



**black hat** abu dhabi  
+2011

In partnership with:



Supported by:



## Android: From Reversing to Decompilation

*Anthony Desnos, Geoffroy Gueguen  
ESIEA: Operational Cryptology and Virology Laboratory  
desnos@esiea.fr, gueguen@esiea.fr*

# Current section

Android

Analysis

Static Analysis

Visualization

Demos

Conclusion



# Android

## The platform

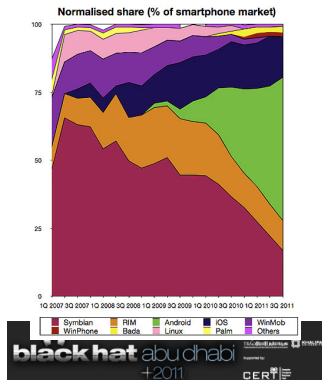
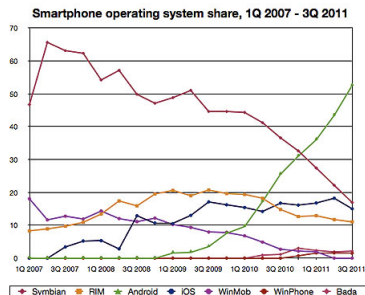
- ▶ Google purchased the initial developer of the software, Android Inc., in 2005
- ▶ The unveiling of the Android distribution on November 5, 2007
- ▶ October 2008: Android Market
- ▶ 295.000 applications on the Android Market, 6 billions downloads
- ▶ Percentage of apps that are free : 60%



# Android

## The platform

- ▶ Android runs 52% of smartphones sold (Gartner)



# Android

## The platform

- ▶ Third party applications written in Java, executed on the Dalvik Virtual Machine
- ▶ Java bytecode converted in Dalvik bytecode (stack-based machine vs register based machine)
- ▶ Applications are packaged in the APK format
- ▶ A virtual machine (Linux user-based protection) per application
- ▶ Permissions per application



# Android

## APK

- ▶ ZIP format
- ▶ classes.dex: Dalvik Executable Format
- ▶ ressources: images, strings ...
- ▶ assets: raw ressources
- ▶ native libraries
- ▶ manifest file: what to do with all the top-level components (specifically activities, services, broadcast receivers, and content providers) and specifies which permissions are required in an application



# Android

## Disassembling Dalvik bytecode

- ▶ Instructions use registers,
- ▶ Impossible to change the bytecode on the fly,
- ▶ Less than 0xff instructions,
- ▶ Instruction format:
  - ▶ nop, move\*, invoke\*, goto\*, cmp\*, \*-switch, add\*, sub\* ...



## Dalvik bytecode

```
In [3]: d.CLASS_Lcom_xxx_yyy_ApkReceiver.METHOD_onReceive.pretty_show()
        ENCODED_METHOD method_idx_diff=885 access_flags=1 code_off=0x16f3c (Lcom/xxx/yyy/ApkReceiver; (Landroid/content/Context; Landroid/content/Intent;)V,onReceive)
        *****
        DALK_VIK_CODE :
            REGISTERS_SIZE 0x5
            INS_SIZE 0x3
            OUTS_SIZE 0x3
            TRIES_SIZE 0x0
            DEBUG_INFO_OFF 0x343bb
            INSNS_SIZE 0xb

        onReceive-BB@0x0 :
            0(0) new-instance v0 , [type@ 27 Landroid/content/Intent;]
            1(4) const-class v1 , [type@ 257 Lcom/xxx/yyy/MyService;]
            2(8) invoke-direct v0 , v3 , v1 , [meth@ 117 Landroid/content/Intent; (Landroid/content/Context; Ljava/lang/Class;) V <init>]
            3(e) invoke-virtual v3 , v0 , [meth@ 115 Landroid/content/Context; (Landroid/content/Intent;) Landroid/content/ComponentName; startService]
            4(14) return-void

        *****
```



# Android

## Manifest file

- ▶ Activities, services, content providers, and broadcast receivers
- ▶ Permissions:
  - ▶ Camera functions
  - ▶ Location (GPS) functions
  - ▶ Bluetooth functions
  - ▶ Telephony functions
  - ▶ SMS/MMS functions
  - ▶ Network functions
- ▶ Before the installation of an application, all permissions are asked and detailed to the end user



# Android

## Protecting Your Applications

- ▶ Obfuscators like ProGuard (GPL), Dasho,
- ▶ Works mainly at the java bytecode level,
- ▶ Techniques:
  - ▶ names obfuscation,
  - ▶ optimization,
  - ▶ CFG obfuscation.



