DeStroid -
Fighting String Encryption in Android Malware

Fraunhofer Institute for Communication, Information Processing and Ergonomics

Daniel Baier **daniel.baier@fkie.fraunhofer.de**

Martin Lambertz martin.lambertz@fkie.fraunhofer.de

Fraunhofer FKIE

String Encryption
Introduction

- Android is an attractive target for attackers (especially malware developers)
  - Security analyst are interested in the functionalities and goals of Android malware
  - Malware developers are hiding their malicious behavior with different obfuscation techniques

http://pctechmag.com/wp-content/uploads/2012/08/threat android.jpg

String Encryption

Introduction

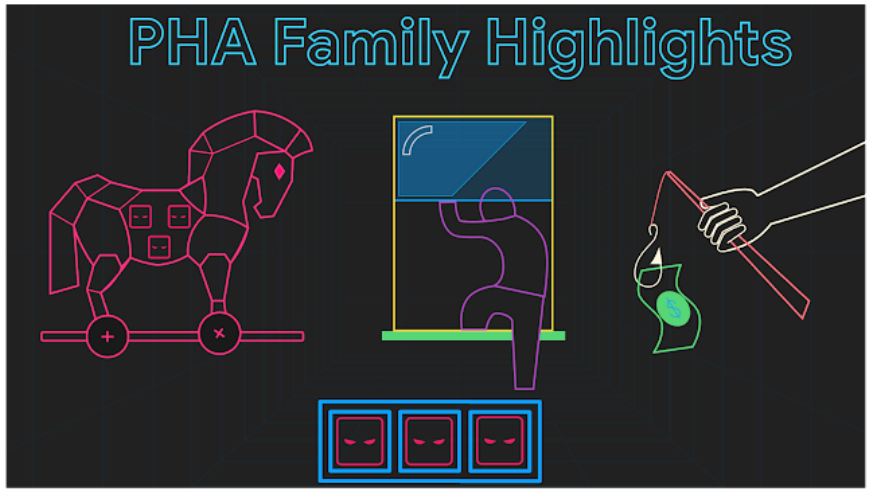
Google Security Blog

The latest news and insights from Google on security and safety on the Internet

PHA Family Highlights: Triada

June 6, 2019

Posted by Lukasz Siewierski, Android Security & Privacy Team



Fraunhofer FKIE

String Encryption

Introduction

Wirtschaft 〉 Digitec 〉 Malware befällt 25 Millionen Android-Smartphones

E-Paper ABONNEMENT

ANZEIGE

Testen Sie Ihr persönliches Anlageverhalten.

Frankfurter Allgemeine
Digitec

ANZEIGE

VOR:DENKER

ANDROID-GERÄTE BETROFFEN

Malware befällt 25 Millionen Mobiltelefone

VON BASTIAN BENRATH - AKTUALISIERT AM 10.07.2019 - 15:43

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

PHA Family Highlights: Triada

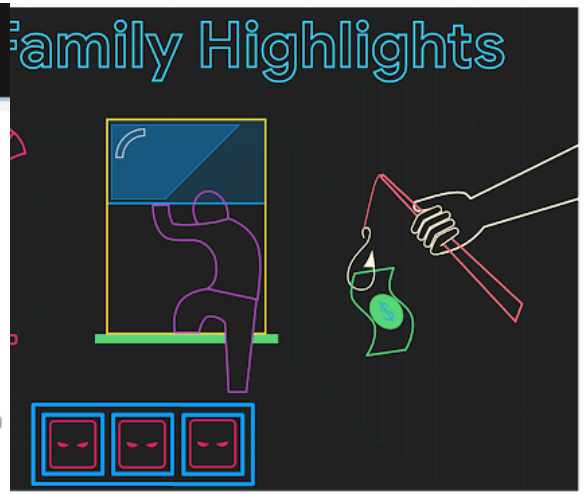
June 6, 2019

Posted by Lukasz Siewierski, Android Security & Privacy Team

Forbes Billionaires Innovation Leadership Money Business Small Business

EDITOR'S PICK | 278,103 views | Nov 2, 2019, 07:59am

Android Security Threat As 'Unremovable' Malware Infects 45,000 Phones So Far



Fraunhofer FKIE

String Encryption

Introduction

Wirtschaft 〉 Digitec 〉 Malware befällt 25 Millionen Android-Smartphones

ANZEIGE

Testen Sie Ihr persönliches Anlageverhalten.

Frankfurter Allgemeine

Digitec

E-Paper

VOR:D

ANDROID-GERÄTE BETROFFEN

Malware befällt 25 Millionen Mobiltelefone

VON BASTIAN BENRATH - AKTUALISIERT AM 10.07.2019 - 15:43

Forbes Billionaires Innovation Leadership

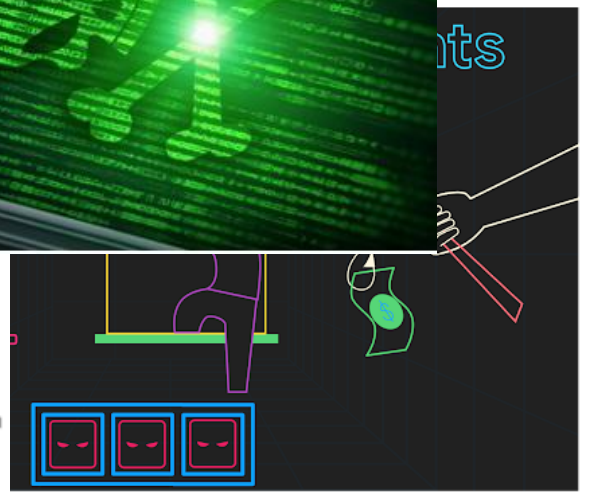
EDITOR'S PICK | 278,103 views | Nov

Android Security Threat As 'Unremovable' Malware Infects 45,000 Phones So Far

Android banking botnet targets thousands

By Anthony Spadafora October 03, 2019 Computing

Researchers discover new botnet that has already infected over 800,000 Android devices



Fraunhofer FKIE

String Encryption

Introduction

Wirtschaft 〉 Digitec 〉 Malware befällt 25 Millionen Android-Smart

ANZEIGE

Testen Sie Ihr persönliches Anlageverhalten.

Fran

ANDROID-GERÄTE BETROFFEN

Malware befällt Mobiltelefone

VON BASTIAN BENRATH - AKTUALISIE

Forbes

Billi

Warnung des BSI

Schadsoftware auf Tablets und Smartphones

Stand: 26.02.2019 13:40 Uhr

Schadsoftware auf neu gekauften Geräten - davor warnt das Bundesamt für Sicherheit in der Informationstechnik. Tausende Nutzer werden mutmaßlich von vorinstallierten Programmen ausgespäht.

Von Philipp Wundersee, WDR

"Der gesunde Menschenverstand ist durch nichts zu ersetzen", sagt Jörg Schwenk vom Lehrstuhl für Netz- und Datensicherheit der Universität Bochum. "Wenn das Gerät sehr günstig ist, wird der Hersteller ein verstecktes Geschäftsmodell eingepflegt haben. Da sollte man skeptisch sein."

Fachleute des Bundesamts für Sicherheit in der Informationstechnik hatten im Internet bei Amazon drei Billiggeräte bestellt und getestet. Es geht um das Tablet Eagle 804 von Krüger&Matz, das Smartphone S8 Pro von Ulefone und das Smartphone A10 von Blackview. [illegible] Tablet fanden sie sofort ein Schadprogramm. Bei den zwei chinesischen Sm[illegible] ältere Betriebssysteme betroffen, die man zur Nutzung des Handys über d[illegible] herunterladen muss.

"Wenn der Hersteller nicht zuverlässig ist, kann er beliebige Software auf s[illegible] Beim Kauf haben die Kunden erst einmal keinen Verdacht", sagt Schwenk. [illegible] Handy oder Tablet sei leider manchmal die Privatsphäre. Nicht immer würd[illegible] die Schadsoftware erkennen.

king botnet targets thousands

ober 03, 2019 Computing

ver new botnet that has already infected over evices

SOPHOS PRODUCTS SOLUTIONS

Overview Investo

Sophos **News**

The price of a cheap mobile phone may include your privacy

SophosLabs · SophosLabs Uncut · Android · Android malware · Pre-installed malware · Sophos Mobile · Supply chain

Inexpensive mobile phones may be subject to “supply chain compromise,” with Trojaned third party apps. We look at a phone that shipped with factory-installed malware

Fraunhofer FKIE

String Encryption

Motivation

What we want:

```
public static JSONObject m64b(Context context) {
 JSONObject jsonObject = new JSONObject();
 try {
  jsonObject.put("imei", C0012m.m68c(context));
  jsonObject.put("imsi", C0012m.m71d(context));
  jsonObject.put("uuid", C0012m.m49a());
  jsonObject.put("net", C0005f.m37b(context));
  jsonObject.put("channel", TextUtils.isEmpty(C0012m.m61b())? "channn" : C0012m.m61b());
  jsonObject.put("existPackages", C0012m.m76f(context));
 } catch (Exception e) {}
 return jsonObject;
}
```

Decompiled excerpt of a sample of the backdoor *Android.OS.Triada2016* malware

String Encryption

Motivation

What we get:

```
public static JSONObject m64b(Context context) {
 JSONObject jsonObject = new JSONObject();
 try {
  jsonObject.put(C0012m.m56a(C001b.f17I), C0012m.m68c(context));
  jsonObject.put(C0012m.m56a(C001b.f18J), C0012m.m71d(context));
  jsonObject.put(C0012m.m56a(C001b.f20L), C0012m.m49a());
  jsonObject.put(C0012m.m56a(C001b.f21M), C0005f.m37b(context));
  jsonObject.put(C0012m.m56a(C001b.f19K), TextUtils.isEmpty(C0012m.m61b()) ? "channn" …
  jsonObject.put(C0012m.m56a(C001b.f22N), C0012m.m76f(context));
 } catch (Exception e) {}
 return jsonObject;
}
```

Decompiled excerpt of a sample of the backdoor *Android.OS.Triada2016* malware

String Encryption

Motivation

What we get:

```
public static JSONObject m64b(Context context) {
 JSONObject jsonObject = new JSONObject();
 try {
  jsonObject.put(C0012m.m56a(C001b.f17
                                    (C001b.f183),
                                    (C001b.f20L), C0012m.m49a(
                                    (C001b.f21M), C0005f.m37b(co
                                    (C001b.f19K), TextUtils.isEmpty(C0012m.m61b()) ? "channn" …
                                    (C001b.f22N), C0012m.m76f(context));
```


```
static {
    f35a = new byte[]{C0001b.m47a(17)};
    f36b = new byte[]{(byte) 43, (byte) 115, (byte) 120, C0001b.m47a(92), C0001b.m47a(91), C0001b.m47a(81), (by
    f37c = new byte[]{(byte) 43, (byte) 115, (byte) 120, C0001b.m47a(92), C0001b.m47a(91), C0001b.m47a(81), (by
    f38d = new byte[]{(byte) 99, (byte) 111, (byte) 110, (byte) 102, (byte) 105, C0001b.m47a(81), (byte) 111,
    f39e = new byte[]{C0001b.m47a(81), (byte) 111, (byte) 111, C0001b.m47a(81), C0001b.m47a(86), (byte) 101, (
    f40f = new byte[]{(byte) 72, (byte) 68, (byte) 94, (byte) 77, (byte) 76, (byte) 64, (byte) 91, (byte) 84,
    f41g = new byte[]{(byte) 116, (byte) 109, C0001b.m47a(11), (byte) 116, (byte) 107, (byte) 107, C0001b.m47a
    f42h = new byte[]{(byte) 116, (byte) 109, C0001b.m47a(11), (byte) 104, (byte) 107, C0001b.m47a(91), (byte)
    f43i = new byte[]{C0001b.m47a(96), (byte) 109, C0001b.m47a(11), (byte) 44, (byte) 119, (byte) 34};
    f44j = new byte[]{(byte) 101, (byte) 109, C0001b.m47a(11), C0001b.m47a(92), C0001b.m47a(91), (byte) 99, (by
    f45k = new byte[]{(byte) 101, (byte) 109, C0001b.m47a(11), C0001b.m47a(92), C0001b.m47a(91), (byte) 99, (by
    f46l = new byte[]{C0001b.m47a(81), (byte) 111, (byte) 111, (byte) 99, C0001b.m47a(81), (byte) 34, (byte) 1
    f47m = new byte[]{(byte) 97, (byte) 120, C0001b.m47a(82), (byte) 117};
    f48n = new byte[]{C0001b.m47a(-8)};
    f49o = new byte[]{C0001b.m47a(81), (byte) 104, C0001b.m47a(86), (byte) 110, (byte) 97, (byte) 34, (byte) 5
    f50p = new byte[]{C0001b.m47a(96), (byte) 119, (byte) 101};
    f51q = new byte[]{(byte) 86, (byte) 97, (byte) 111, (byte) 102, C0001b.m47a(74)};
    f52r = new byte[]{(byte) 102, (byte) 121, (byte) 117, (byte) 100, C0001b.m47a(96), (byte) 63, (byte) 44, (
    f53s = new byte[]{C0001b.m47a(91), (byte) 116, C0001b.m47a(81), (byte) 44, (byte) 61};
    f54t = new byte[]{(byte) 105, (byte) 83, (byte) 100, (byte) 115, (byte) 115, (byte) 107, C0001b.m47a(84),
    f55u = new byte[]{(byte) 101, (byte) 110, (byte) 101, (byte) 115, C0001b.m47a(84), (byte) 107, (byte) 109,
    f56v = new byte[]{(byte) 119, (byte) 101, (byte) 117, (byte) 76, C0001b.m47a(84), (byte) 96, (byte) 96, (by
    f57w = new byte[]{(byte) 115, (byte) 105, C0001b.m47a(81), (byte) 104};
    f58x = new byte[]{(byte) 101, (byte) 110, (byte) 101, (byte) 115, C0001b.m47a(84), (byte) 107, (byte) 109,
    f59y = new byte[]{(byte) 101, (byte) 110, (byte) 101, (byte) 115, C0001b.m47a(84), (byte) 107, (byte) 109,
    f60z = new byte[]{(byte) 101, (byte) 110, (byte) 101, (byte) 115, C0001b.m47a(84), (byte) 107, (byte) 109,
```

