
- **APK Share**
- **…**




6:49
Apk manager
Total: 7
Selected: 7
Google Pinyin Input
[Newer]
Select
[All] [Total: 7, Selected: 7]
[Newer] [Total: 2, Selected: 2]
[Older] [Total: 0, Selected: 0]
[Installed] [Total: 3, Selected: 3]
[Other] [Total: 2, Selected: 2]
Actions
Move to directory
Batch rename
Delete duplicate APK
Version: 4.0.2
11.22MB
Delete
Install

Getting APK from Phone

- **Using ADB** (Android Debugging Bridge)
- Determine the package name of the app
  - `adb shell pm list packages`
- Get the full path name of the APK file
  - `adb shell pm path com.someapp`
- Pull the APK file from the Android device
  - `adb pull /data/app/com.someapp.apk`

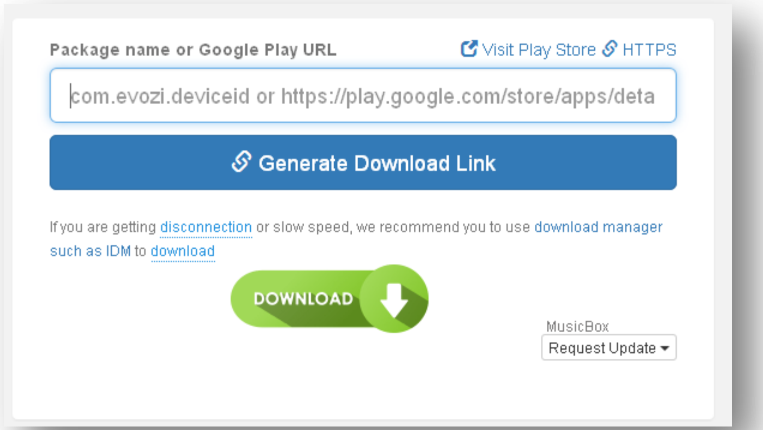


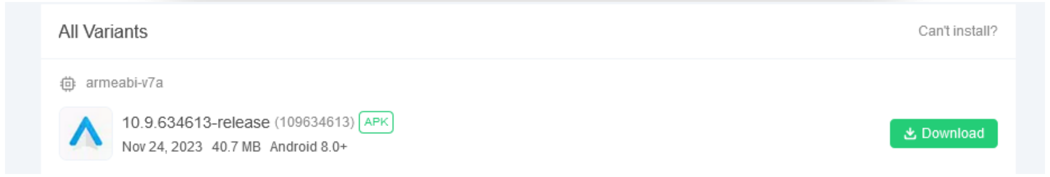
```
~
% adb devices                lanabegunova@Lanas-iMac
List of devices attached
emulator-5554   device

~
```

Source: https://github.com/lana-20/android-debug-bridge

Getting APK from Internet (Google Play)

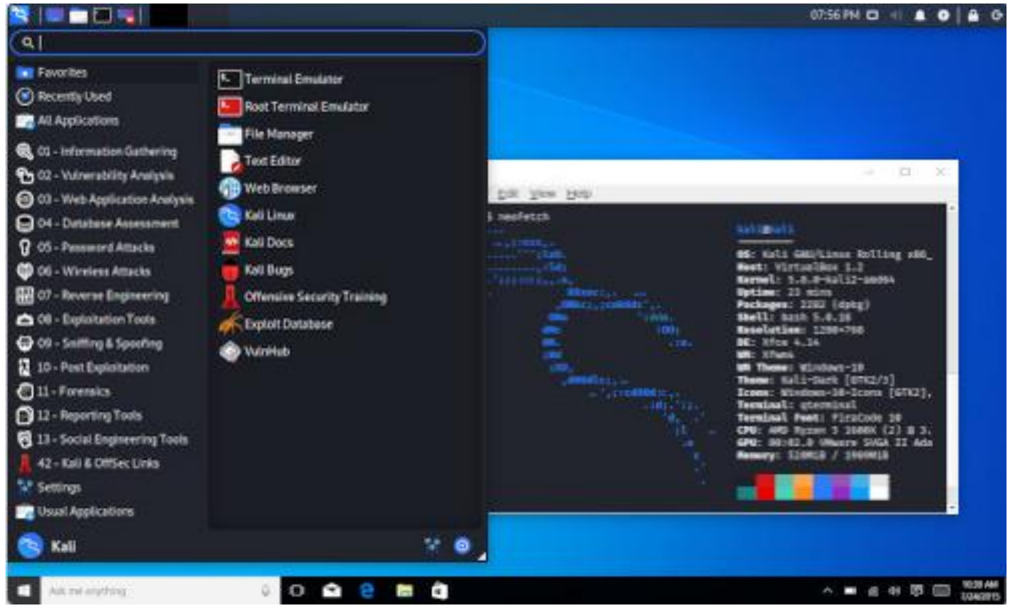
- **APKpure**
  - https://m.apkpure.com/
- Using unofficial Google Play API:
  - https://pypi.org/project/playstoreapi/
- Using a web service or browser extension:
  - http://apps.evozi.com/apk-downloader/

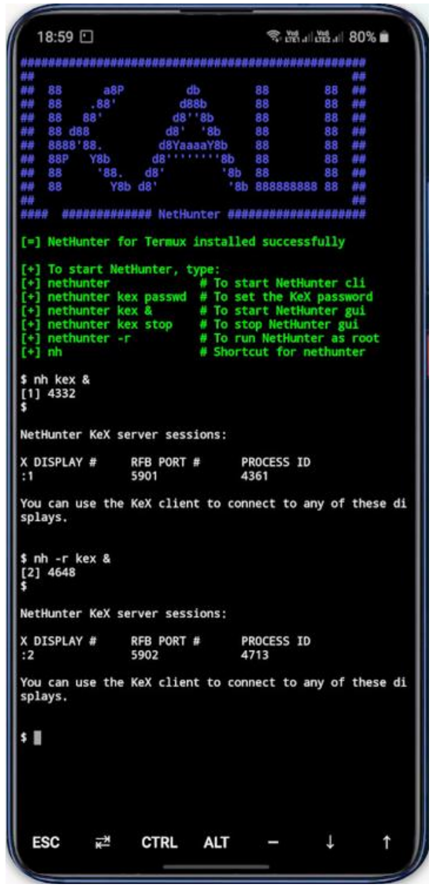

Package name or Google Play URL
Visit Play Store HTTPS
com.evozi.deviceid or https://play.google.com/store/apps/deta
Generate Download Link
If you are getting disconnection or slow speed, we recommend you to use download manager such as IDM to download


All Variants
Can't install?
armeabi-v7a
10.9.634613-release (109634613) APK
Nov 24, 2023 40.7 MB Android 8.0+

Code Analysis Tools

- Dexdump
- Dex2jar
- Smali
- APKTool
- IDA
- JD-GUI
- …





```
##################################################
##                                              ##
##  88      a8P         db        88        88  ##
##  88    .88'         d88b       88        88  ##
##  88   88'          d8''8b      88        88  ##
##  88 d88           d8'  '8b     88        88  ##
##  8888'88.        d8YaaaaY8b    88        88  ##
##  88P   Y8b      d8''''''''8b   88        88  ##
##  88     '88.   d8'        '8b  88        88  ##
##  88       Y8b d8'          '8b 888888888 88  ##
##                                              ##
####  ############## NetHunter ####################

[=] NetHunter for Termux installed successfully

[+] To start NetHunter, type:
[+] nethunter              # To start NetHunter cli
[+] nethunter kex passwd   # To set the KeX password
[+] nethunter kex &        # To start NetHunter gui
[+] nethunter kex stop     # To stop NetHunter gui
[+] nethunter -r           # To run NetHunter as root
[+] nh                     # Shortcut for nethunter

$ nh kex &
[1] 4332
$

NetHunter KeX server sessions:

X DISPLAY #     RFB PORT #     PROCESS ID
:1              5901           4361

You can use the KeX client to connect to any of these di
splays.


$ nh -r kex &
[2] 4648
$

NetHunter KeX server sessions:

X DISPLAY #     RFB PORT #     PROCESS ID
:2              5902           4713

You can use the KeX client to connect to any of these di
splays.


$ 
```