# Android

## Problem

- ▶ A major problem in the Android market is the theft of applications:
  - ▶ download an application (free or not) on the official Android Market
  - ▶ crack/re-package/infect it by using smali/baksmali/apk-tool
  - ▶ push it (free or not) on the market



# Android

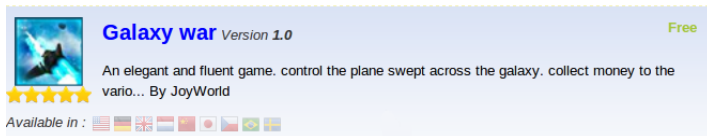
## Is it your application ? :)

- ▶ Kevin Baker (an android developer, Neolithic Software), interviewed by The Guardian about his application: Sinister Planet
  - ▶ "I have a game on the market called Sinister Planet which was released about eight months ago"
  - ▶ "One of my customers emailed me three weeks ago, and informed me that another company was selling a version of my app - pirated and uploaded as their own. Of course I contacted Google right away. It took Google two days to take the app down. This publisher was also selling other versions of pirated games. [...] You'd think [Google] might have a hotline for things like that!"



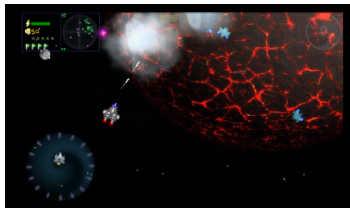
# Android

Is it your application ? :)



# Android

Is it your application ? :)




# Android

Is it your application ? :)

► ElectricSleep (Jon Willis)


**ElectricSleep (Free Beta)**  
Jon Willis



★★★★★ (738)

INSTALL

More from developer



**RAMDroid - RAM Widget**  
JON WILLIS  
★★★★★ (226)  
Free

OVERVIEWUSER REVIEWSWHAT'S NEWPERMISSIONS

## Description

Improve the quality of your sleep with this smart alarm clock.

ElectricSleep is an alarm clock that records your sleep cycles and wakes you up gently during a light sleep cycle. The sleep data it records is saved and analyzed so that you can understand and improve upon your sleeping habits.

Please donate to support development!

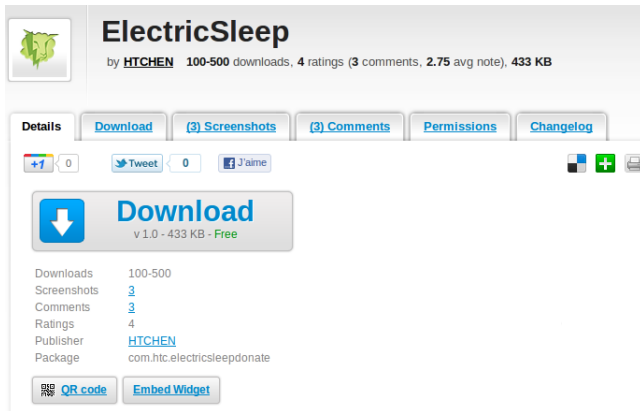
Features:

- \*records and saves your sleep cycle data.
- \*alarm clock gently wakes you during light sleep.
- \*analyzes sleep trends (debt, quality, duration, etc.)