String Encryption

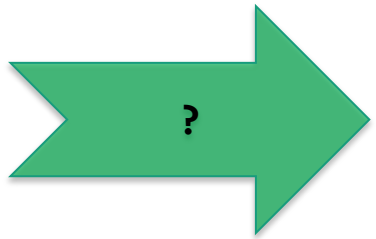
Motivation

```
public static JSONObject m64b(Context context) {
 JSONObject jsonObject = new JSONObject();
 try {
  jsonObject.put(C0012m.m56a(C001b.f17I),…;
  jsonObject.put(C0012m.m56a(C001b.f18J),…;
  jsonObject.put(C0012m.m56a(C001b.f20L),…);
  jsonObject.put(C0012m.m56a(C001b.f21M), …;
  jsonObject.put(C0012m.m56a(C001b.f19K), …;
  jsonObject.put(C0012m.m56a(C001b.f22N), …;
 } catch (Exception e) {}
 return jsonObject;
}
```

String Encryption

Motivation

```
public static JSONObject m64b(Context context) {
 JSONObject jsonObject = new JSONObject();
 try {
  jsonObject.put(C0012m.m56a(C001b.f17I),…;
  jsonObject.put(C0012m.m56a(C001b.f18J),…;
  jsonObject.put(C0012m.m56a(C001b.f20L),…);
  jsonObject.put(C0012m.m56a(C001b.f21M), …;
  jsonObject.put(C0012m.m56a(C001b.f19K), …;
  jsonObject.put(C0012m.m56a(C001b.f22N), …;
 } catch (Exception e) {}
 return jsonObject;
}
```



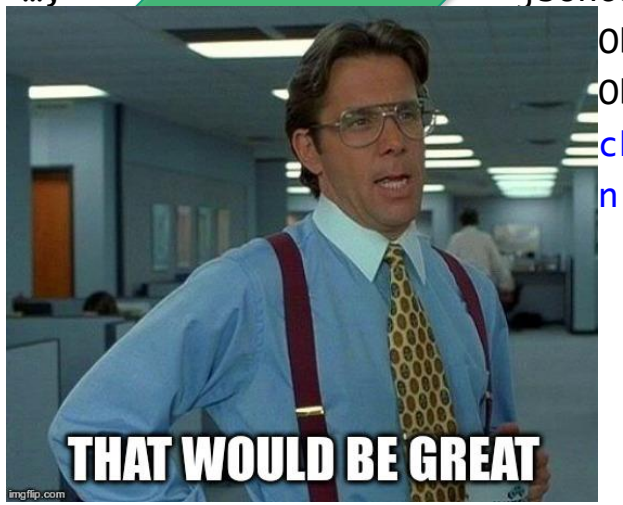

```
public static JSONObject m64b(Context context) {
 JSONObject jsonObject = new JSONObject();
 try {
  jsonObject.put("imei", …;
  jsonObject.put("imsi", …;
  jsonObject.put("uuid", …;
  jsonObject.put("net", …;
  jsonObject.put("channel", …;
  jsonObject.put("existPackages", …;
 } catch (Exception e) {}
 return jsonObject;
}
```

String Encryption

Motivation

```
public static JSONObject m64b(Context context) {
 JSONObject jsonObject = new JSONObject();
 try {
  jsonObject.put(C0012m.m56a(C001b.f17I),…;
  jsonObject.put(C0012m.m56a(C001b.f18J),…;
  jsonObject.put(C0012m.m56a(C001b.f20L),…);
  jsonObject.put(C0012m.m56a(C001b.f21M), …;
  jsonObject.put(C0012m.m56a(C001b.f19K),
  jsonObject.put(C0012m.m56a(C001b.f22N),
 } catch (Exception e) {}
 return jsonObject;
}
```

DeStroid




```
public static JSONObject m64b(Context context) {
 JSONObject jsonObject = new JSONObject();
 try {
  jsonObject.put("imei", …;
  jsonObject.put("imsi", …;
  jsonObject.put("uuid", …;
  jsonObject.put("net", …;
  Object.put("channel", …;
  Object.put("existPackages", …;
 ch (Exception e) {}
 n jsonObject;
```

String Encryption

Roadmap to DeStroid

- How?

String Encryption

Roadmap to DeStroid

- How?
  - ground truth?

String Encryption

Roadmap to DeStroid

- How?
  - ground truth?

malpedia

Fraunhofer FKIE

String Encryption

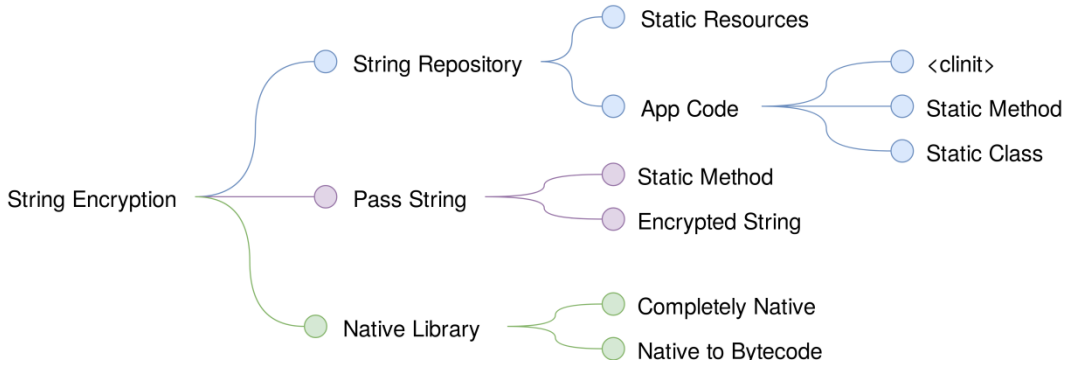
Roadmap to DeStroid

- How?
  - analyze every sample by hand

String Encryption

Roadmap to DeStroid

- How?
  - analyze every sample by hand



String Encryption

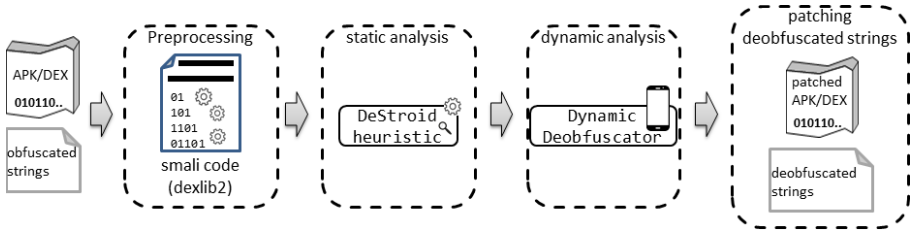
Roadmap to DeStroid

- How?
  - implement a tool to automatically decrypt the major String Encryption types

String Encryption

Roadmap to DeStroid

- How?
  - implement a tool to automatically decrypt the major String Encryption types

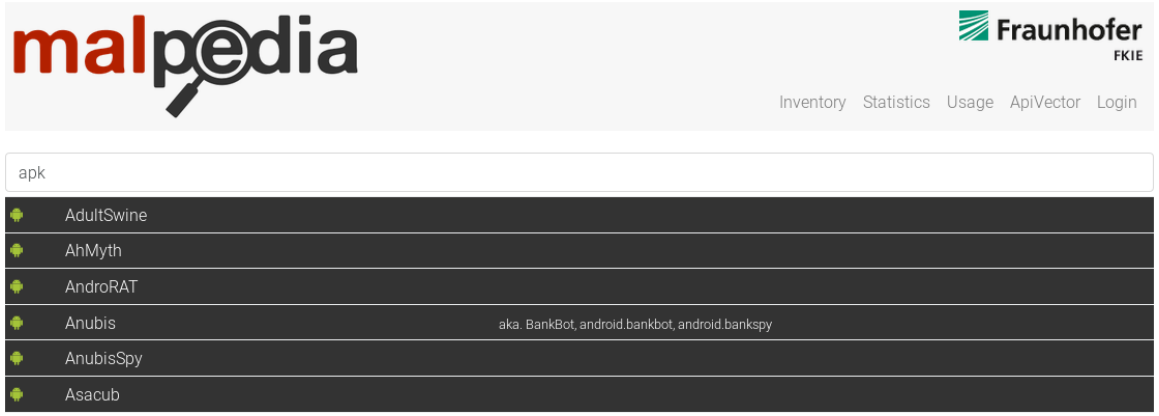


DeStroid

String Encryption
Dataset

- Malpedia APK dataset
  - Maintained and updated with new families as they occur
  - 96 Android malware families (as of October 2019)
  - 50/96 Android malware families using String Encryption

https://malpedia.caad.fkie.fraunhofer.de/

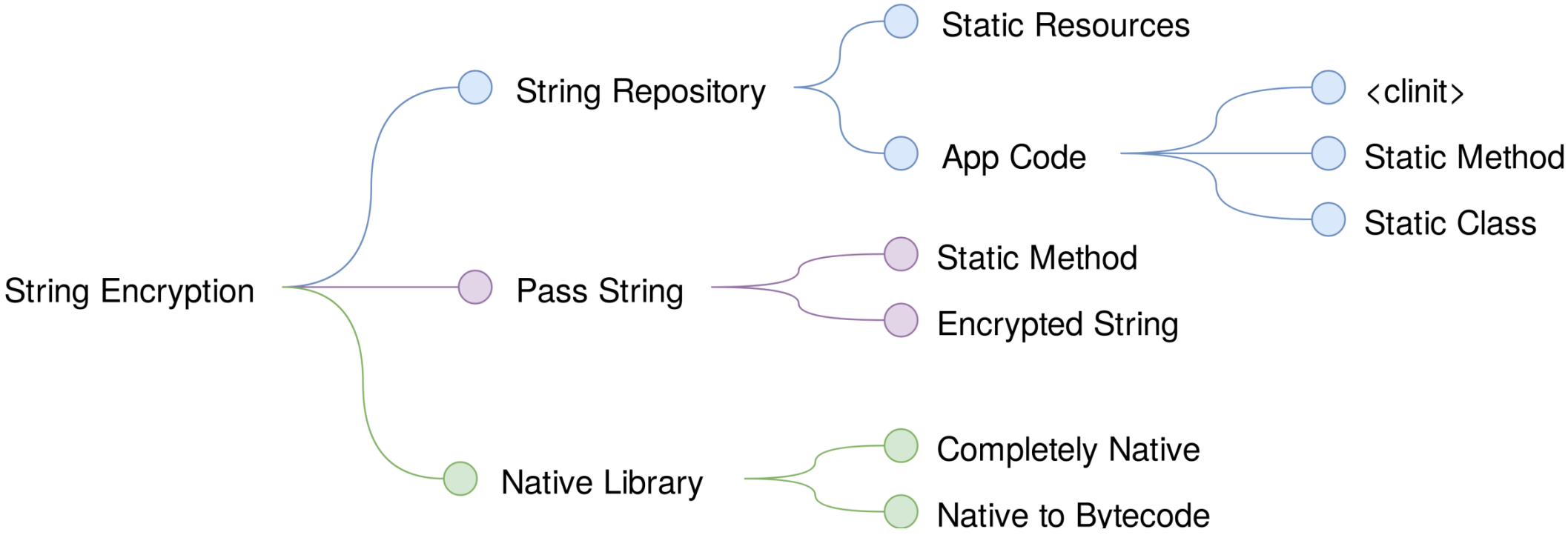


Want Access?

Talk to Daniel Plohmann here at

String Encryption

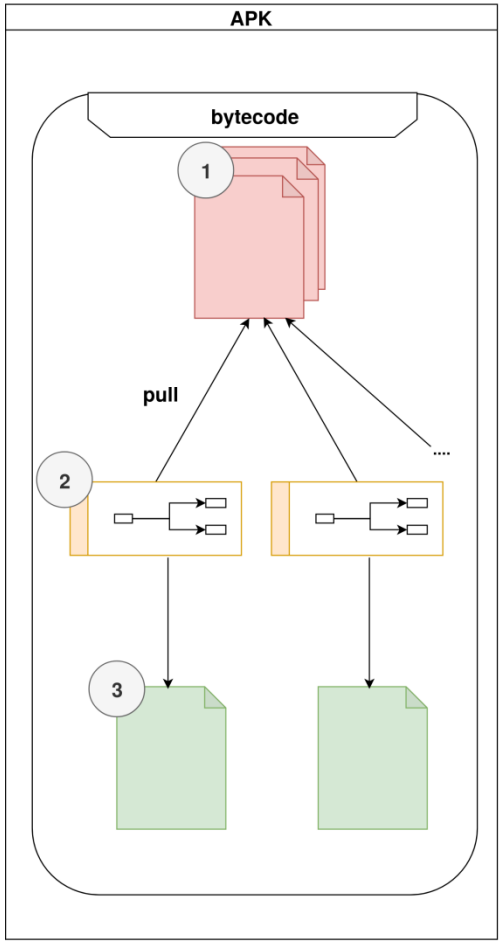
Taxonomy of String Encryption Techniques