Dexdump

- **Included with the Android SDK**
- Basic dex file dissector
- **Disassemble Dalvik bytecode**
  - Uses linear sweep to find instructions

```
dexdump -d classes.dex

Class #0            -
  Class descriptor  : 'Lcom/example/android/snake/R$attr;'
  Access flags      : 0x0011 (PUBLIC FINAL)
  Superclass        : 'Ljava/lang/Object;'
  Interfaces        -
  Static fields     -
    #0              : (in Lcom/example/android/snake/R$attr;)
      name          : 'tileSize'
      type          : 'I'
      access        : 0x0019 (PUBLIC STATIC FINAL)
insns size    : 4 16-bit code units

001010:                                        |[001010] com.example.android.snake.R.attr.<init>:()V
001020: 7010 5e00 0000                         |0000: invoke-direct {v0}, Ljava/lang/Object;.<init>:()V // method@005e
001026: 0e00                                   |0003: return-void
```

smali / baksmali

- **smali** – assembler
- **baksmali** – disassembler
    - Uses recursive traversal approach
    - Better performance for obfuscated code
- **Used by other reverse engineering tools as a basic disassembler.**

```
# direct methods
.method static constructor <clinit>()V
    .locals 1

    .prologue
    .line 37
    const-string v0, "snake-view"

    sput-object v0, Lcom/example/android/snake/Snake;->ICICLE_KEY:Ljava/lang/String;

    .line 33
    return-void
.end method
```

Apktool

- Multi platform, Apache 2.0 license
- **Decode resources to original form (and rebuild after modification)**
- Transforms binary Dalvik bytecode (classes.dex) into Smali source

```
java -jar ../apktool_2.0.0rc2.jar d Snake2.apk

I: Using Apktool 2.0.0-RC2 on Snake2.apk
I: Loading resource table...
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/kru13/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

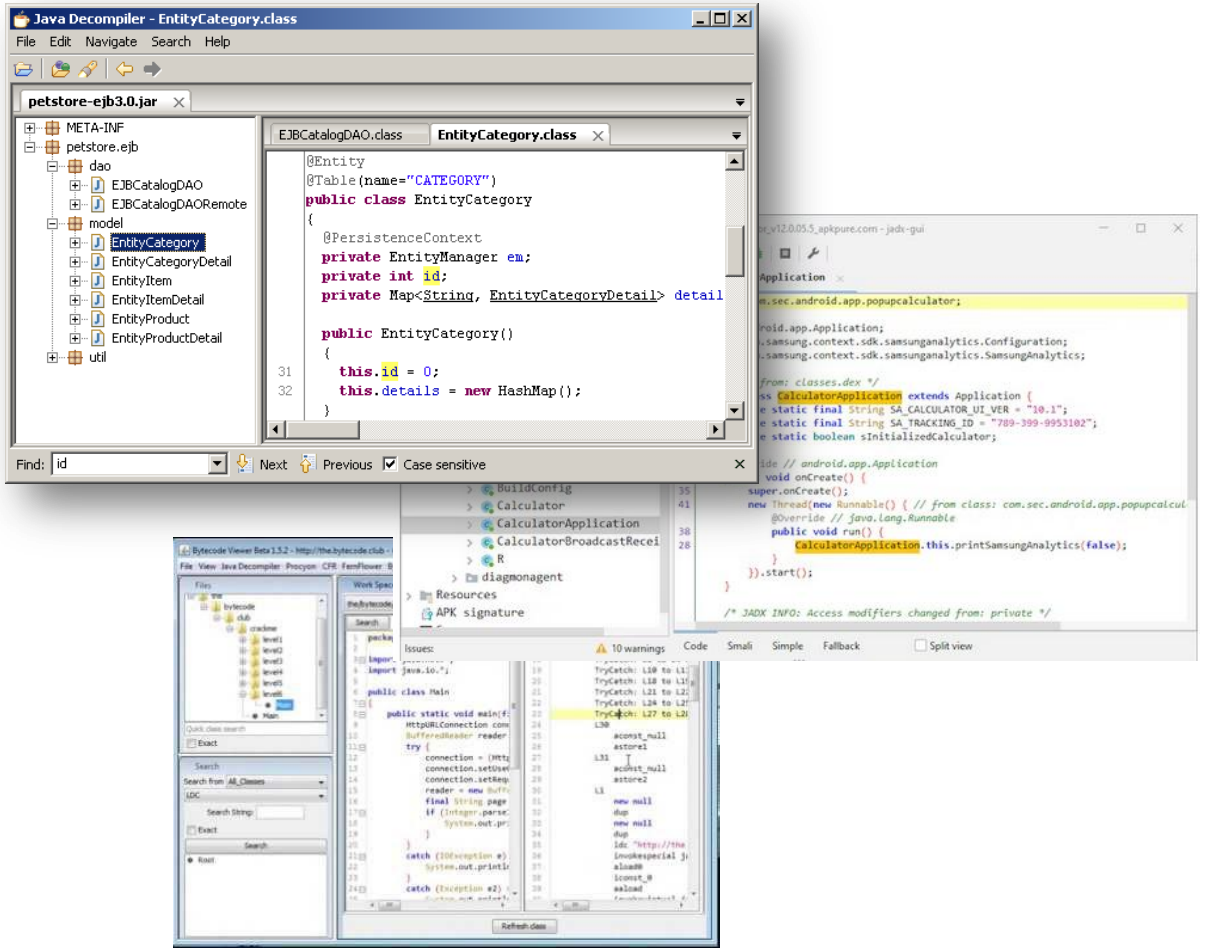
dex2jar

- Multi platform, Apache 2.0 license
- **Converts Dalvik bytecode (DEX) to java bytecode (JAR)**
- **Allows to use any existing Java decompiler with the resulting JAR file**
  - e.g. JD-GUI

```
../dex2jar-0.0.9.15/d d2j-dex2jar sh Snake2.apk
dex2jar version: translator-0.0.9.15
dex2jar Snake2.apk -> Snake2_dex2jar.jar
Done.
```

Decompilers

- **JD-GUI**
  - https://github.com/java-decompiler/jd-gui
  - Multi platform
- **JADX**
  - https://github.com/skylot/jadx
  - Multi platform, CLI
- **Bytecode Viewer**
  - https://github.com/Konloch/bytecode-viewer/
- **Dare, Mocha, Procyon**



```
@Entity
@Table(name="CATEGORY")
public class EntityCategory
{
  @PersistenceContext
  private EntityManager em;
  private int id;
  private Map<String, EntityCategoryDetail> detail

  public EntityCategory()
  {
    this.id = 0;
    this.details = new HashMap();
  }
```

```
m.sec.android.app.popupcalculator;

roid.app.Application;
.samsung.context.sdk.samsunganalytics.Configuration;
.samsung.context.sdk.samsunganalytics.SamsungAnalytics;

from: classes.dex */
ss CalculatorApplication extends Application {
e static final String SA_CALCULATOR_UI_VER = "10.1";
e static final String SA_TRACKING_ID = "789-399-9953102";
e static boolean sInitializedCalculator;

ide // android.app.Application
 void onCreate() {
    super.onCreate();
    new Thread(new Runnable() { // from class: com.sec.android.app.popupcalcul
        @Override // java.lang.Runnable
        public void run() {
            CalculatorApplication.this.printSamsungAnalytics(false);
        }
    }).start();
}