# Android

Is it your application ? :)



**ElectricSleep**  
by [HTCHEN](#) 100-500 downloads, 4 ratings (3 comments, 2.75 avg note), 433 KB

**Details** | [Download](#) | [\(3\) Screenshots](#) | [\(3\) Comments](#) | [Permissions](#) | [Changelog](#)

[+1](#) 0 | [Tweet](#) 0 | [J'aime](#)

[Download](#)  
v 1.0 - 433 KB - **Free**

Downloads	100-500
Screenshots	<a href="#">3</a>
Comments	<a href="#">3</a>
Ratings	4
Publisher	<a href="#">HTCHEN</a>
Package	com.htc.electricsleepdonate

[QR code](#) | [Embed Widget](#)





# Android

## Is it your application ? :)

### Comments and ratings for ElectricSleep

by Jonathan on 16/11/2011



**MALWARE!** This version puts spam adverts in your notification bar! Look instead for the version that says "Jon Willis", that's the real one (and a great app).

by Jon on 12/10/2011



**Beware POSSIBLY MALWARE.** I am the original developer of ElectricSleep. This app is a repost of my app, with added permissions and no new features.

by Sun on 06/10/2011



Very detailed and user friendly tutorial! I can see that dev actually spent a lot of time perfecting this app. Will report back once I'm done testing.



# Android

## Is it your application ? :)

### ► HTCHEN



#### Pedometer

HTCHEN

This app can help to do exercise. It counts your steps, displays your pace, ap...

★★★★★

INSTALLER



#### 五子棋

HTCHEN

五子棋是一款休闲益智类的游戏，由于其规则简单，深受人们的喜爱，老少皆宜。本游戏分为人机对战和人人

★★★★★

INSTALLER



#### NinjaDash

HTCHEN

NinjaDash is a type of action game, which is operated by making use of a mobi...

★★★★★

INSTALLER



#### Bonfire

HTCHEN

A pile of burning bonfire, realistic effects, can give you warm in winter.

★★★★★

INSTALLER



#### Sudoku

HTCHEN

Simple and easy-to-use Sudoku. 4000 free Sudoku puzzles in multiple difficult...

INSTALLER



#### Replicalsland

HTCHEN

Fly, stomp, and roll your way through 40 challenging 2D side scrolling levels...

★★★★★

INSTALLER



#### Piano

HTCHEN

A simple piano application. Everyone can easily play the piano, even if you ne...

★★★★★

INSTALLER



#### Daily Money

HTCHEN

Can't handle your daily finance? Daily Money is here to help you. Daily Money...

INSTALLER



#### TippyTipper

HTCHEN

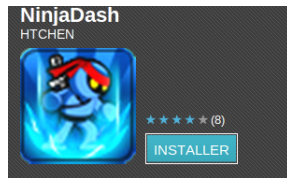
A simple Tip Calculator. \* Enter bill via custom keypad \* Select tip by side...

INSTALLER



# Android

## Is it your application ? :)



### Autres articles du même développeur



#### Pedometer

HTCHEN

★★★★★ (7)

Gratuit



#### 五子棋

HTCHEN

★★★★★ (2)

Gratuit



#### Bonfire

HTCHEN

★★★★★ (4)

Gratuit

### PRÉSENTATION

### AVIS DES UTILISATEURS

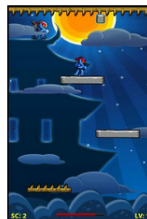
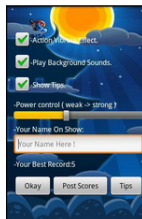
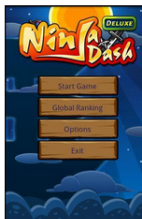
### NOUVEAUTÉS

### AUTORISATIONS

## Description

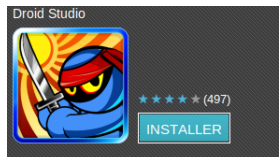
NinjaDash is a type of action game, which is operated by making use of a mobile phone's gravitational sensor. Tilt the phone left or right to control the ROLE moving left or right. The more angle in the tilt, the faster the ROLE moves. Avoid touching the spike or the ROLE will lose its lifespan. When the lifespan runs out, the game is over. You can post your records to our global ranking server. We all hope to see your name in the top 25!