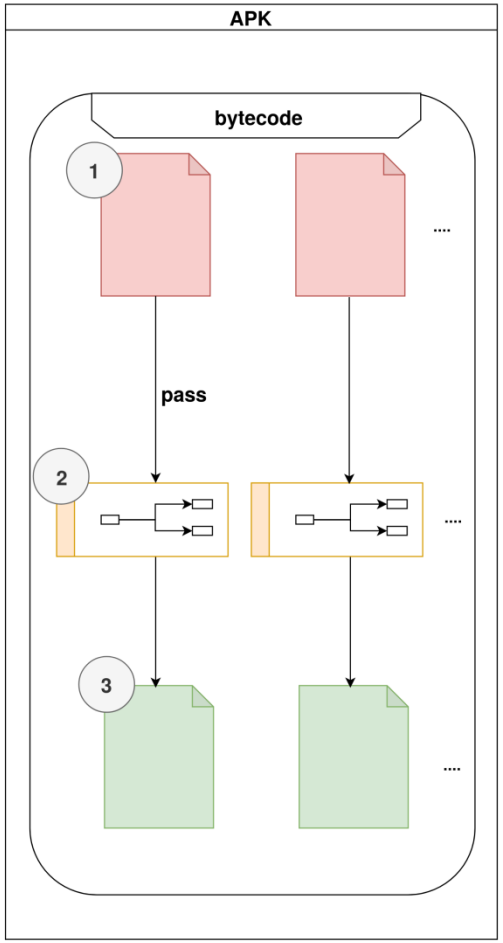
String Encryption

Taxonomy of String Encryption Techniques

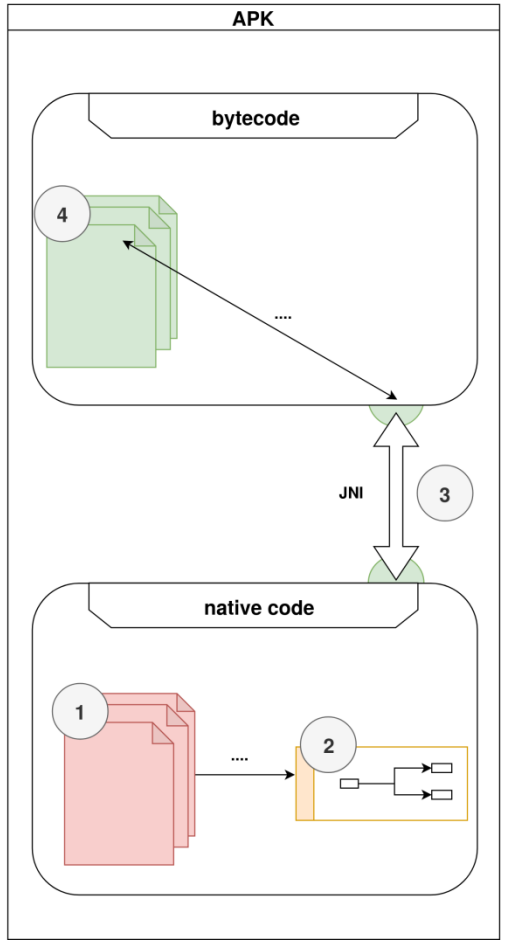
String Repository



Pass String



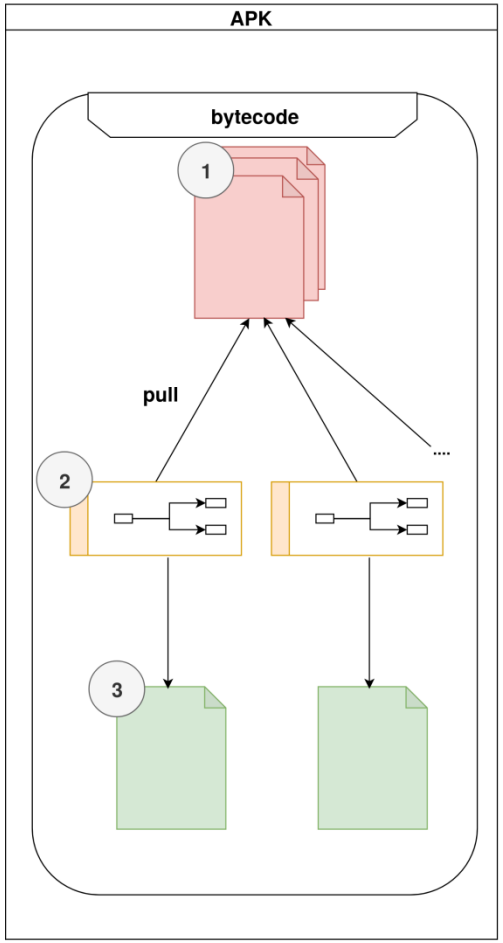
Native Library



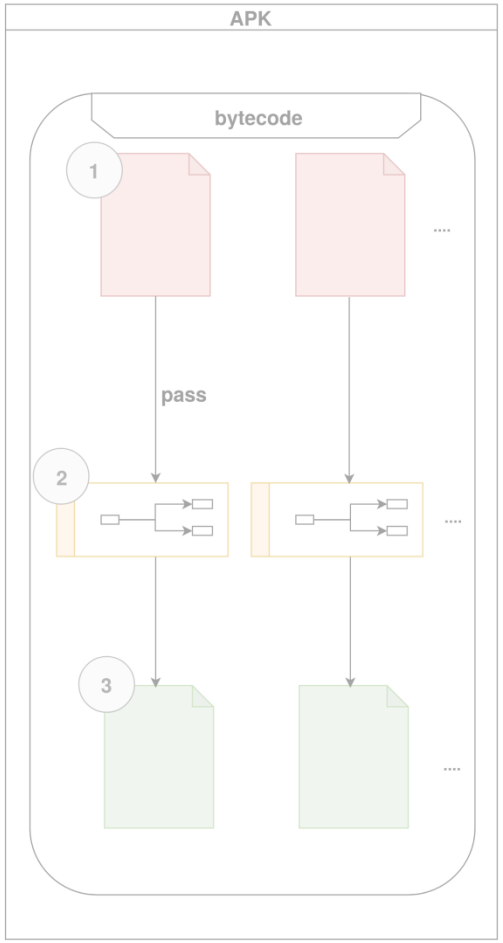
String Encryption

Taxonomy of String Encryption Techniques

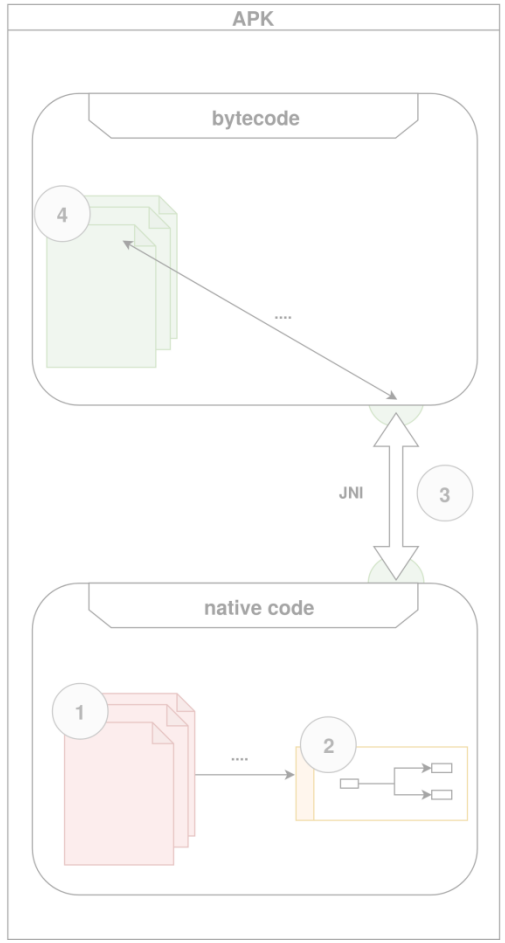
String Repository



Pass String



Native Library



Fraunhofer FKIE

String Encryption

String Repository: <clinit>

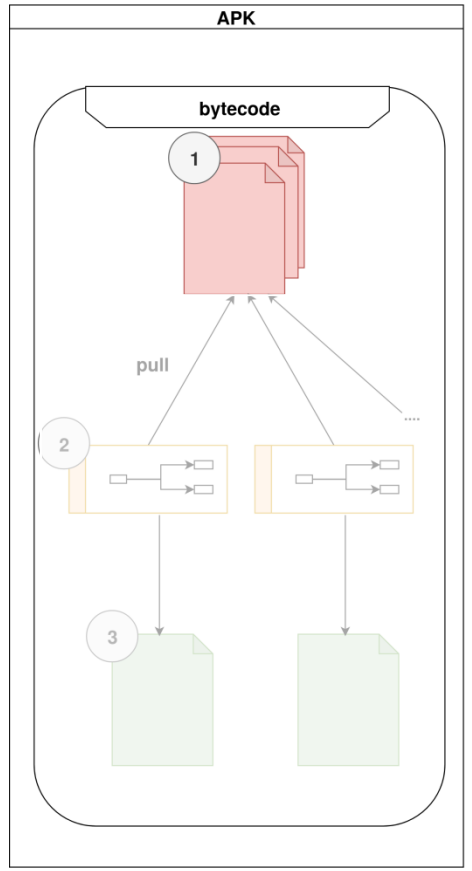
1

```
.field public static x:[B
.field public static y:[B
.field public static z:[B

# direct methods
.method static constructor <clinit>()V
    .locals 8
 ….
 new-array v0, v0, [B
 ….
 const/4 v1, 0x5
 const/16 v2, 0x22
 aput-byte v2, v0, v1
 sput-object v0, Lcom/android/system/conect/test/b;->y:[B
```

Android.OS.Triada2016 malware

String Repository



Fraunhofer FKIE

String Encryption

String Repository: <clinit>

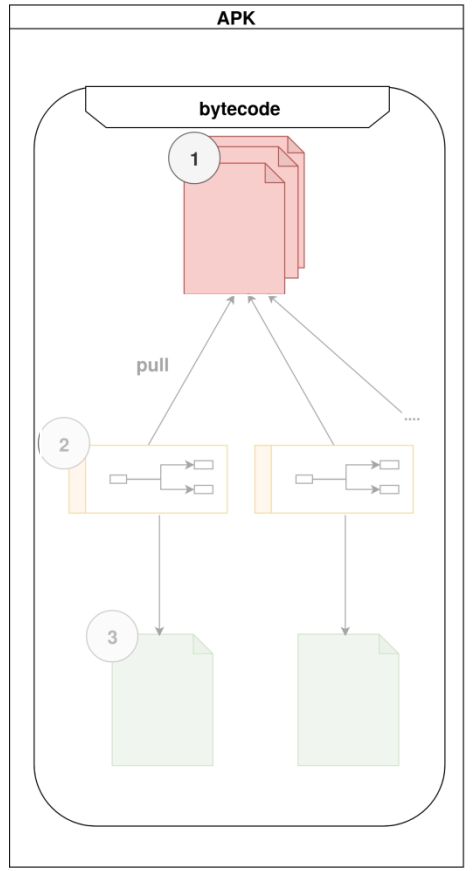
1

```
.field public static x:[B
.field public static y:[B
.field public static z:[B

# direct methods
.method static constructor <clinit>()V
    .locals 8
 ….
 new-array v0, v0, [B
 ….
 const/4 v1, 0x5
 const/16 v2, 0x22
 aput-byte v2, v0, v1
 sput-object v0, Lcom/android/system/conect/test/b;->y:[B
```

Android.OS.Triada2016 malware

String Repository



Fraunhofer FKIE

String Encryption

String Repository: <clinit>

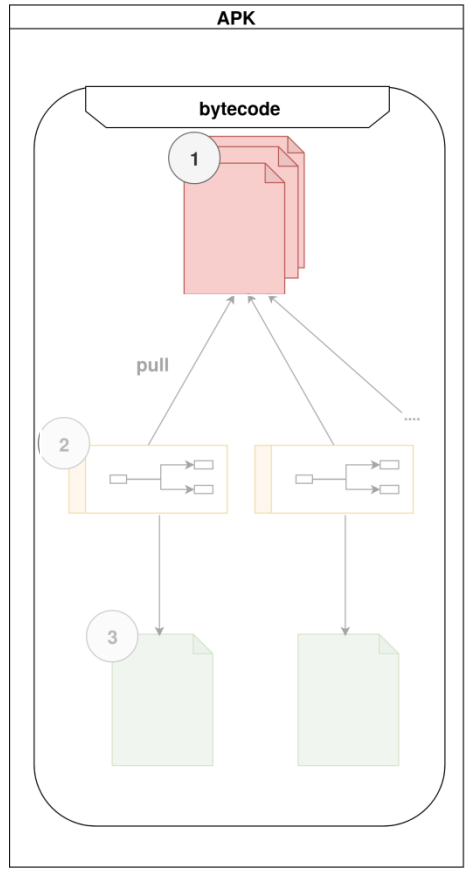
1

```
.field public static x:[B
.field public static y:[B
.field public static z:[B

# direct methods
.method static constructor <clinit>()V
    .locals 8
 ….
 new-array v0, v0, [B
 ….
 const/4 v1, 0x5
 const/16 v2, 0x22
 aput-byte v2, v0, v1
 sput-object v0, Lcom/android/system/conect/test/b;->y:[B
```

Android.OS.Triada2016 malware

String Repository



Fraunhofer FKIE

String Encryption

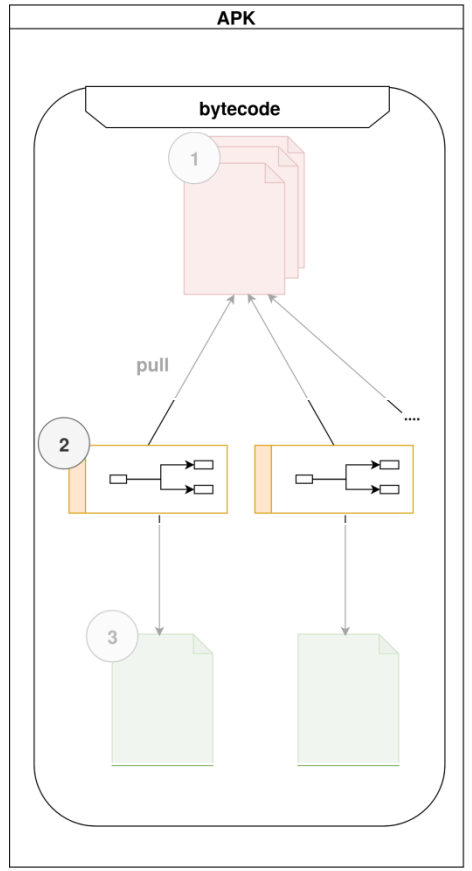
String Repository: <clinit>

2

```
sget-object v0, Lcom/android/system/conect/test/b;->y:[B

invoke-static {v0}, Lcom/android/system/conect/test/m;->a([B)Ljava/lang/String;

move-result-object v0
```



Android.OS.Triada2016 malware

String Encryption

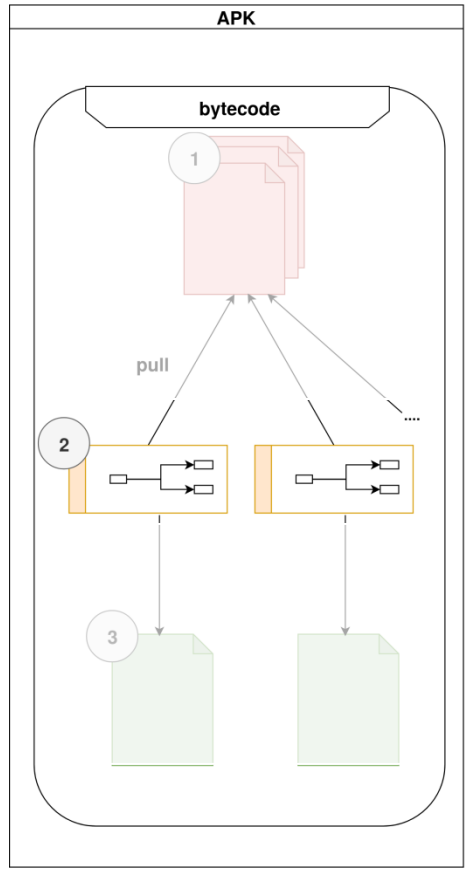
String Repository: <clinit>

2

```
sget-object v0, Lcom/android/system/conect/test/b;->y:[B

invoke-static {v0}, Lcom/android/system/conect/test/m;->a([B)Ljava/lang/String;

move-result-object v0
```