/* JADX INFO: Access modifiers changed from: private */
```

Resigning APK

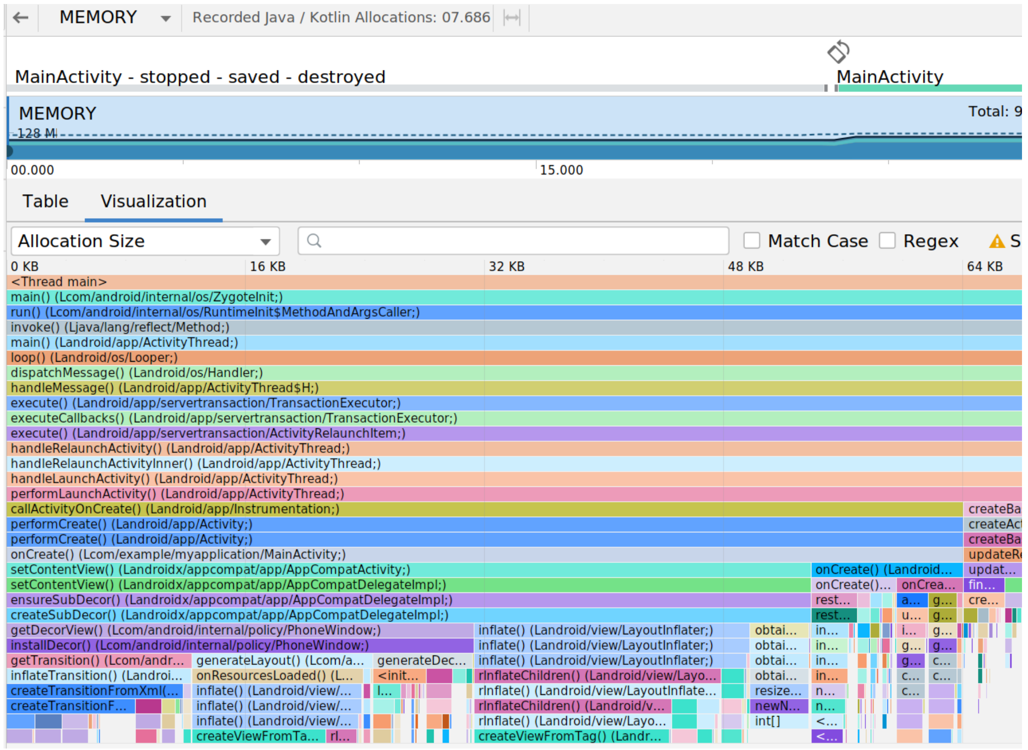
- JDK Keytool / JAR signer
  - Optinally **generate keystore**
    - keytool -genkey -alias someone -validity 100000 -keystore someone.keystore
  - **Sign** application
    - jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore my_application.apk alias_name
    - (jarsigner -keystore someone.keystore fake.apk someone)
  - **Verify** signature
    - jarsigner -verify -verbose -certs my_application.apk

- **Align** the ZIP archive in APK
  - zipalign -v 4 your_project_name-unaligned.apk your_project_name.apk
- **Install** application
  - adb install my_application.apk

- http://developer.android.com/tools/publishing/app-signing.html

Heap dump

- Snapshot of all the objects that are in memory in the JVM at a certain moment.
  - **adb shell am dumpheap <process> <file>**
- **hprof-conv** tool converts the HPROF file that is generated by the Android SDK tools to a standard format
- View memory allocations in Android Studio
  - **Record Java / Kotlin allocations**


MEMORY
Recorded Java / Kotlin Allocations: 07.686
MainActivity - stopped - saved - destroyed
MainActivity
MEMORY
Total: 9
128 M
00.000
15.000
Table
Visualization
Allocation Size
Match Case
Regex
0 KB
16 KB
32 KB
48 KB
64 KB
<Thread main>
main() (Lcom/android/internal/os/ZygoteInit;)
run() (Lcom/android/internal/os/RuntimeInit$MethodAndArgsCaller;)
invoke() (Ljava/lang/reflect/Method;)
main() (Landroid/app/ActivityThread;)
loop() (Landroid/os/Looper;)
dispatchMessage() (Landroid/os/Handler;)
handleMessage() (Landroid/app/ActivityThread$H;)
execute() (Landroid/app/servertransaction/TransactionExecutor;)
executeCallbacks() (Landroid/app/servertransaction/TransactionExecutor;)
execute() (Landroid/app/servertransaction/ActivityRelaunchItem;)
handleRelaunchActivity() (Landroid/app/ActivityThread;)
handleRelaunchActivityInner() (Landroid/app/ActivityThread;)
handleLaunchActivity() (Landroid/app/ActivityThread;)
performLaunchActivity() (Landroid/app/ActivityThread;)
callActivityOnCreate() (Landroid/app/Instrumentation;)
performCreate() (Landroid/app/Activity;)
performCreate() (Landroid/app/Activity;)
onCreate() (Lcom/example/myapplication/MainActivity;)
setContentView() (Landroidx/appcompat/app/AppCompatActivity;)
setContentView() (Landroidx/appcompat/app/AppCompatDelegateImpl;)
ensureSubDecor() (Landroidx/appcompat/app/AppCompatDelegateImpl;)
createSubDecor() (Landroidx/appcompat/app/AppCompatDelegateImpl;)
getDecorView() (Lcom/android/internal/policy/PhoneWindow;)
installDecor() (Lcom/android/internal/policy/PhoneWindow;)
inflate() (Landroid/view/LayoutInflater;)
rInflateChildren() (Landroid/view/Layo...
rInflate() (Landroid/view/LayoutInflate...
createViewFromTag() (Landr...

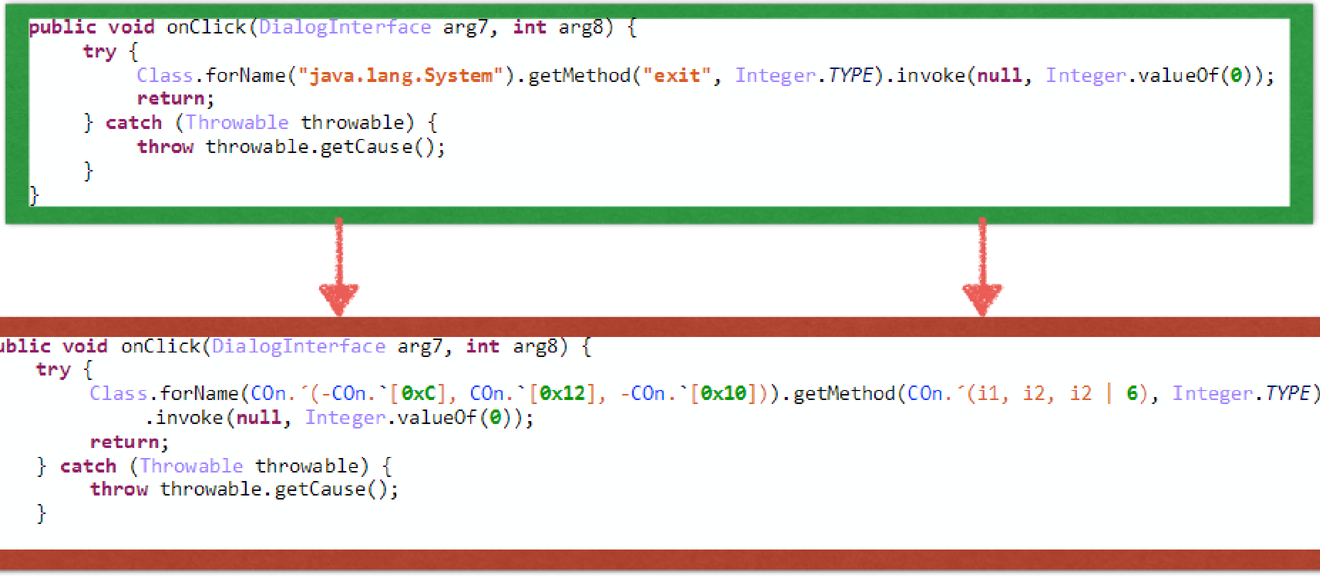
Code Protectors

- **Packers**
- **Optimizers / obfuscators**
- **Protectors**

- Software packages
  - **Proguard**
  - **Dexguard**
  - **Allatori**

Optimizers / Obfuscators

- Good practice for developers
- **Removes dead code / debug code**
- **Potentially encrypt / obfuscate / hide via reflection**



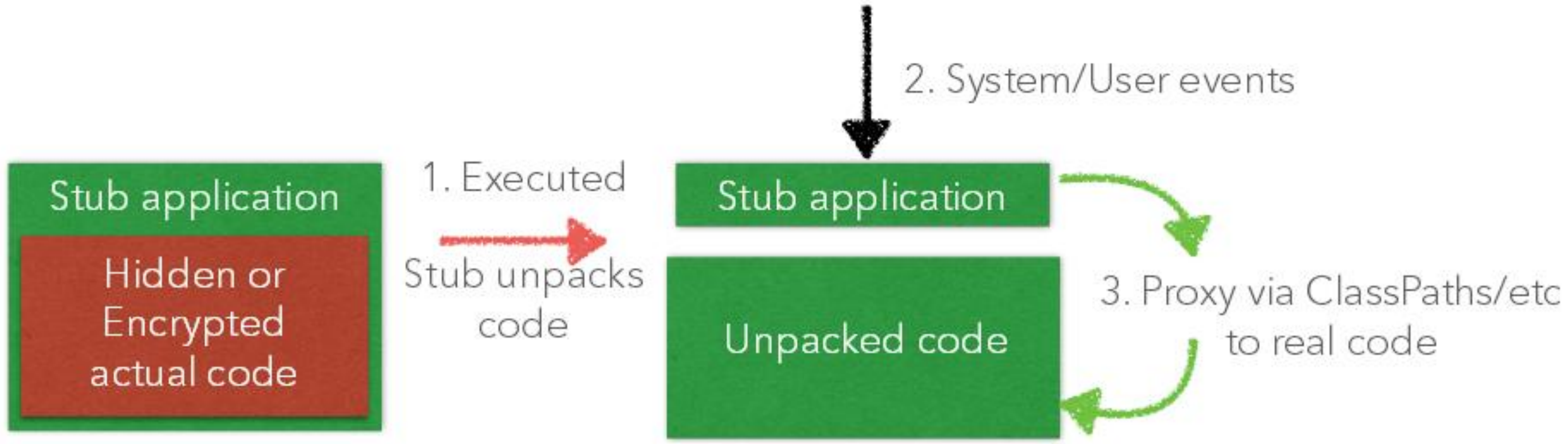
```java
public void onClick(DialogInterface arg7, int arg8) {
    try {
        Class.forName("java.lang.System").getMethod("exit", Integer.TYPE).invoke(null, Integer.valueOf(0));
        return;
    } catch (Throwable throwable) {
        throw throwable.getCause();
    }
}
```