## Captures d'écran de l'application



# Android

## Is it your application ? :)



### Autres articles du même développeur



#### Ninja au Démon 2

DROID STUDIO

★★★★★ (12 209)

Gratuit



#### Diable Ninja

DROID STUDIO

★★★★★ (6 239)

Gratuit



#### Diable Ninja (version bêta)

DROID STUDIO

★★★★★ (1 745)

Gratuit



#### Ninja Dash-Deluxe

DROID STUDIO

★★★★★ (1 840)

Gratuit

### PRÉSENTATION

### AVIS DES UTILISATEURS

### NOUVEAUTÉS

### AUTORISATIONS

## Description

Ninja Dash is an jump-and-run action game, In this fast paced ninja game, your goal is to dodge the approaching barriers, And there are various props to increase your running ability.

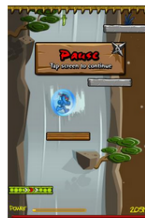
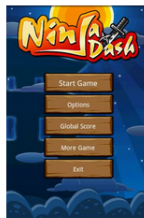
This the most addictive ninja jumping game in Android Market, enjoy it!

How to play:

- \* Tile to move left or right
- \* Caution: the falling darts & knives will hurt you!
- \* Power up: foods give Role powerful items for survival. such as Saiyan, lightning, and armor etc.

PLUS

## Captures d'écran de l'application



# Android

## Is it your application ? :)

### Applications Au moins 1 000 résultats



#### Daily Money

DENNIS CHEN / FINANCE

★★★★★ (2 558)

INSTALLER

daily-money, free and open source, daily expense tracker \*!!!Please read this note!!!\*  
\*Please post issues to Facebook page, I can't response you here\* \*Do you know th...



#### Journaux français et du monde

ANDROID APPS TEAM / ACTUALITÉS ET MAGAZINES

★★★★★ (1 141)

INSTALLER

Accédez facilement a vos sites de journaux préféré pour les nouvelles locales et dans le monde: Lisez les nouvelles majeur pour votre pays, ou changer de pays facilement...



#### Daily Money

HTCHEN / OUTILS

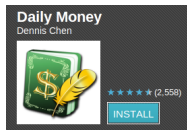
INSTALLER

Can't handle your daily finance? Daily Money is here to help you. Daily Money is great application for managing your expenses and incomes: • Tracking expenses and inc...



# Android

Is it your application ? :)



## Permissions

**THIS APPLICATION HAS ACCESS TO THE FOLLOWING:**

### NETWORK COMMUNICATION

#### FULL INTERNET ACCESS

Allows an application to create network sockets.

### STORAGE

#### MODIFY/DELETE USB STORAGE CONTENTS MODIFY/DELETE SD CARD CONTENTS

Allows an application to write to the USB storage. Allows an application to write to the SD card.

Show all

### NETWORK COMMUNICATION

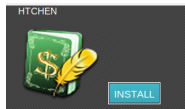
#### VIEW NETWORK STATE

Allows an application to view the state of all networks.



# Android

Is it your application ? :)



## Permissions

**THIS APPLICATION HAS ACCESS TO THE FOLLOWING:**

### **YOUR LOCATION**

#### **COARSE (NETWORK-BASED) LOCATION**

Access coarse location sources such as the cellular network database to determine an approximate device location, where available. Malicious applications can use this to determine approximately where you are.

#### **FINE (GPS) LOCATION**

Access fine location sources such as the Global Positioning System on the device, where available. Malicious applications can use this to determine where you are, and may consume additional battery power.

### **NETWORK COMMUNICATION**

#### **FULL INTERNET ACCESS**

Allows an application to create network sockets.

### **PHONE CALLS**

#### **READ PHONE STATE AND IDENTITY**



# Current section

Android

Analysis

Static Analysis

Visualization

Demos

Conclusion





# Android

## Reverse Engineering

- ▶ Reverse engineering tools like IDA Pro (not free), Baksmali (free), Androguard (free)
- ▶ Decompiler better than DED, jd-gui ...

## Plagiarism

- ▶ It is very time consuming and inefficient
- ▶  $\implies$  Automated approaches ?