Android.OS.Triada2016 malware

String Encryption

String Repository: <clinit>

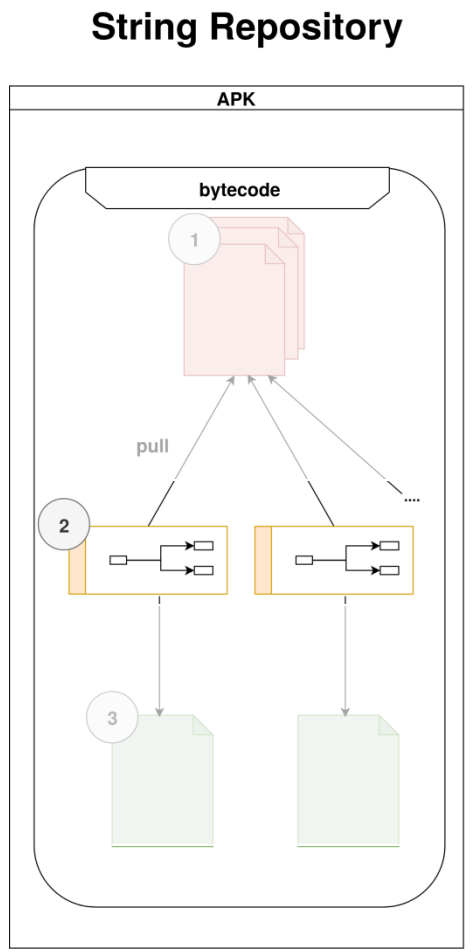
2

```
sget-object v0, Lcom/android/system/conect/test/b;->y:[B

invoke-static {v0}, Lcom/android/system/conect/test/m;->a([B)Ljava/lang/String;

move-result-object v0
```

```
.method public static a([B)Ljava/lang/String;
    …
     invoke-virtual {v0, p0}, Lcom/android/system/conect/test/j;->a([B)[B
```



Android.OS.Triada2016 malware

String Encryption

String Repository: <clinit>

2

```
sget-object v0, Lcom/android/system/conect/test/b;->y:[B

invoke-static {v0}, Lcom/android/system/conect/test/m;->a([B)Ljava/lang/String;

move-result-object v0
```

```
.method public static a([B)Ljava/lang/String;
    …
    invoke-virtual {v0, p0}, Lcom/android/system/conect/test/j;->a([B)[B
```

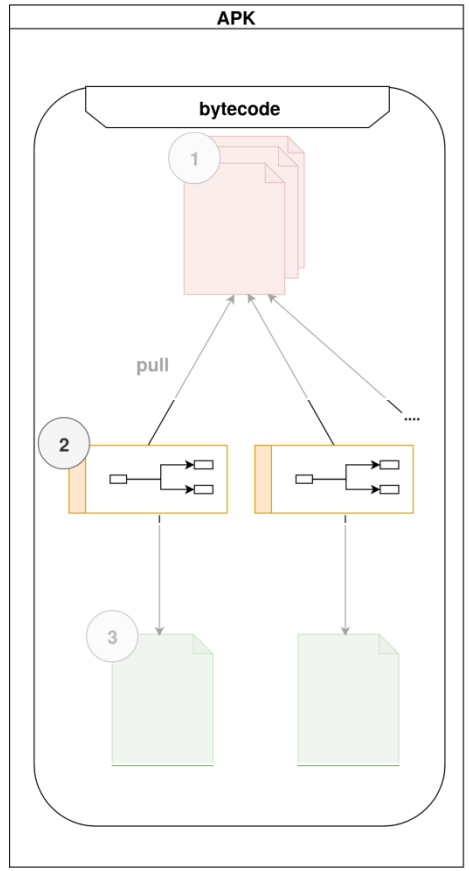
the actual decryption routine

```
aget-byte v5, v5, v6

xor-int/2addr v4, v5
…
```

Android.OS.Triada2016 malware

String Repository



Fraunhofer FKIE

String Encryption

String Repository: <clinit>

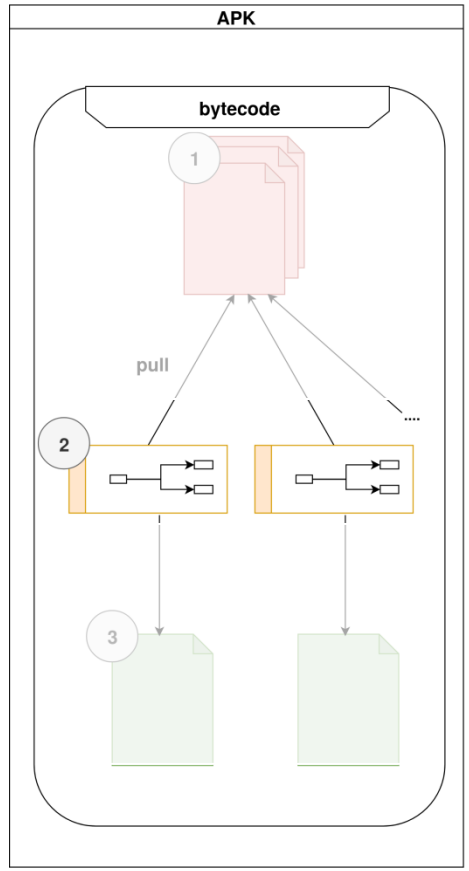
2

the used decryption algorithm differs between families and sometimes even between different versions of a family

```
const-string/jumbo v1, "DES/CBC/PKCS5Padding"

invoke-static {v1}, Ljavax/crypto/Cipher;->getInstance(Ljava/lang/String;)Ljavax/crypto/Cipher;
…
```

String Repository



String Encryption

String Repository: <clinit>

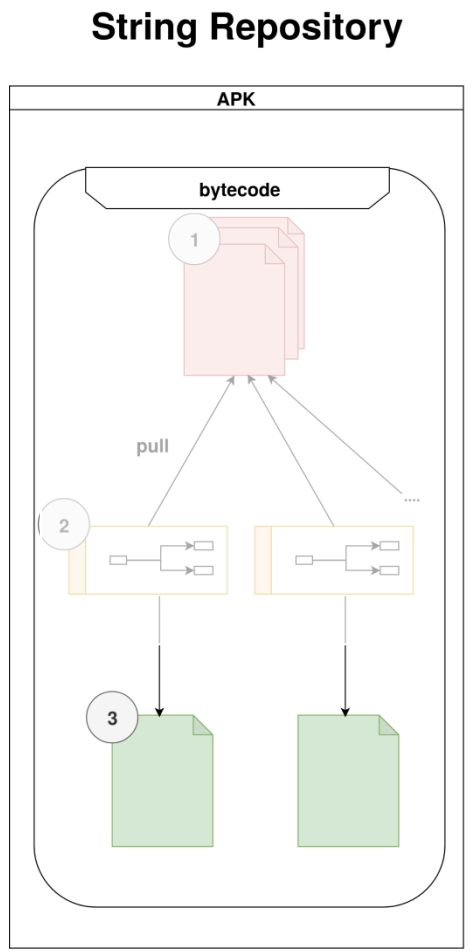
3

```
sget-object v0, Lcom/android/system/conect/test/b;->y:[B

invoke-static {v0}, Lcom/android/system/conect/test/m;->a([B)Ljava/lang/String;

move-result-object v0

# const-string v0, "android.intent.action.SCREEN_OFF"
```

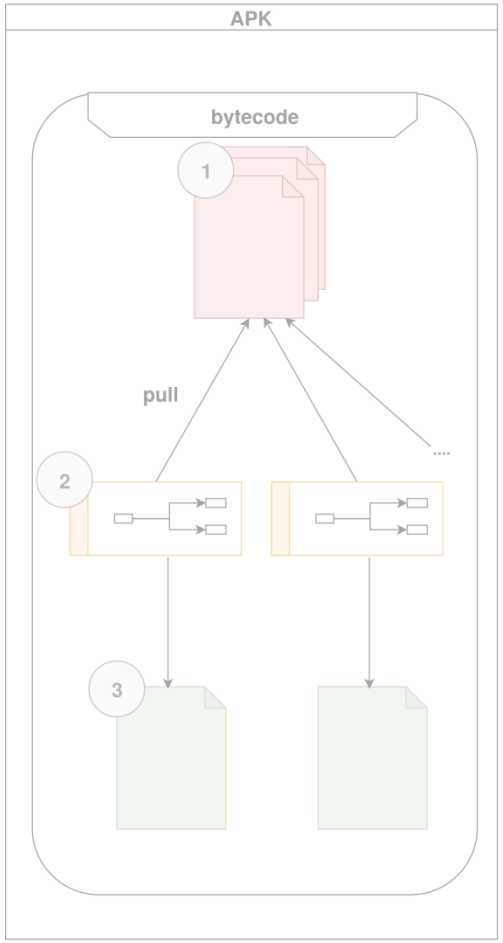


Android.OS.Triada2016 malware

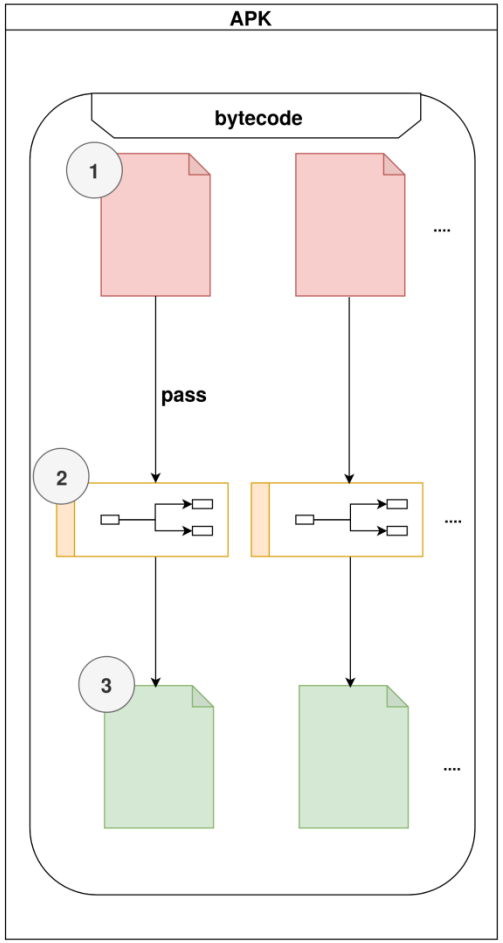
String Encryption

Taxonomy of String Encryption Techniques

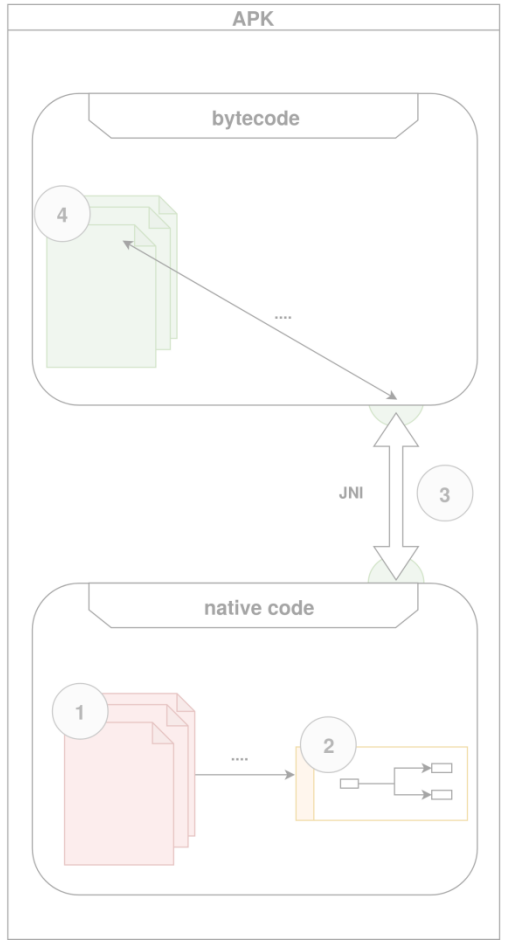
String Repository



Pass String



Native Library



String Encryption

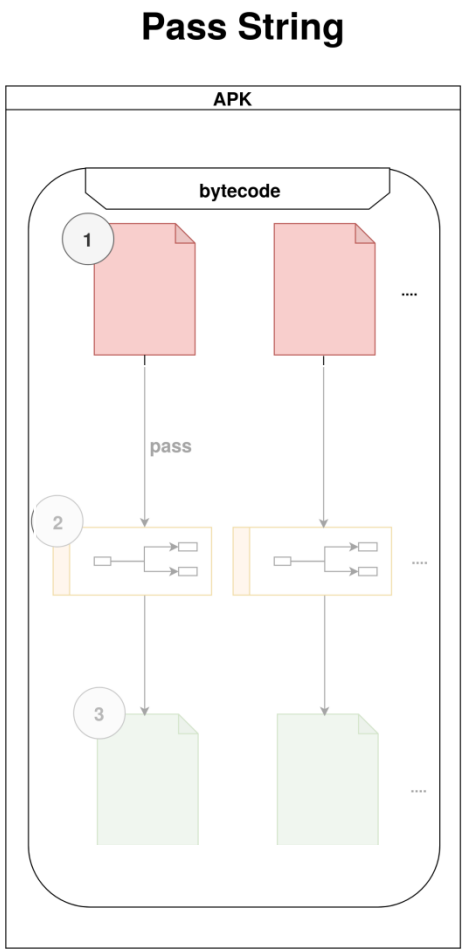
Pass String: Encrypted String

1

```
const-string v1, "7O9L5MxPf92ieYmDMGGpZIYAa0k3q/kQCI1amkd4ulrebFJy0bLs8PyOkJCdmu4L4U99b5FJ1d8="

invoke-static {v1}, Lcom/android/support/utils/g;->b(Ljava/lang/String;)Ljava/lang/String;

move-result-object v1
```