```java
public void onClick(DialogInterface arg7, int arg8) {
    try {
        Class.forName(COn.´(-COn.`[0xC], COn.`[0x12], -COn.`[0x10])).getMethod(COn.´(i1, i2, i2 | 6), Integer.TYPE)
            .invoke(null, Integer.valueOf(0));
        return;
    } catch (Throwable throwable) {
        throw throwable.getCause();
    }
}
```

Source: Tim Strazzere, Jon Sawyer, Android Hacker Protection Level 0, Defcon 22, 2014

Packers

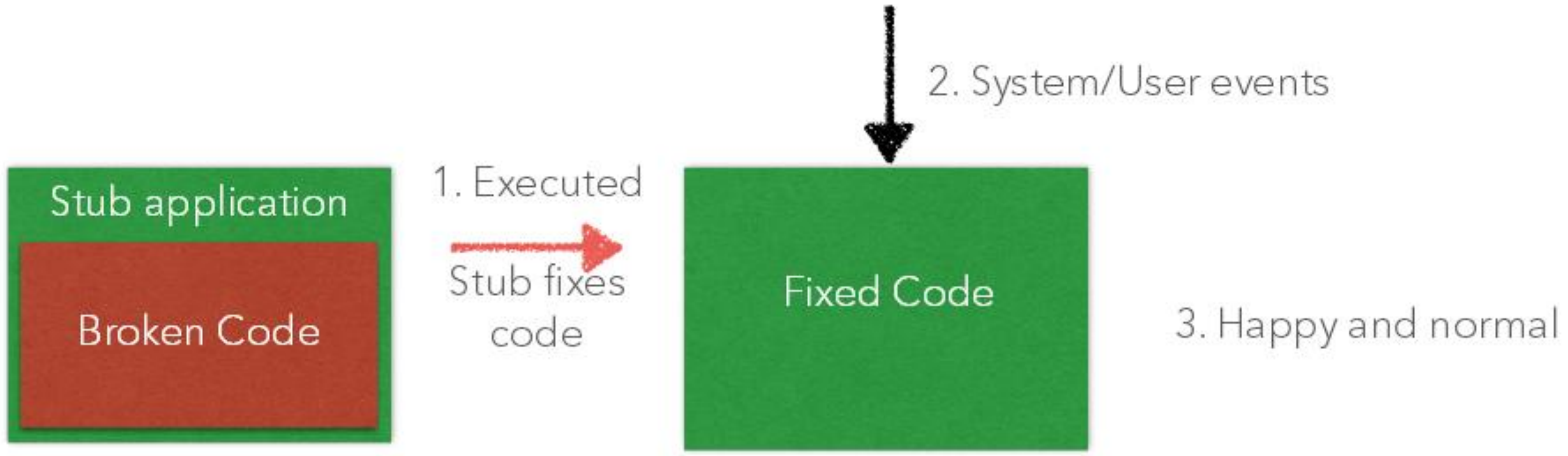
- Similar to UPX and others - **launcher stub and unfolding main application into memory**
- Performs anti-analysis/emulator tricks



Source: Tim Strazzere, Jon Sawyer, Android Hacker Protection Level 0, Defcon 22, 2014

Protectors

- Classification similar to packers - **manipulating "bad" code into workable things post execution**
- Performs anti-analysis/emulator tricks



Source: Tim Strazzere, Jon Sawyer, Android Hacker Protection Level 0, Defcon 22, 2014

Proguard

- 8 years older than Android
- Open source tool **integrated in the Android SDK.**
- Shrinks, optimizes, and obfuscates code
- Gradle configuration

```
buildTypes {
        release {
            minifyEnabled true
            shrinkResources true
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
            'proguard-rules.pro'
        }
    }
```

https://developer.android.com/build/shrink-code

- Methods
  - **Dead code elimination**
  - **Constant propagation**
  - **Method inlining**
  - **Class merging**
  - **Remove logging code**
  - **Peephole optimizations**
  - **Devirtualizations**
  - **…**

Proguard - Optimization

```
int answer = computeAnswer (1, 2, 3, 7);

int computeAnswer(int f1, int f2, int f3, int f4) {
           if (f2 == 1 && f3 == 1 && f4 == 1) {
                      return f1;
           } else {
                      return computeAnswer(f1 * f2, f3, f4, 1);
           }
}
```

Proguard - Optimization

```
int answer = computeAnswer (1, 2, 3, 7);