# Outline

Android

Analysis

Static Analysis

Visualization

Demos

Conclusion



# Analysis

## Control Flow Graph

- ▶ In each method, you have a list of basic blocks
  - ▶ one entry point, meaning no code within it is the destination of a jump instruction anywhere in the program;
  - ▶ one exit point, meaning only the last instruction can cause the program to begin executing code in a different basic block.
- ▶ Modification of the control flow :
  - ▶ "if\*", "goto\*", "return\*", "packed\*", "sparse"
  - ▶ exceptions



# Permissions

## Where ?

- ▶ Useful to know where a specific permission is used in the application,
- ▶ You must search specific API in the bytecode,
- ▶ Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, David Wagner (UC Berkeley): create a permission map:
  - ▶ SEND\_SMS: sendTextMessage



# Permissions

## Where ?

```
In [2]: show_Permissions(dx)
READ_PHONE_STATE :
Lcom/flashp/FlashApplication; onCreate ()V (@onCreate-BB@0x0-0x24) ---> Landroid/telephony/TelephonyManager; getDeviceId ()Ljava/lang/String;
SEND_SMS :
Lcom/flashp/FlashService; sendSMS (Ljava/lang/String; Ljava/lang/String;)V (@sendSMS-BB@0x0-0x2)
---> Landroid/telephony/SmsManager; getDefault ()Landroid/telephony/SmsManager;
Lcom/flashp/FlashService; sendSMS (Ljava/lang/String; Ljava/lang/String;)V (@sendSMS-BB@0x0-0x14)
---> Landroid/telephony/SmsManager; sendTextMessage (Ljava/lang/String; Ljava/lang/String; Ljava/lang/String; Landroid/app/PendingIntent; Landroid/app/PendingIntent;)V
INTERNET :
Lcom/flashp/http/HttpClient; <init> ()V (@<init>-BB@0x0-0xb4) ---> Lorg/apache/http/impl/client/DefaultHttpClient; <init> (Lorg/apache/http/conn/ClientConnectionManager; Lorg/apache/http/params/HttpParams;)V
Lcom/flashp/http/HttpClient; getResponse (Lorg/apache/http/client/methods/HttpRequest; Ljava/lang/String; (@getResponse-BB@0x14-0x18) ---> Lorg/apache/http/impl/client/DefaultHttpClient; execute (Lorg/apache/http/client/methods/HttpRequest; Lorg/apache/http/HttpResponse;
```



# AndroidManifest.xml

What ?

- ▶ "Every application must have an AndroidManifest.xml file (with precisely that name) in its root directory",
- ▶ Essential information about the application :
  - ▶ activities, services, broadcast receivers,
  - ▶ permissions,
  - ▶ package name...
- ▶ XML file converted in a specific binary xml file.



# Analysis

## Signature

- ▶ Create a signature in order to identify a particular method in a set of methods (not exactly the same method, but also variants of this method),
- ▶ Based on a paper of Silvio Cesare: Fast Automated Unpacking and Classification of Malware,
- ▶ It's a simple grammar which used: Control Flow Graph, Fields, Packages, Strings and Exceptions.

```
Procedure ::= StatementList
StatementList ::= Statement | Statement StatementList
Statement ::= BasicBlock | Return | Goto | If | Field | Package | String | Exception
Return ::= 'R'
Goto ::= 'G'
If ::= 'I'
BasicBlock ::= 'B'
Field ::= 'F'0 | 'F'1
Package ::= 'P' PackageNew | 'P' PackageCall
PackageNew ::= 'C'
PackageCall ::= 'M'
PackageName ::= Epsilon | Id
String ::= 'S' Number | 'S' Id
Exception ::= Id
Number ::= \d+
Id ::= [a-zA-Z]\w+
```



# Analysis

## Signature

- ▶ Several signatures :
  - ▶ V0: no specific information about string, packages, fields,
  - ▶ V1: V0 + but with the size of strings,
  - ▶ V2: V0 + filtering android packages names,
  - ▶ V3: V0 + filtering java packages names,
  - ▶ V4: V0 + filtering android/java packages.