Trojan-SMS.AndroidOS.Prizmes.a malware

String Encryption

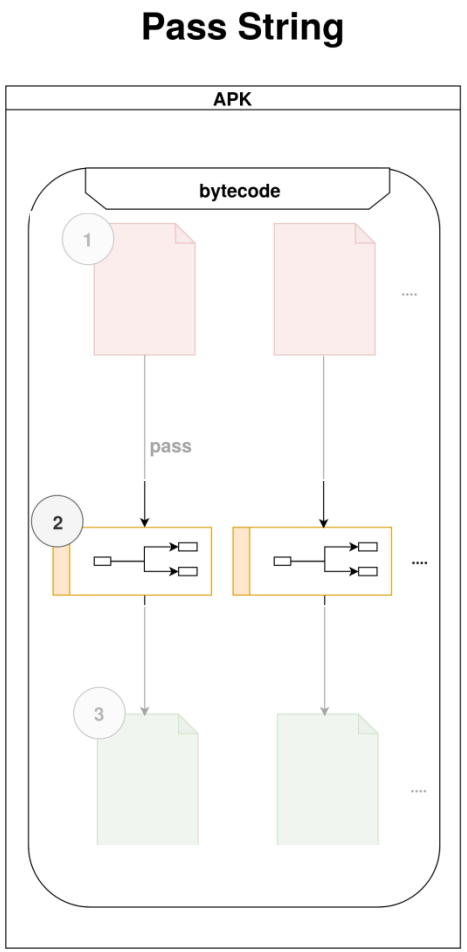
Pass String: Encrypted String

2

```
const-string v1, "7O9L5MxPf92ieYmDMGGpZIYAa0k3q/kQCI1amkd4ulrebFJy0bLs8PyOkJCdmu4L4U99b5FJ1d8="

invoke-static {v1}, Lcom/android/support/utils/g;->b(Ljava/lang/String;)Ljava/lang/String;

move-result-object v1
```



Trojan-SMS.AndroidOS.Prizmes.a malware

String Encryption

Pass String: Encrypted String

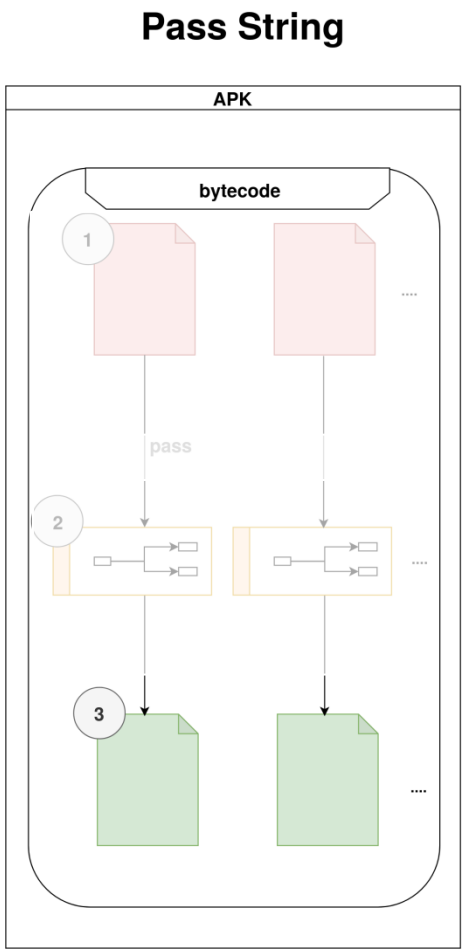
3

```
const-string v1, "7O9L5MxPf92ieYmDMGGpZIYAa0k3q/kQCI1amkd4ulrebFJy0bLs8PyOkJCdmu4L4U99b5FJ1d8="

invoke-static {v1}, Lcom/android/support/utils/g;->b(Ljava/lang/String;)Ljava/lang/String;

move-result-object v1

#const-string v1, "http://play.xhxt2016.com/logcollect/log-information"
```

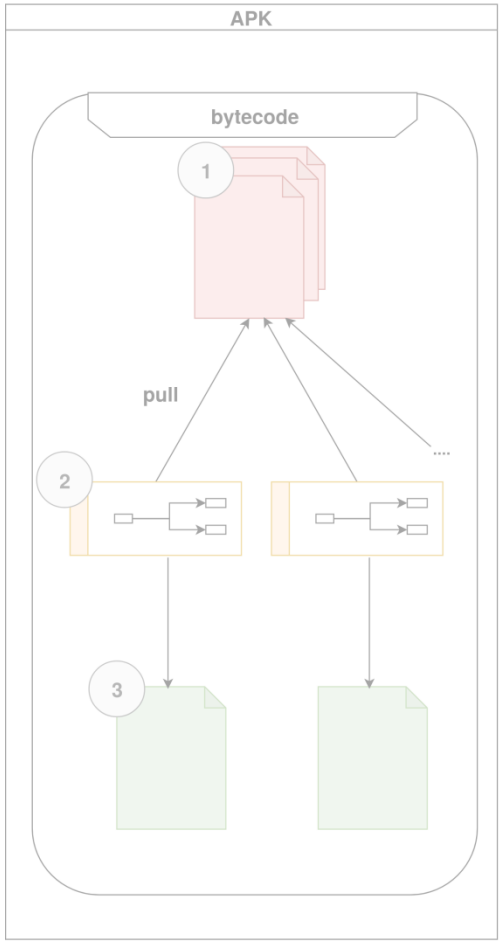


Trojan-SMS.AndroidOS.Prizmes.a malware

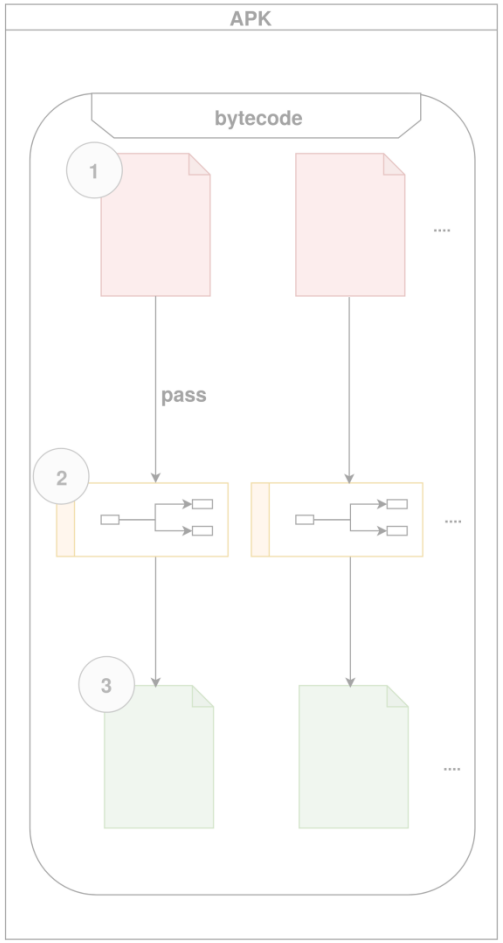
String Encryption

Taxonomy of String Encryption Techniques

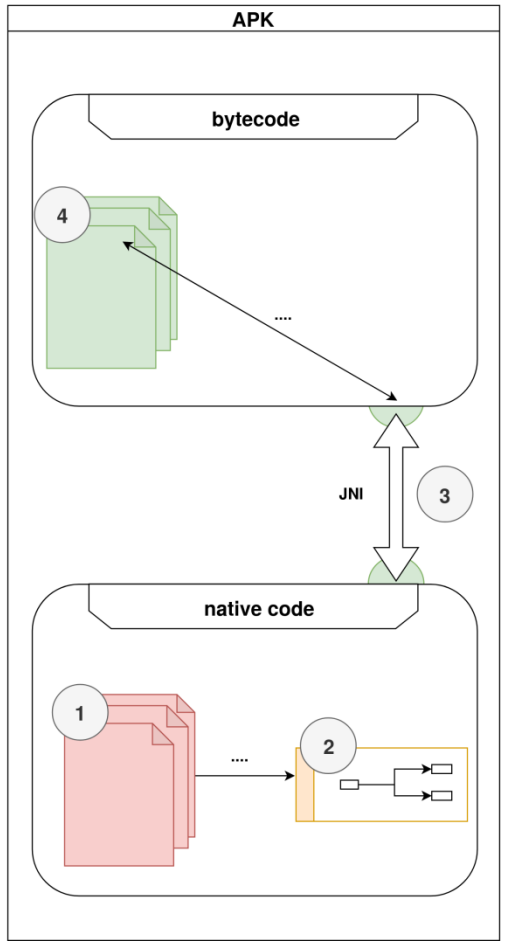
String Repository



Pass String



Native Library



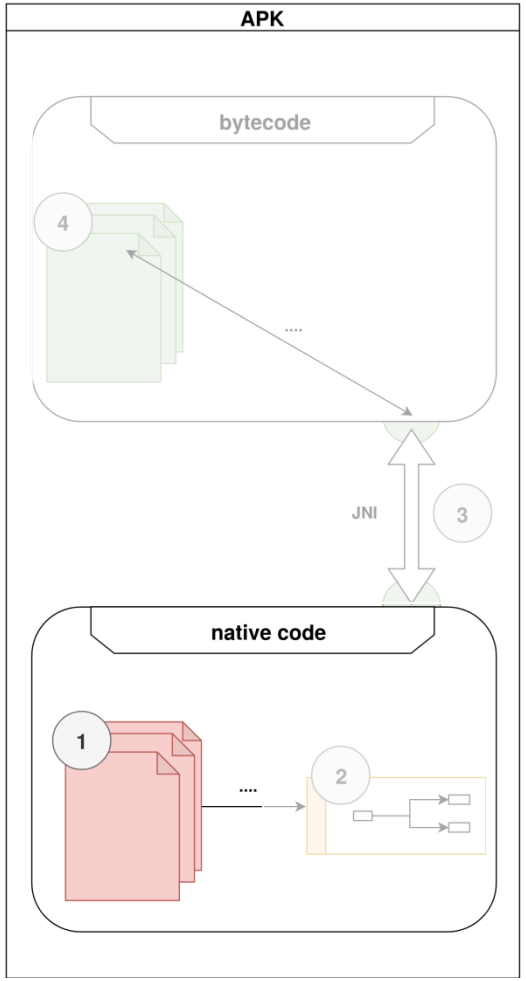
String Encryption

Native Library: Native to Bytecode

1

```
MOVS        R0, #0xA6
BX          LR
; End of function ox11105561640
```

flexispy malware



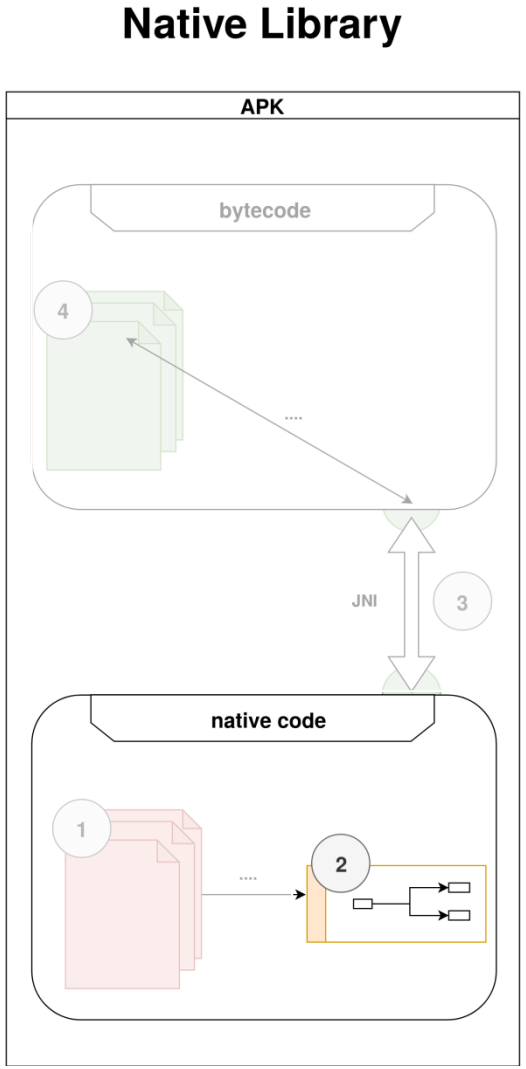
String Encryption

Native Library: Native to Bytecode

2

```
STRB.W      R0, [SP, #0x68+var_59]
BL          ox11105561640
ADDS        R0, #0x19
STRB.W      R0, [SP, #0x68+var_58]
BL          ox11163188672
ADDS        R0, #0x19
```

flexispy malware



String Encryption

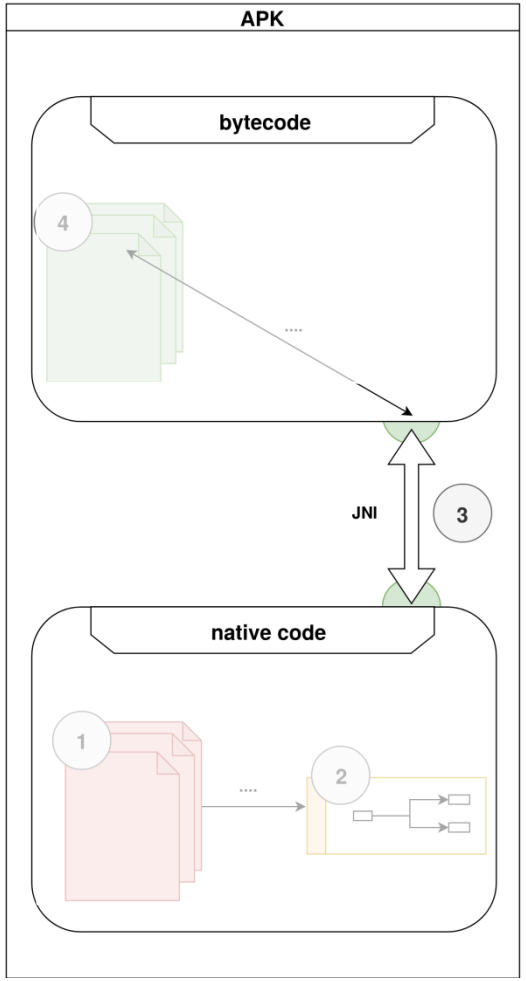
Native Library: Native to Bytecode

3

```
.class public Lk/v/F;
…
# virtual methods
.method public native fpu()[B
.end method
```

flexispy malware

Native Library



String Encryption

Native Library: Native to Bytecode

4

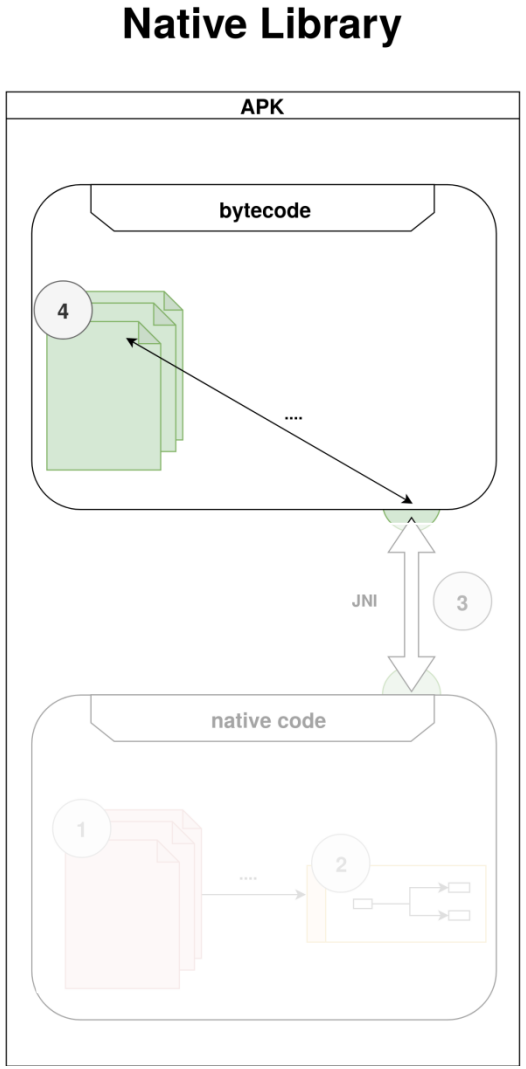
```
.method public static a()[B

    invoke-virtual {v0}, Lk/v/F;->fpu()[B

    move-result-object v0

    return-object v0
.end method
```

flexispy malware



String Encryption

Ground truth

- Labelled dataset
  - APKs from malpedia with String Encryption
  - number of deobfuscated strings of each sample
  - location of the deobfuscation routine