int computeAnswer(int f1, int f2, int f3, int f4) {
          if (f2 == 1 && f3 == 1 && f4 == 1) {
                    return f1;
          } else {
                    return computeAnswer(f1 * f2, f3, f4, 1);
          }
}
```

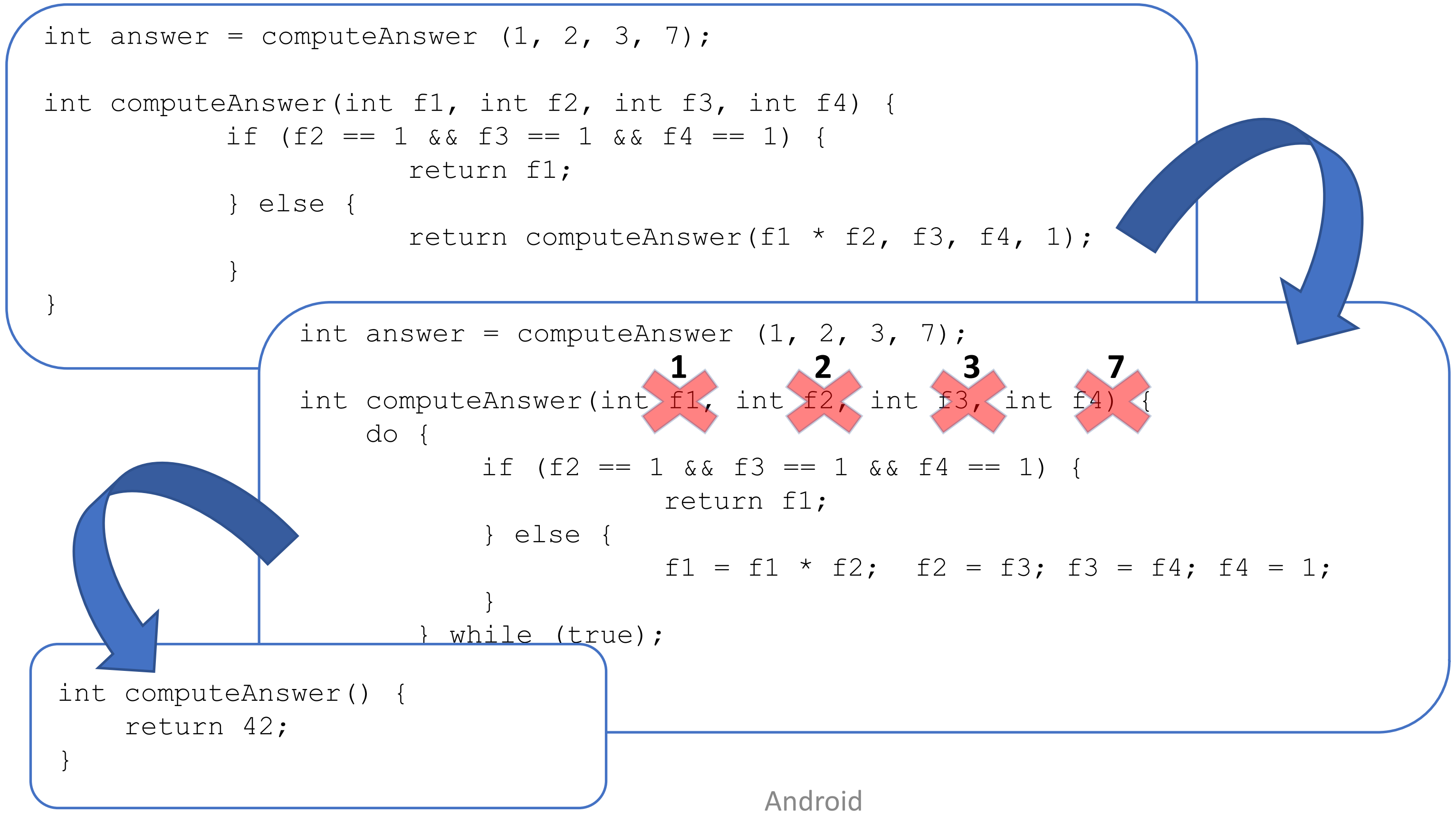
```
int answer = computeAnswer (1, 2, 3, 7);

int computeAnswer(int f1, int f2, int f3, int f4) {
    do {
          if (f2 == 1 && f3 == 1 && f4 == 1) {
                    return f1;
          } else {
                    f1 = f1 * f2;  f2 = f3; f3 = f4; f4 = 1;
          }
      } while (true);
}
```

Proguard - Optimization

```
int answer = computeAnswer (1, 2, 3, 7);

int computeAnswer(int f1, int f2, int f3, int f4) {
          if (f2 == 1 && f3 == 1 && f4 == 1) {
                    return f1;
          } else {
                    return computeAnswer(f1 * f2, f3, f4, 1);
          }
}
```

```
int answer = computeAnswer (1, 2, 3, 7);
                      1       2       3       7
int computeAnswer(int f1, int f2, int f3, int f4) {
    do {
          if (f2 == 1 && f3 == 1 && f4 == 1) {
                    return f1;
          } else {
                    f1 = f1 * f2;  f2 = f3; f3 = f4; f4 = 1;
          }
       } while (true);
}
```

Proguard - Optimization

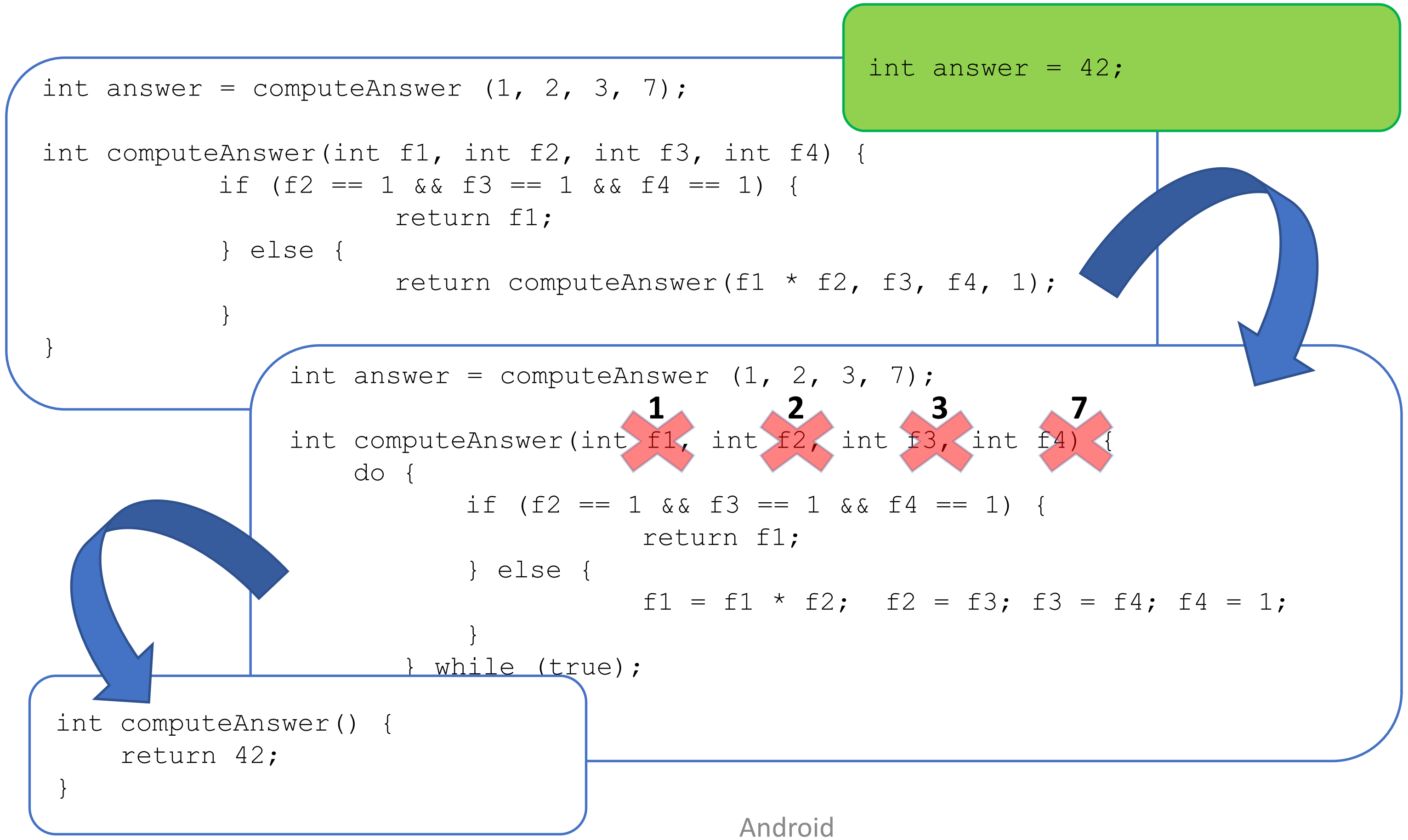


```
int answer = computeAnswer (1, 2, 3, 7);

int computeAnswer(int f1, int f2, int f3, int f4) {
            if (f2 == 1 && f3 == 1 && f4 == 1) {
                        return f1;
            } else {
                        return computeAnswer(f1 * f2, f3, f4, 1);
            }
}
```

```
int answer = computeAnswer (1, 2, 3, 7);

int computeAnswer(int ~~f1~~, int ~~f2~~, int ~~f3~~, int ~~f4~~) {
    do {
            if (f2 == 1 && f3 == 1 && f4 == 1) {
                        return f1;
            } else {
                        f1 = f1 * f2;  f2 = f3; f3 = f4; f4 = 1;
            }
        } while (true);
```

```
int computeAnswer() {
    return 42;
}
```

Proguard - Optimization



```
int answer = computeAnswer (1, 2, 3, 7);

int computeAnswer(int f1, int f2, int f3, int f4) {
           if (f2 == 1 && f3 == 1 && f4 == 1) {
                      return f1;
           } else {
                      return computeAnswer(f1 * f2, f3, f4, 1);
           }
}
```