## Signature Example

```

0(0) const/4 v0, [# +0], {0} [ testMultipleLoops-BB@x2 ]

testMultipleLoops-BB@x2 :
1(2) const/16 v1, [#+ 50], {50}
2(6) if-lt v0, v1, [+ 15] [ testMultipleLoops-BB@0xa testMultipleLoops-BB@0x24 ]

testMultipleLoops-BB@0xa :
3(a) rem-int/lit8 v1, v0, [#+ 3]
4(e) if-eqz v1, [+ 14] [ testMultipleLoops-BB@0x12 testMultipleLoops-BB@0x2a ]

testMultipleLoops-BB@0x12 :
5(12) const/16 v1, [#+ 789], {789}
6(16) if-ge v0, v1, [+ 6] [ testMultipleLoops-BB@0x1a testMultipleLoops-BB@0x22 ]

testMultipleLoops-BB@0x1a :
7(1a) const/16 v1, [#+ 901], {901}
8(1e) if-gt v0, v1, [+ 9] [ testMultipleLoops-BB@0x22 testMultipleLoops-BB@0x30 ]

testMultipleLoops-BB@0x22 :
9(22) return-void

testMultipleLoops-BB@0x24 :
10(24) add-int/lit8 v0, v0, [#+ 2]
11(28) goto [+ -19] [ testMultipleLoops-BB@0x2 ]

testMultipleLoops-BB@0x2a :
12(2a) mul-int/lit8 v0, v0, [#+ 5]
13(2e) goto [+ -18] [ testMultipleLoops-BB@0xa ]

testMultipleLoops-BB@0x30 :
14(30) sget-object v1, [field@ 0 Java/lang/System; Ljava/io/PrintStream; out]
15(34) const-string v2, [string@ 335 'woo']
16(38) invoke-virtual v1, v2, [meth@ 7 Ljava/io/PrintStream; (Ljava/lang/String;) V pr
intln]
17(3e) goto [+ -22] [ testMultipleLoops-BB@0x12 ]

```

```
[TestsAndroguardTestLoops; testMultipleLoops ()\n->      B[B][B][B][B][B][B][R][B][G][B][G][FOSP1G]\n->      B[B][B][B][B][B][B][R][B][G][B][G][FOSP1G]\n->      B[B][B][B][B][B][B][R][B][G][B][G][FOSP3P1G]\n->      B[B][B][B][B][B][B][R][B][G][B][G][FOSP3P1G]\n->      B[B][B][B][B][B][B][R][B][G][B][G][B][FOSP1P[Ljava/io/PrintStream;println(Ljava/lang/String;)V\njGI\n->      B[B][B][B][B][B][B][R][B][G][B][G][B][FOSP1P[Ljava/io/PrintStream;println(Ljava/lang/String;)V\njGI
```

# Analysis

## Signature Example

```
Ltests/androguard/TestActivity; <init> ()V
-> : B[P1F1F1F1F1F1R]B[]
-> : B[P1F1F1F1F1F1R]B[]
-> : B[P1{Landroid/app/Activity;<init>()V}F1F1F1F1F1R]B[]
-> : B[P1F1F1F1F1F1R]B[]
-> : B[P1{Landroid/app/Activity;<init>()V}F1F1F1F1F1R]B[]

Ltests/androguard/TestActivity; <init> (D D)V
-> : B[P1F1F1F1F1F1R]B[]
-> : B[P1F1F1F1F1F1R]B[]
-> : B[P1{Landroid/app/Activity;<init>()V}F1F1F1F1F1R]B[]
-> : B[P1F1F1F1F1F1R]B[]
-> : B[P1{Landroid/app/Activity;<init>()V}F1F1F1F1F1R]B[]

Ltests/androguard/TestActivity; <init> (I I)V
-> : B[P1F1F1F1F1F1R]B[]
-> : B[P1F1F1F1F1F1R]B[]
-> : B[P1{Landroid/app/Activity;<init>()V}F1F1F1F1F1R]B[]
-> : B[P1F1F1F1F1F1R]B[]
-> : B[P1{Landroid/app/Activity;<init>()V}F1F1F1F1F1R]B[]
```



# Analysis

## Signatures Similarity

- ▶ How to know if two strings are similar ?