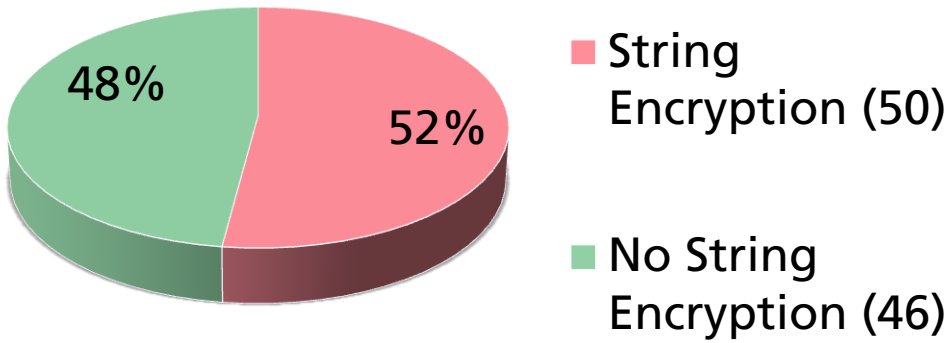
https://github.com/fkie-cad/destroid


```
{
    "family" : "apk.anubis",
    "alt_names" : "apk.bankbot",
    "state" : "finished",
    "analyzed" : true,
    "string encryption" : {
        "is" : true,
        "type" : [
            "PS.encrypted string (2)",
            "PS.encrypted string (2)",
            "PS.encrypted string (4)"
        ],
        "deobfuscations" : [
            "2dfde3d394b7eaf3a45693dc95f9c5540c9fd2b3bc7e89e9ebc9d12963c00bee bankbot_sample1.apk (426)",
            "aeaccdc3fb0ddb674770ff87007b4454b0a8d706ebd57ee7e75599ca7bda19d8 bankbot_sample2.apk (454)",
            "16cb502391b003ec9c8a4fc604f7939d8f9224eecb6d0f8a801f72069508943e bankbot_sample3.apk (98)"
        ],
        "deobfuscation routine" : [
            "bankbot_sample1|Lturs/rickertsit/a;->a(Ljava/lang/String;)Ljava/lang/String;",
            "bankbot_sample1|Lturs/rickertsit/b;->a(Ljava/lang/String;)Ljava/lang/String;",
            "bankbot_sample2|Lcom/google/android/gms/common/zzc$zza;->zzcm(Ljava/lang/String;)[B",
            "bankbot_sample2|Lcom/google/android/gms/internal/zzal;->zzb([BLjava/lang/String;)Ljava/lang/String;",
            "bankbot_sample3|Lorg/support/q;->M(Ljava.lang.Object;)Ljava.lang.String;",
            "bankbot_sample3|Lcom/google/android/gms/common/zzh;->zzgf(Ljava.lang.String)B[",
            "bankbot_sample3|Lcom/wierdhammer/expenses/s;->d(Ljava.lang.Object;)Ljava.lang.String",
            "bankbot_sample3|Lcom/wierdhammer/expenses/h;->d(Ljava.lang.Object;)Ljava.lang.String"
        ]
    }
}
```