```
int answer = computeAnswer (1, 2, 3, 7);

int computeAnswer(int f1, int f2, int f3, int f4) {
    do {
           if (f2 == 1 && f3 == 1 && f4 == 1) {
                      return f1;
           } else {
                      f1 = f1 * f2;  f2 = f3; f3 = f4; f4 = 1;
           }
       } while (true);
```

```
int computeAnswer() {
    return 42;
}
```

```
int answer = 42;
```

Proguard - Obfuscation

- Hiding program semantics through choosing semantically equivalent but complex and ambiguous representations in order to aggravate analysis.
- **Traditional approaches**
  - **Rename identifiers**
    - class / fields / method
  - **Remove debugging information**
    - Line numbers / local variable names / logcat / …

```
public class MyClass {
          private MySettings settings;
          private MyAlgorithm algorithm;
          private int answer;

          public int computeAnswer(int input) {
                    …
                    return answer;
          }
}
```

Proguard - Obfuscation

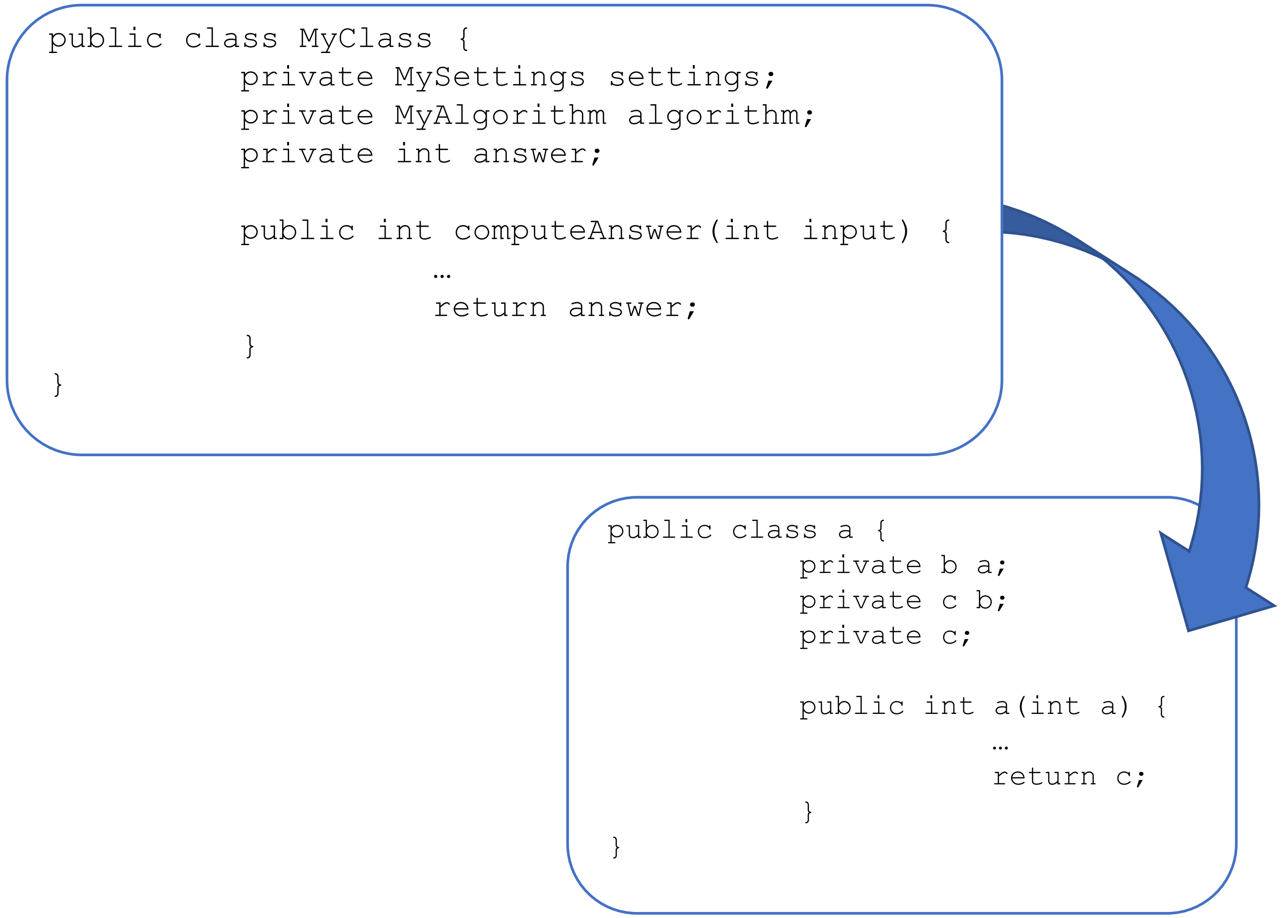
- Hiding program semantics through choosing semantically equivalent but complex and ambiguous representations in order to aggravate analysis.
- **Traditional approaches**
  - **Rename identifiers**
    - class / fields / method
  - **Remove debugging information**
    - Line numbers / local variable names / logcat / …

```
public class MyClass {
          private MySettings settings;
          private MyAlgorithm algorithm;
          private int answer;

          public int computeAnswer(int input) {
                    …
                    return answer;
          }
}
```

```
public class a {
          private b a;
          private c b;
          private c;

          public int a(int a) {
                    …
                    return c;
          }
}
```

Proguard - Advantages

- **Decreases dex file size**
- **Increases app speed/performance**
- **Decreases memory usage**
- **Removes debug information**
  - slightly increase reversing complexity
- **Doesn’t do much obfuscation**



```
public class MyClass {
        private MySettings settings;
        private MyAlgorithm algorithm;
        private int answer;

        public int computeAnswer(int input) {
                …
                return answer;
        }
}
```

```
public class a {
        private b a;
        private c b;
        private c;

        public int a(int a) {
                …
                return c;
        }
}
```

Allatori

- **Commercial product** from Smardec.
- Provides methods to modify the program code
  - Loop constructions are dissected in a way that reverse engineering tools cannot recognize them.
  - Strings are obfuscated and decoded at runtime.
- **Watermarker**
- The obfuscation methods used in Allatori are a superset of ProGuards.
- Cost: $290
- Free Academic Version

Allatori – Flow Obfuscation

```
 /**
     * Returns sum of the elements in the first rowsCount rows
     * and columnsCount columns.
     */
    int sumOfElements(int[][] matrix, int rowsCount, int columnsCount)
{
        int sum = 0;
        for (int row = 0; row < rowsCount; row++)
            for (int column = 0; column < columnsCount; column++)
                sum += matrix[row][column];
        return sum;
```

Allatori – Flow Obfuscation

```
  /**
     * Returns sum of the elements in the first rowsCount rows
     * and columnsCount columns.
     */
    int sumOfElements(int[][] matrix, int rowsCount, int columnsCount)
{
        int sum = 0;
        for (int row = 0; row < rowsCount; row++)
            for (int column = 0; column < columnsCount; column++)
                sum += matrix[row][column];
        return sum;
```

```
    int a(int a[][], int a, int a) {
           int i = 0;
           int j = 0;
             goto _L1
_L6:
           int k = 0;
             goto _L2
_L4:
           i += a[j][k];
           ++k;
_L2:
           a;
           JVM INSTR icmplt 17;
              goto _L3 _L4
_L3:
           ++j;
_L1:
           a;
           JVM INSTR icmplt 10;
              goto _L5 _L6
_L5:
           return i;
   }
```

Allatori – String Encryption

```
private void checkLicense() throws Exception {
        if (!isLicenseValid())
            throw new Exception("Invalid License.");
        else
            return;
    }
```

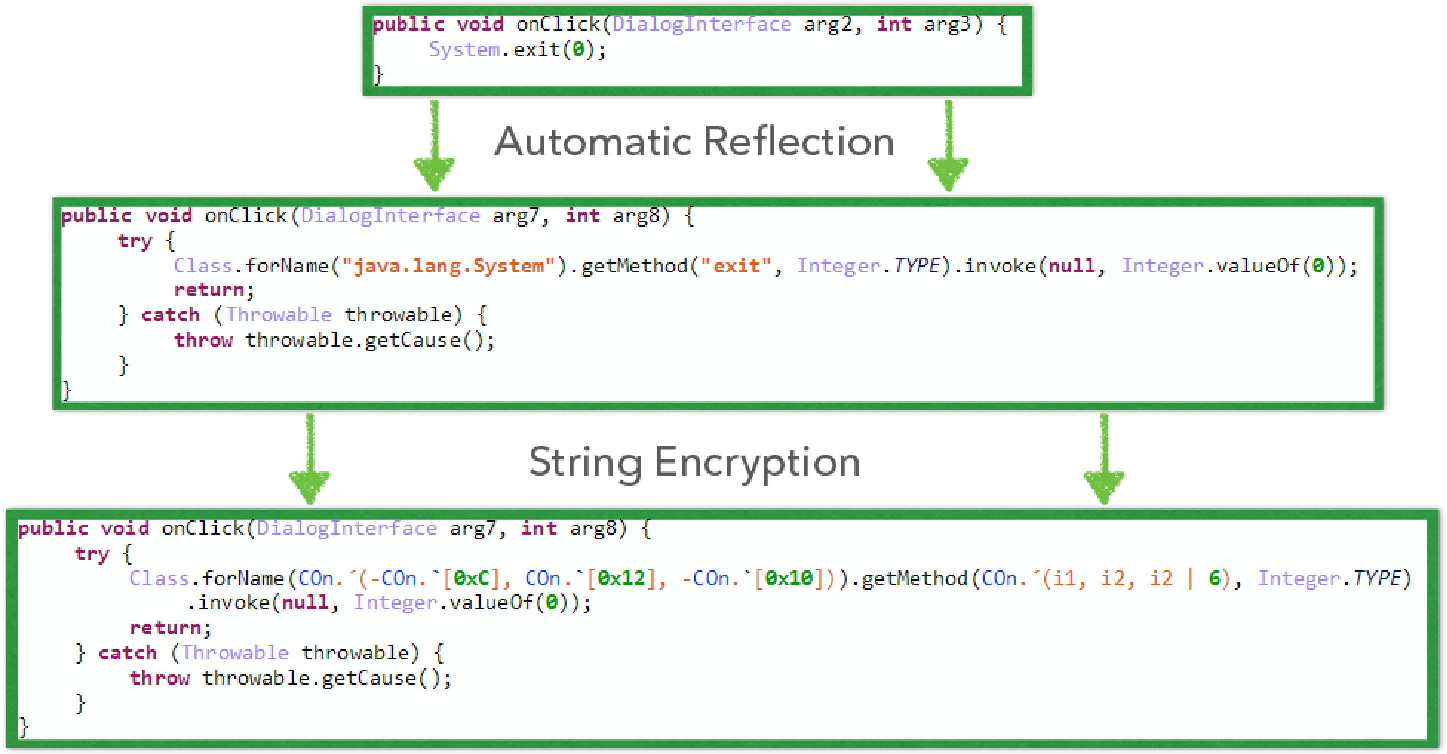
Allatori – String Encryption

```
private void checkLicense() throws Exception {
        if (!isLicenseValid())
            throw new Exception("Invalid License.");
        else
            return;
    }
```

```
 private void b() throws Exception {
        if(!a())
            throw new
Exception(a.a("\\z`t}}v5Q}}pwg\177{"));
        else
            return;
    }
```

Dexguard

- Commercial product from GuardSquare
  - Everything ProGuard does
  - **Automatic reflection**
  - Encrypt strings
  - Encrypt entire classes.
  - Obfuscate native code
  - Obfuscate resources
  - Encrypt assets
  - Add tamper detection
  - Add environment checks
- Cost: $650 - $1300



```
public void onClick(DialogInterface arg2, int arg3) {
    System.exit(0);
}
```

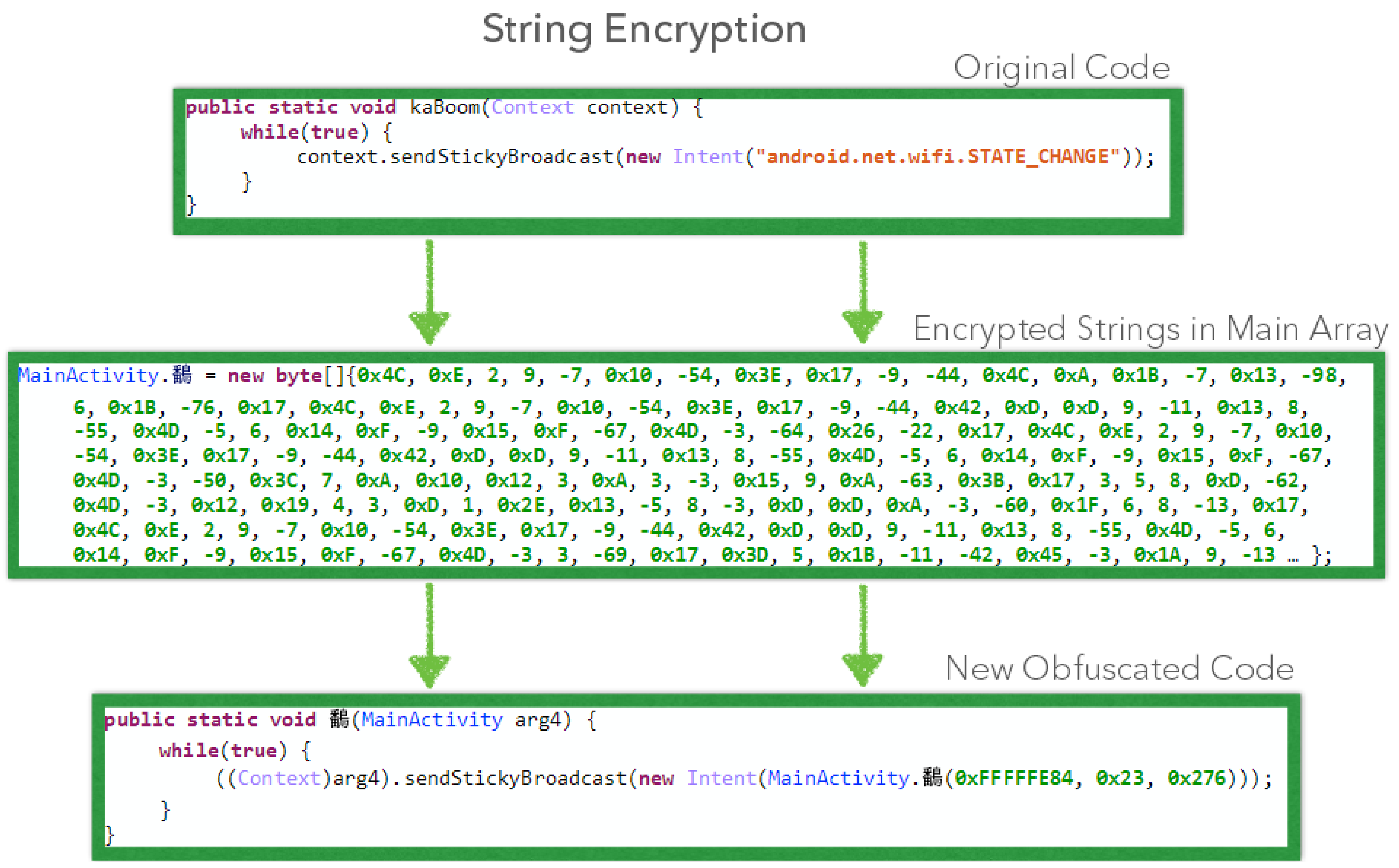
```
public void onClick(DialogInterface arg7, int arg8) {
    try {
        Class.forName("java.lang.System").getMethod("exit", Integer.TYPE).invoke(null, Integer.valueOf(0));
        return;
    } catch (Throwable throwable) {
        throw throwable.getCause();
    }
}
```