String Encryption

Ground truth

malpedia Android families

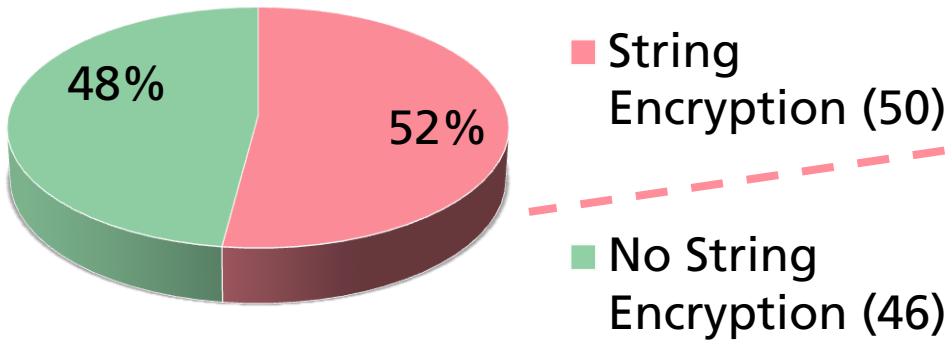


Details in paper

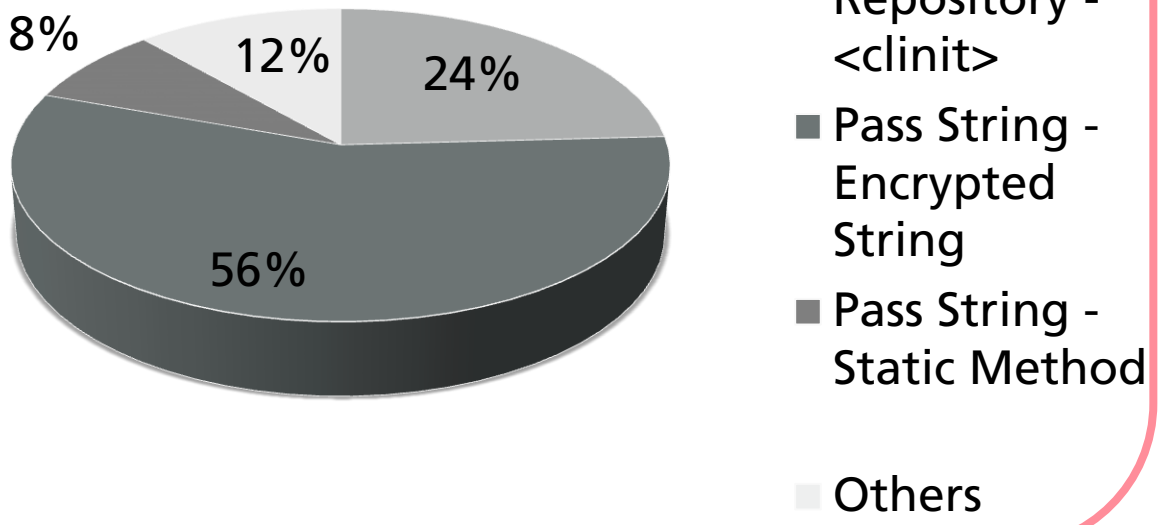
String Encryption

Ground truth

malpedia Android families



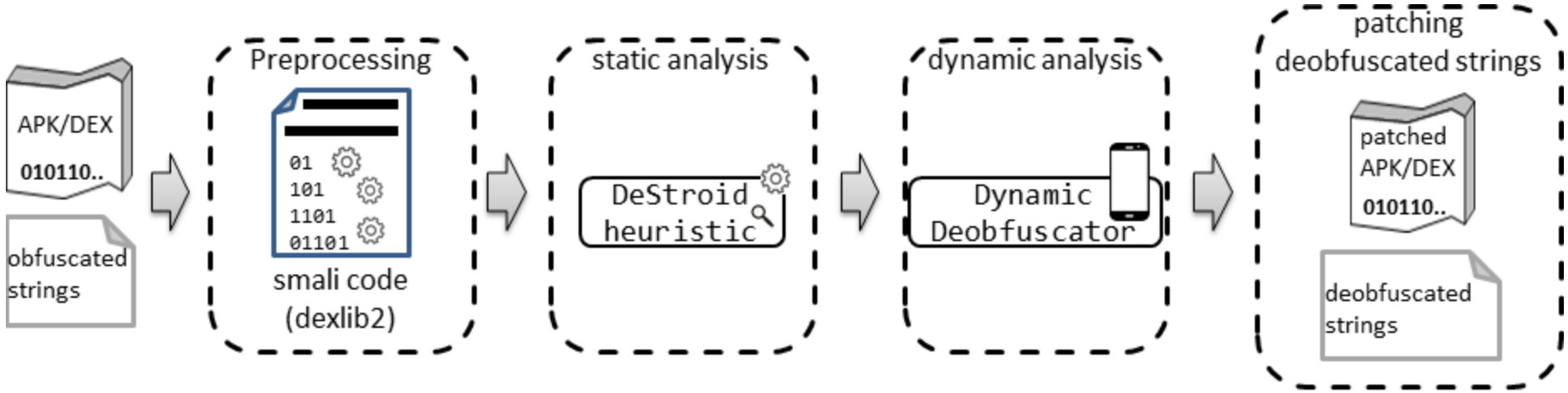
Distribution of String Encryption types



Details in paper

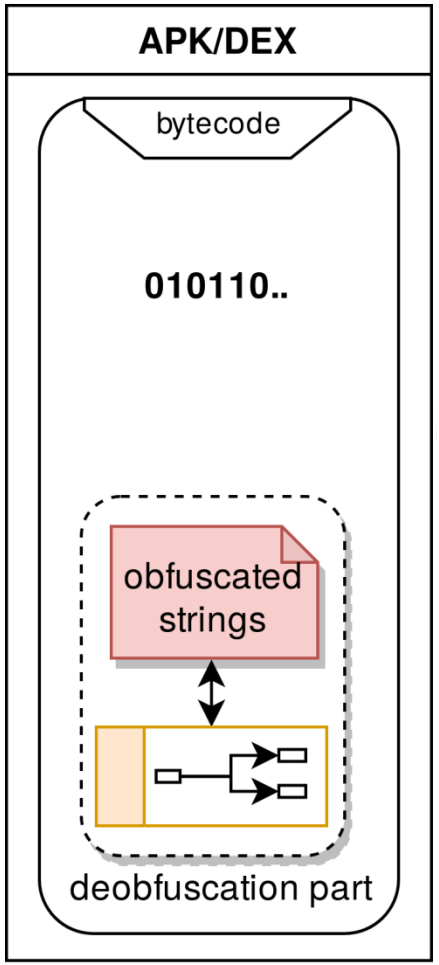
DeStroid

High-Level Workflow



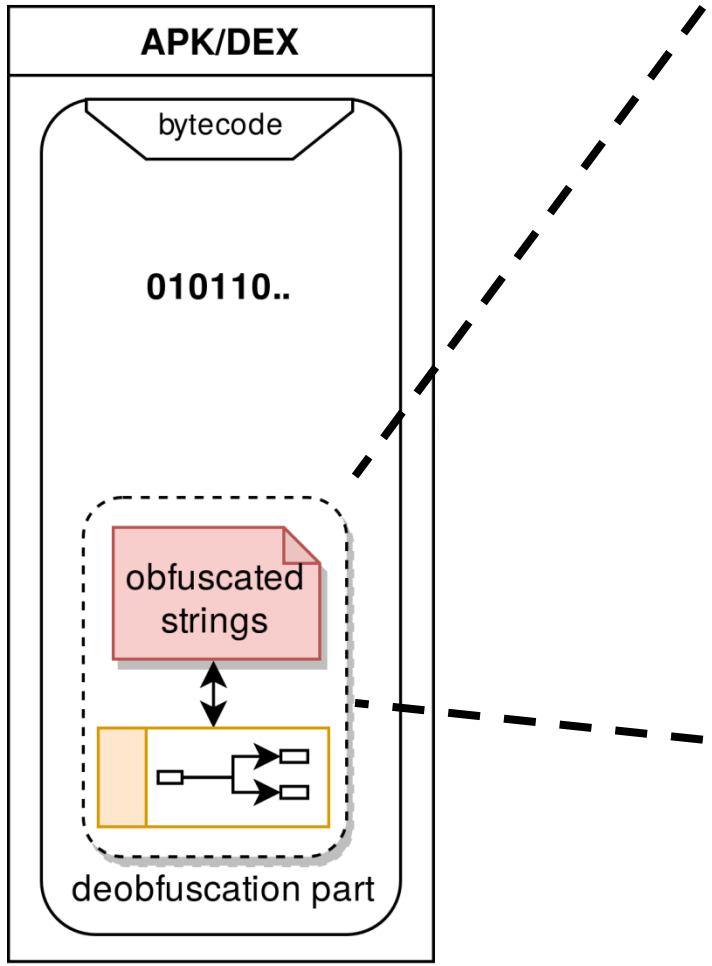
DeStroid

Static Analysis: DeStroid-Heuristic



DeStroid

Static Analysis: DeStroid-Heuristic



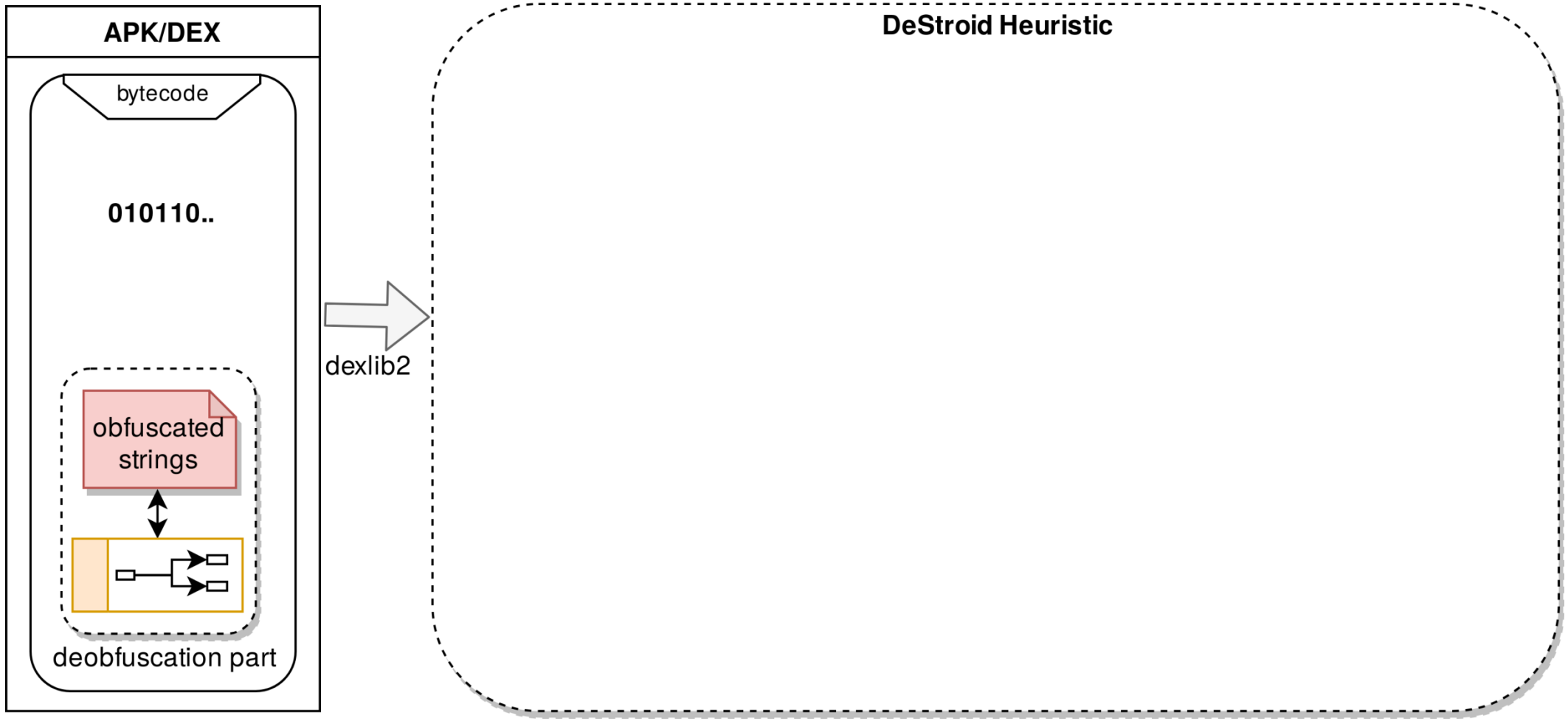

```
public class MortgagesApplication extends Application {
    private static final String a = Transfusion.a[20];
    private static final String b = Transfusion.a[5];
    private static final String c = Transfusion.a[8];
    private static final String d = Transfusion.a[25];
    private static Boolean e = Boolean.valueOf(false);
    private ClassLoader f;
    private File g;
    private File h;

    private String a(int i) {
        String name = getClass().getName();
        int lastIndexOf = name.lastIndexOf(i, name.length());
        return lastIndexOf == -1 ? name : name.substring(0, lastIndexOf) + Transfusion.a[24];
    }

    private static Field a(Object obj, String str) {
        Object obj2 = obj.getClass();
        while (obj2 != null) {
            try {
                Field field = (Field) obj2.getClass().getDeclaredMethod(Transfusion.a[13], new Class[]{String.class}).invoke(obj2, new Object
                if (((Boolean) field.getClass().getMethod(Transfusion.a[28], new Class[0]).invoke(field, new Object[0])).booleanValue()) {
                    return field;
                }
                field.getClass().getMethod(Transfusion.a[12], new Class[]{Boolean.TYPE}).invoke(field, new Object[]{Boolean.valueOf(true)});
                return field;
            } catch (Exception e) {
                Class obj22 = obj22.getSuperclass();
            }
        }
        return null;
    }
```