```
public void onClick(DialogInterface arg7, int arg8) {
    try {
        Class.forName(COn.´(-COn.`[0xC], COn.`[0x12], -COn.`[0x10])).getMethod(COn.´(i1, i2, i2 | 6), Integer.TYPE)
            .invoke(null, Integer.valueOf(0));
        return;
    } catch (Throwable throwable) {
        throw throwable.getCause();
    }
}
```

Source: Tim Strazzere, Jon Sawyer, Android Hacker Protection Level 0, Defcon 22, 2014

Dexguard

- Commercial product from GuardSquare
  - Everything ProGuard does
  - Automatic reflection
  - **Encrypt strings**
  - Encrypt entire classes.
  - Obfuscate native code
  - Obfuscate resources
  - Encrypt assets
  - Add tamper detection
  - Add environment checks
- Cost: $650 - $1300



```
public static void kaBoom(Context context) {
    while(true) {
        context.sendStickyBroadcast(new Intent("android.net.wifi.STATE_CHANGE"));
    }
}
```

```
MainActivity.鶕 = new byte[]{0x4C, 0xE, 2, 9, -7, 0x10, -54, 0x3E, 0x17, -9, -44, 0x4C, 0xA, 0x1B, -7, 0x13, -98,
    6, 0x1B, -76, 0x17, 0x4C, 0xE, 2, 9, -7, 0x10, -54, 0x3E, 0x17, -9, -44, 0x42, 0xD, 0xD, 9, -11, 0x13, 8,
    -55, 0x4D, -5, 6, 0x14, 0xF, -9, 0x15, 0xF, -67, 0x4D, -3, -64, 0x26, -22, 0x17, 0x4C, 0xE, 2, 9, -7, 0x10,
    -54, 0x3E, 0x17, -9, -44, 0x42, 0xD, 0xD, 9, -11, 0x13, 8, -55, 0x4D, -5, 6, 0x14, 0xF, -9, 0x15, 0xF, -67,
    0x4D, -3, -50, 0x3C, 7, 0xA, 0x10, 0x12, 3, 0xA, 3, -3, 0x15, 9, 0xA, -63, 0x3B, 0x17, 3, 5, 8, 0xD, -62,
    0x4D, -3, 0x12, 0x19, 4, 3, 0xD, 1, 0x2E, 0x13, -5, 8, -3, 0xD, 0xD, 0xA, -3, -60, 0x1F, 6, 8, -13, 0x17,
    0x4C, 0xE, 2, 9, -7, 0x10, -54, 0x3E, 0x17, -9, -44, 0x42, 0xD, 0xD, 9, -11, 0x13, 8, -55, 0x4D, -5, 6,
    0x14, 0xF, -9, 0x15, 0xF, -67, 0x4D, -3, 3, -69, 0x17, 0x3D, 5, 0x1B, -11, -42, 0x45, -3, 0x1A, 9, -13 … };
```

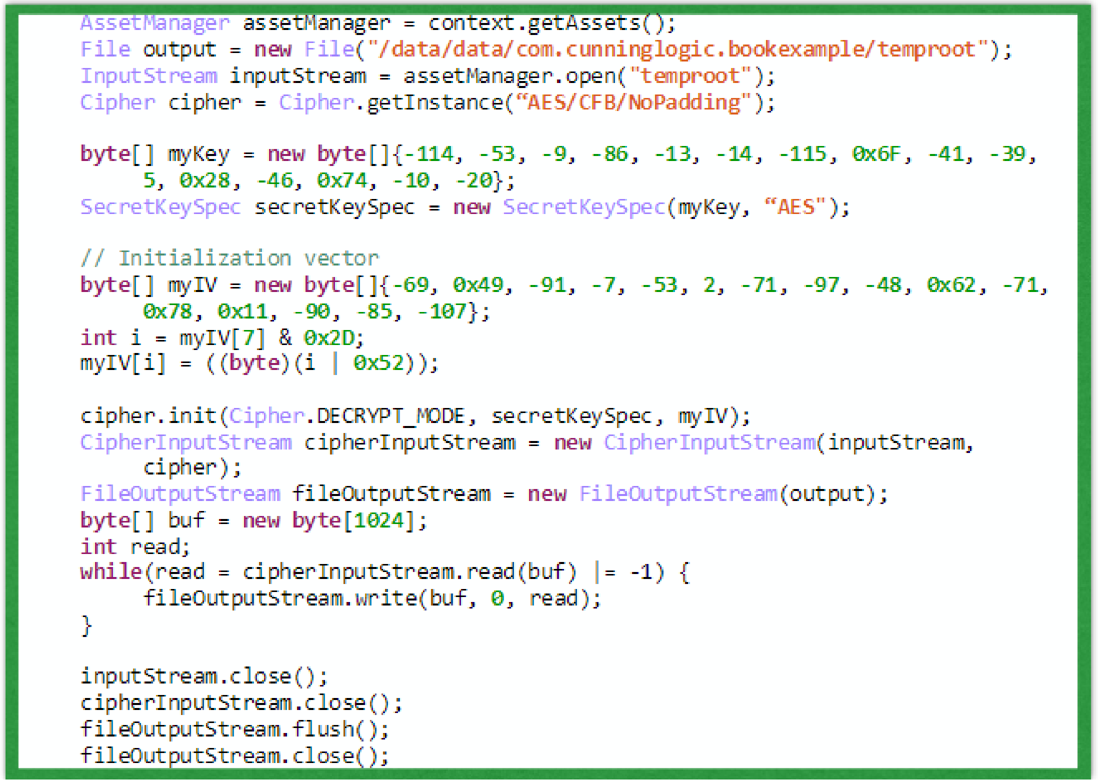
```
public static void 鶕(MainActivity arg4) {
    while(true) {
        ((Context)arg4).sendStickyBroadcast(new Intent(MainActivity.鶕(0xFFFFFE84, 0x23, 0x276)));
    }
}
```

Source: Tim Strazzere, Jon Sawyer, Android Hacker Protection Level 0, Defcon 22, 2014

Dexguard

- Commercial product from GuardSquare
  - Everything ProGuard does
  - Automatic reflection
  - Encrypt strings
  - Encrypt entire classes.
  - Obfuscate native code
  - Obfuscate resources
  - **Encrypt assets**
  - Add tamper detection
  - Add environment checks
- Cost: $650 - $1300

Asset & Library Encryption



```
AssetManager assetManager = context.getAssets();
File output = new File("/data/data/com.cunninglogic.bookexample/temproot");
InputStream inputStream = assetManager.open("temproot");
Cipher cipher = Cipher.getInstance("AES/CFB/NoPadding");

byte[] myKey = new byte[]{-114, -53, -9, -86, -13, -14, -115, 0x6F, -41, -39,
     5, 0x28, -46, 0x74, -10, -20};
SecretKeySpec secretKeySpec = new SecretKeySpec(myKey, "AES");

// Initialization vector
byte[] myIV = new byte[]{-69, 0x49, -91, -7, -53, 2, -71, -97, -48, 0x62, -71,
     0x78, 0x11, -90, -85, -107};
int i = myIV[7] & 0x2D;
myIV[i] = ((byte)(i | 0x52));

cipher.init(Cipher.DECRYPT_MODE, secretKeySpec, myIV);
CipherInputStream cipherInputStream = new CipherInputStream(inputStream,
     cipher);
FileOutputStream fileOutputStream = new FileOutputStream(output);
byte[] buf = new byte[1024];
int read;
while(read = cipherInputStream.read(buf) |= -1) {
     fileOutputStream.write(buf, 0, read);
}

inputStream.close();
cipherInputStream.close();
fileOutputStream.flush();
fileOutputStream.close();
```

Source: Tim Strazzere, Jon Sawyer, Android Hacker Protection Level 0, Defcon 22, 2014

Dexguard Features

- May increase dex file size
- May decrease app speed
- May increase memory usage
- Removes debug information
- Automatic string encryption
- Asset, Library, Class encryption
- Automatic reflection
- Moderately priced & easy to use
- Reversible with moderate effort

Source: Tim Strazzere, Jon Sawyer, Android Hacker Protection Level 0, Defcon 22, 2014

References

- https://www.defcon.org/images/defcon-22/dc-22-presentations/Strazzere-Sawyer/DEFCON-22-Strazzere-and-Sawyer-Android-Hacker-Protection-Level-UPDATED.pdf
- https://www.guardsquare.com/dexguard
- https://allatori.com/features.html

Thank you for your attention

Mgr. Ing. **Michal Krumnikl**, Ph.D.

+420 597 325 867

michal.krumnikl@vsb.cz

www.vsb.cz