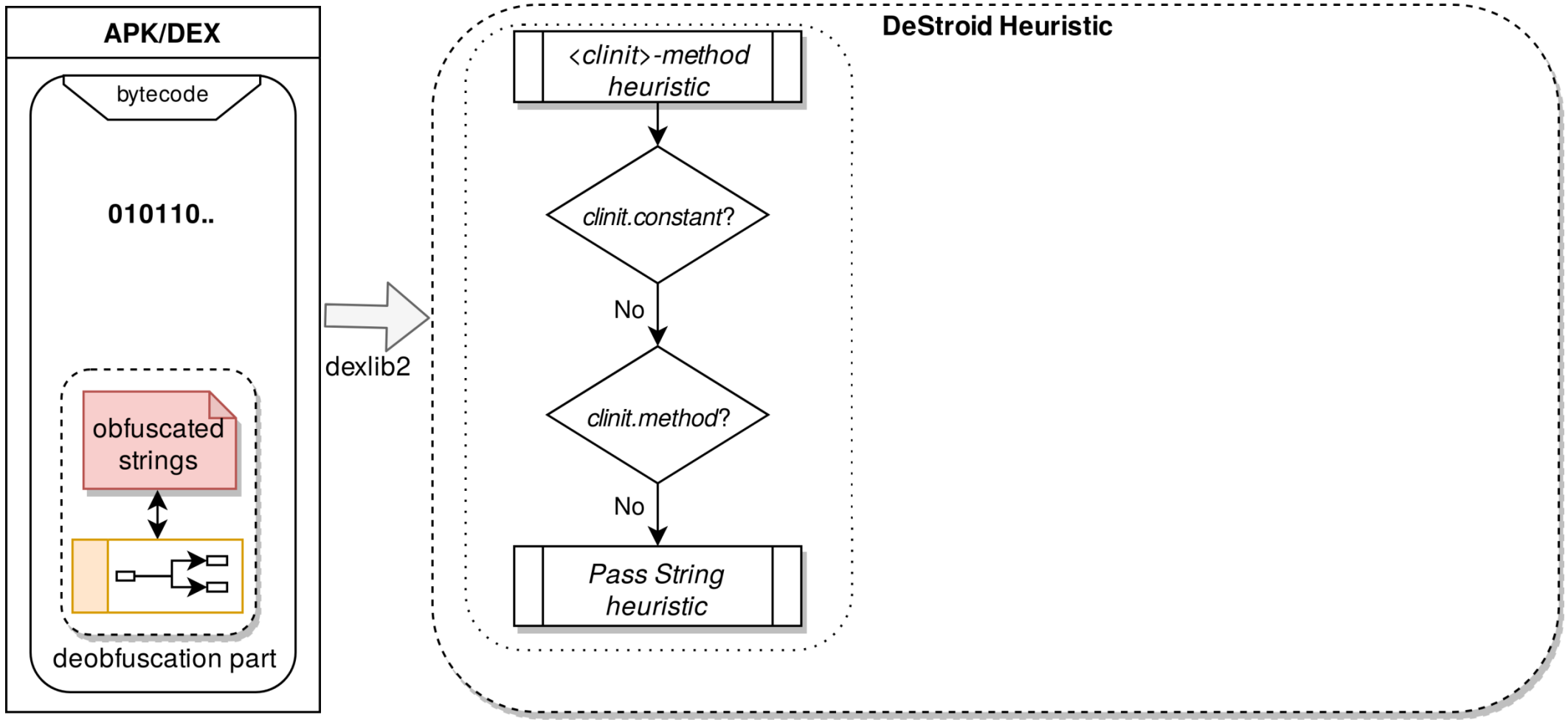
DeStroid

Static Analysis: DeStroid-Heuristic



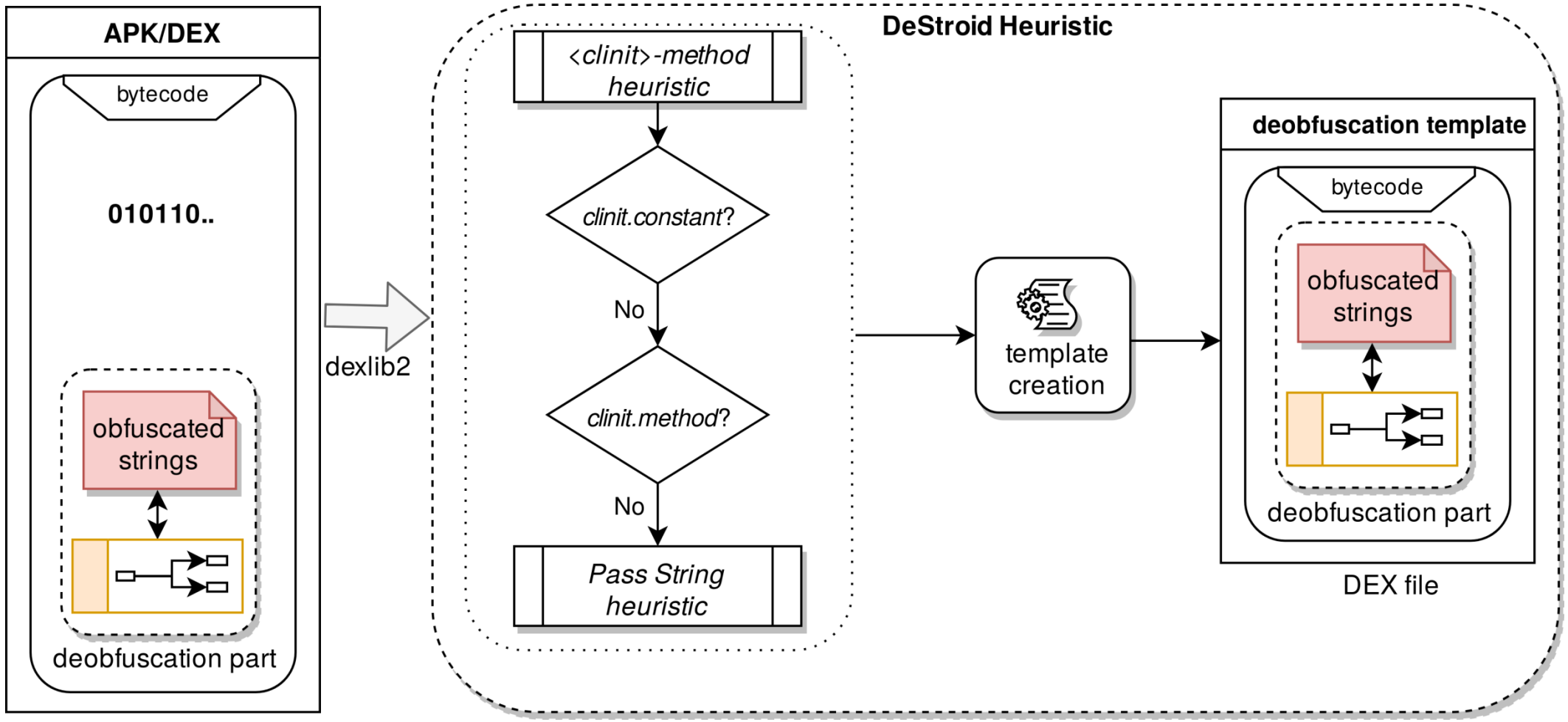
DeStroid

Static Analysis: DeStroid-Heuristic



DeStroid

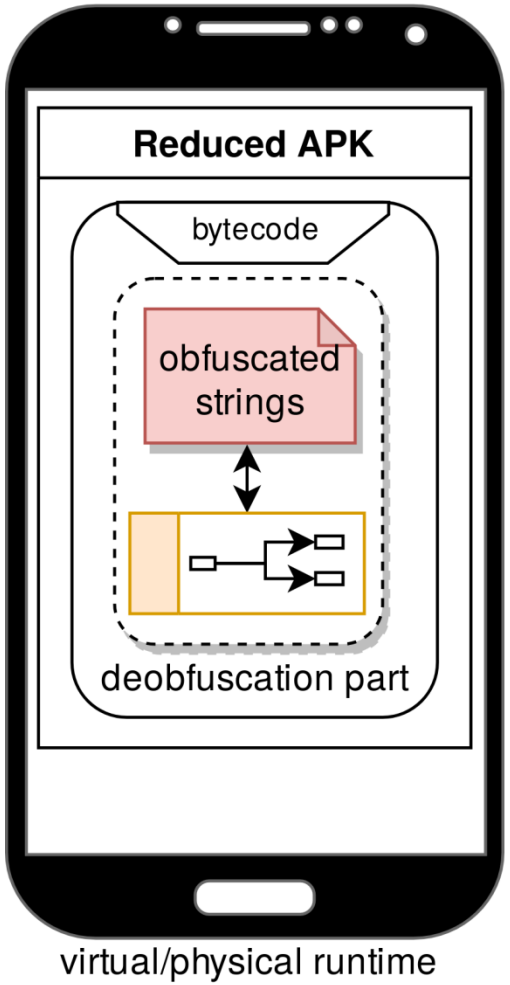
Static Analysis: DeStroid-Heuristic



Fraunhofer FKIE

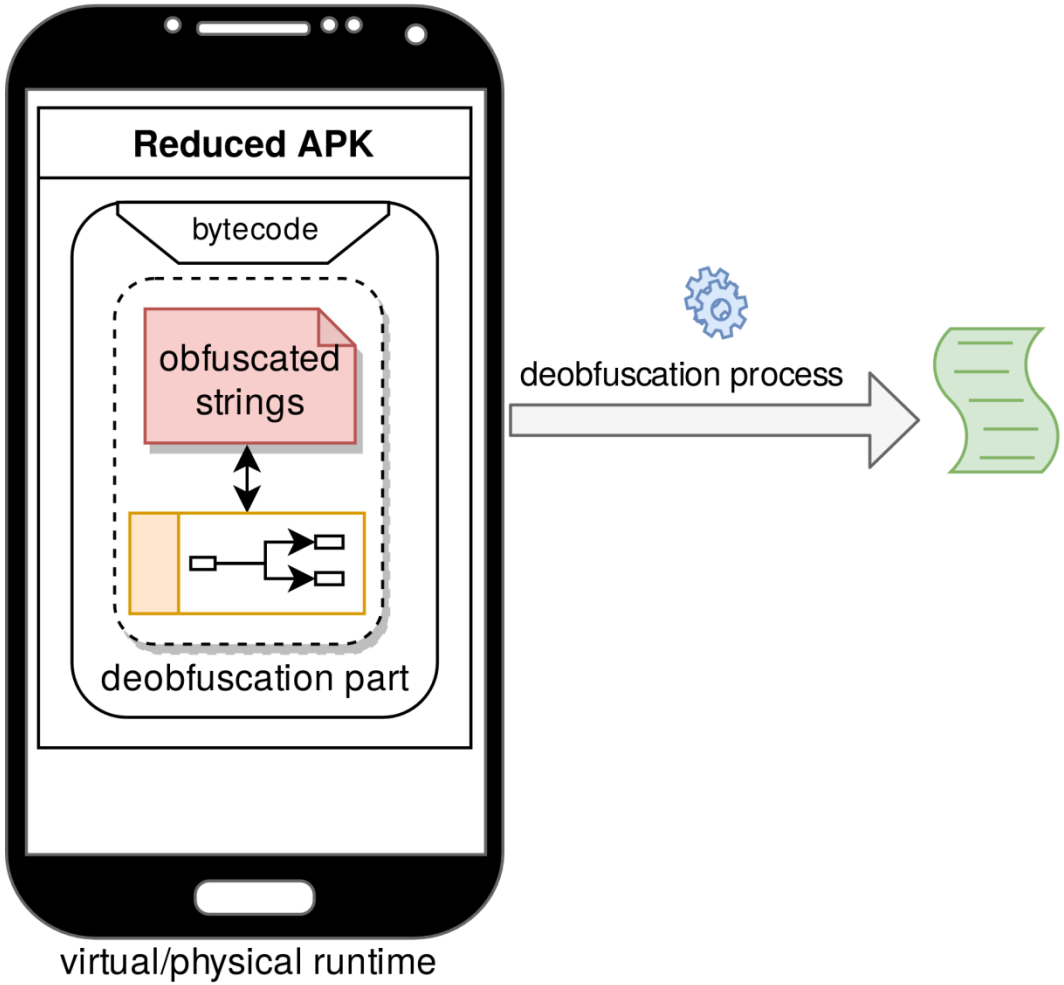
DeStroid

Dynamic analysis: Dynamic Deobfuscation



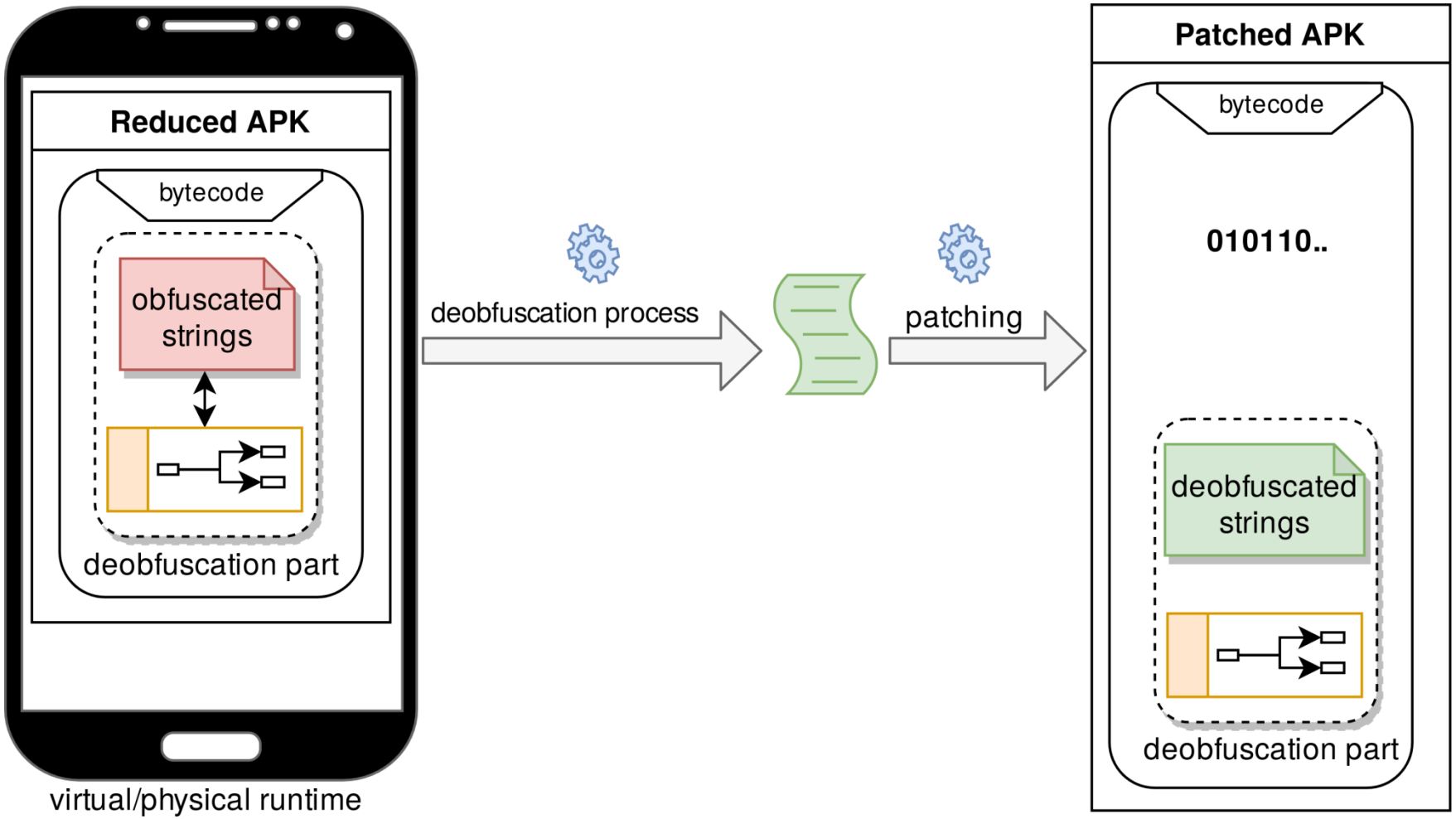
DeStroid

Dynamic analysis: Dynamic Deobfuscation



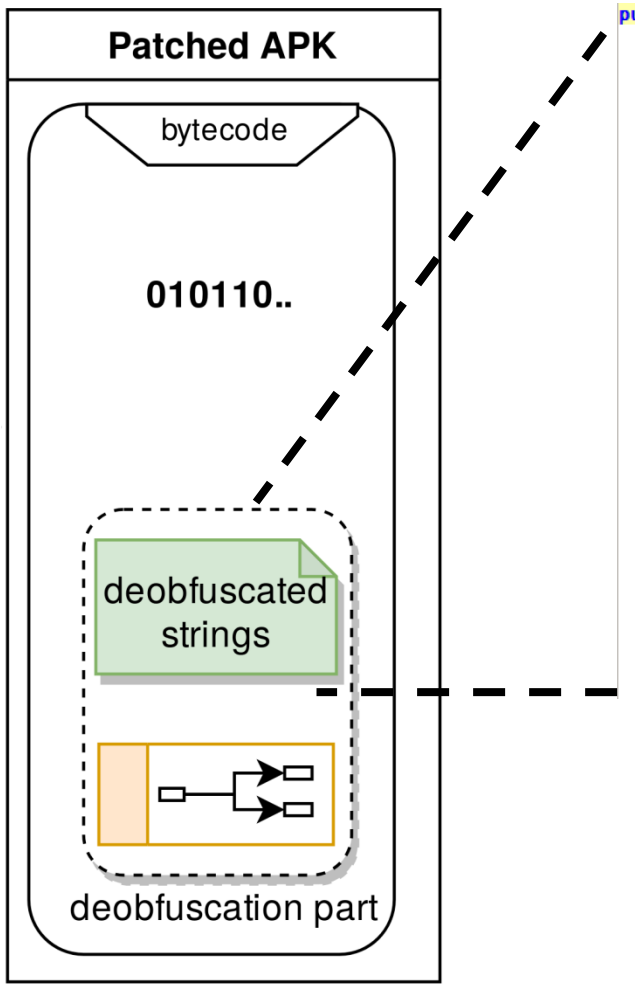
DeStroid

Dynamic Deobfuscation



DeStroid

Patching




```java
public class MortgagesApplication extends Application {
    private static final String a = "'MortgagesApplication'";
    private static final String b = "'dex2'";
    private static final String c = "'reallocates.db3'";
    private static final String d = "'test.apk'";
    private static Boolean e = Boolean.valueOf(false);
    private ClassLoader f;
    private File g;
    private File h;

    private String a(int i) {
        String name = getClass().getName();
        int lastIndexOf = name.lastIndexOf(i, name.length());
        return lastIndexOf == -1 ? name : name.substring(0, lastIndexOf) + "'.UnrecognisablyActivity'";
    }

    private static Field a(Object obj, String str) {
        Object obj2 = obj.getClass();
        while (obj2 != null) {
            try {
                Field field = (Field) obj2.getClass().getDeclaredMethod("'getDeclaredField'", new Class[]{String.class}).invoke(obj2, new Object[]{str});
                if (((Boolean) field.getClass().getMethod("'isAccessible'", new Class[0]).invoke(field, new Object[0])).booleanValue()) {
                    return field;
                }
                field.getClass().getMethod("'setAccessible'", new Class[]{Boolean.TYPE}).invoke(field, new Object[]{Boolean.valueOf(true)});
                return field;
            } catch (Exception e) {
                Class obj22 = obj22.getSuperclass();
            }
        }
        return null;
    }
```

Evaluation

Setup

- ground truth used as dataset for evaluation

malpedia

Fraunhofer FKIE

Evaluation

Setup

- ground truth used as dataset for evaluation

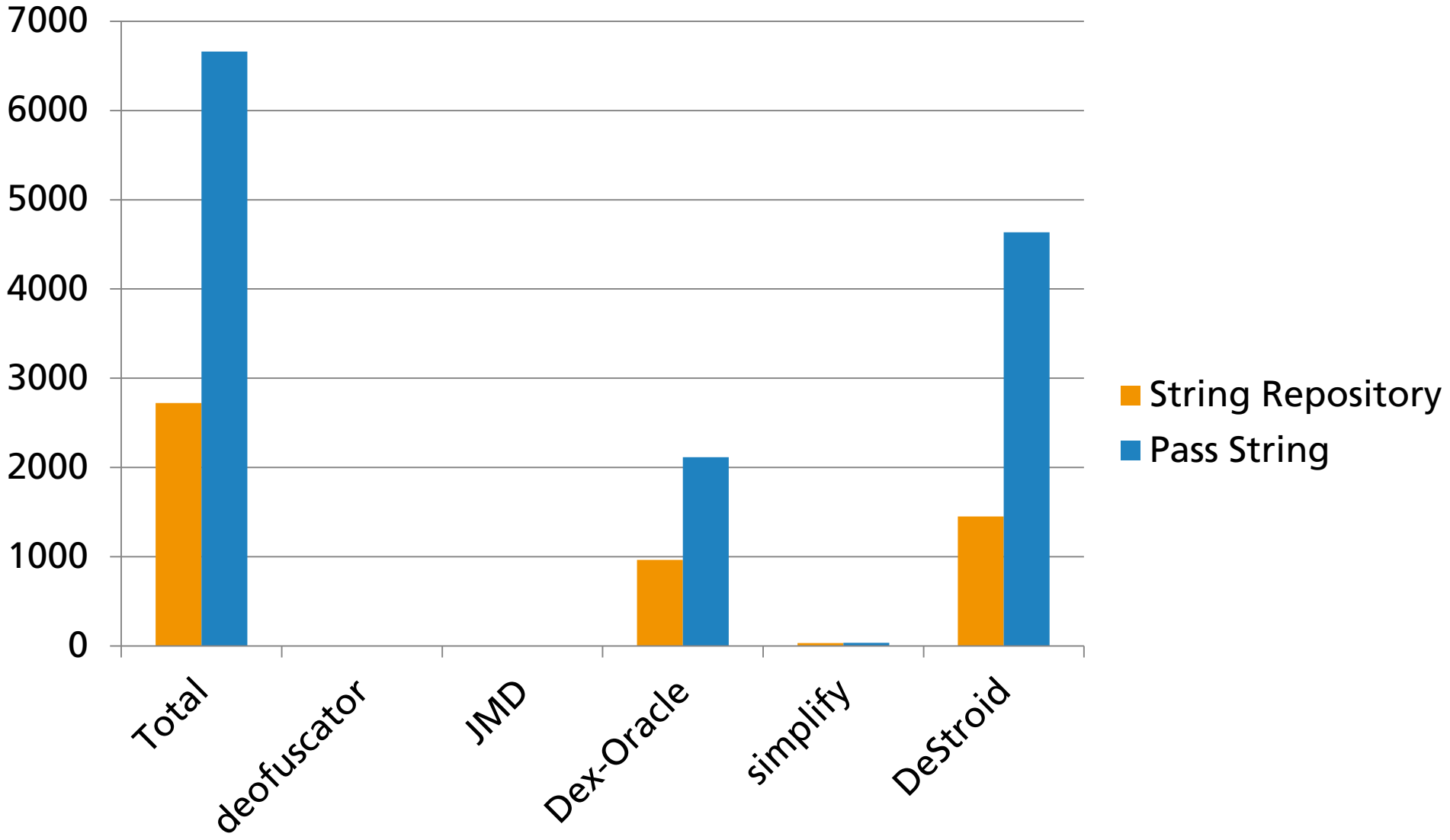
malpedia

- investigate performance of different string encryption deobfuscation approaches
  - `deobfuscator`
  - `JMD`
  - `Dex-Oracle`
  - `simplify`
  - `DeStroid`

- deobfuscation details (in paper)

Evaluation

Results



Details in paper

Fraunhofer FKIE

Summary

- A taxonomy of String Encryption implementations
  - usage in current Android malware
  - identification of their major String Encryption types
- A labelled dataset of Android malware samples
  - number of expected deobfuscated strings
  - location of the deobfuscation part
- DeStroid
  - our solution on automatically deobfuscate decrypted Strings
  - works best for String Repository and Pass String
  - online at https://github.com/fkie-cad/destroid

Thanks for your attention!

https://github.com/fkie-cad/destroid