## Signatures Similarity

- ▶ Hamming distance,
- ▶ Levenshtein distance,
- ▶ Jaccard distance,
- ▶ Cosine similarity,
- ▶ Locality sensitive hashing,
- ▶ Normalized compression distance.



# Analysis

## NCD

- ▶ Designed to be an effective approximation of the noncomputable but universal Kolmogorov complexity between two strings.
- ▶ The NCD of two elements  $A$  and  $B$  is defined as  $d_{NCD}(A, B)$ . We can compute
  - ▶  $C(A)$  and  $L_A = L(C(A))$ ;
  - ▶  $C(B)$  and  $L_B = L(C(B))$ ;
  - ▶  $C(A|B)$  and  $L_{A|B} = L(C(A|B))$ ;
- ▶ *where  $A|B$  is the concatenation of  $A$  and  $B$ ,  $C$  is the compressor, and  $L$  is the length of a string.*



# Analysis

## NCD

- Then  $d_{NCD}(A, B)$  is defined by :

$$d_{NCD}(A, B) = \frac{L_{A|B} - \min(L_A, L_B)}{\max(L_A, L_B)}. \quad (1)$$



# Analysis

## NCD

- ▶ A compressor  $C$  is normal if the following four axioms are satisfied up to an additive  $O(\log n)$ , where  $n$  is the maximal binary length of the elements involved in the inequalities:
  1. Idempotency:  $C(xx) = C(x)$ , and  $C(\varepsilon) = 0$ , where  $\varepsilon$  is the empty string.
  2. Monotonicity:  $C(xy) \geq C(x)$ .
  3. Symmetry:  $C(xy) = C(yx)$ .
  4. Distributivity:  $C(xy) + C(z) \leq C(xz) + C(yz)$ .



# Analysis

## NCD

- ▶ If you take three elements:
  - ▶ X ("HELLO WORLD") and the length of the compression  $Y = C(X) = 6$ ,
  - ▶ X' ("HELLO WOORLD") and the length of the compression of  $Y' = C(X') = 7$ ,
  - ▶ X'' ("HI !!!") and the length of the compression of  $Y'' = C(X'') = 3$ .
- ▶ the compression of  $C(XX')$  will be similar to  $C(X)$  whereas the compression of  $C(XX'')$  will not be similar to  $C(X)$ .



# Analysis

## NCD

- The compression rate is not a determining factor for the choice of the compressor if it complies with the following rules:
  1. C respects the four inequalities,
  2. C(x) is calculated within an acceptable amount of time.





# Analysis

## NCD: compressor ?

- ▶ Compressor: compressed datas, time (s)
- ▶ LZMA: 900, 1.45565796
- ▶ XZ: 1824, 0.72005010
- ▶ ZLIB: 894, 0.00037599
- ▶ BZIP2: 1294, 0.00088286
- ▶ Snappy: 1208, 0.00010705



# Analysis

## NCD: Snappy compressor

- ▶ Snappy is a compression/decompression library (Google),
- ▶ It does not aim for maximum compression, or compatibility with any other compression library; instead, it aims for very high speeds and reasonable compression,
- ▶ Based on text by Zeev Tarantov,
- ▶ LZ77-type compressor with a fixed, byte-oriented encoding,
- ▶ Fast: Compression speeds at 250 MB/sec and beyond, with no assembler code,
- ▶ Stable: Over the last few years, Snappy has compressed and decompressed petabytes of data in Google's production environment.



# Analysis

## Similarity

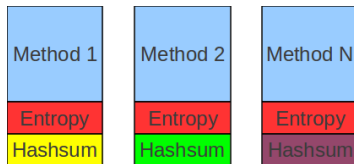
- ▶ Identify identical methods,
- ▶ Identify exact/similar methods,
- ▶ Identify new methods,
- ▶ Identify deleted methods.



# Analysis

Similarity: attributes associated with a method

- ▶ the entropy, based on the raw binary data,
- ▶ a buffer which represents the sequence of instructions, with useless information removed from it,
- ▶ a unique checksum (or hash) based on the previous buffer,
- ▶ a signature.



# Analysis

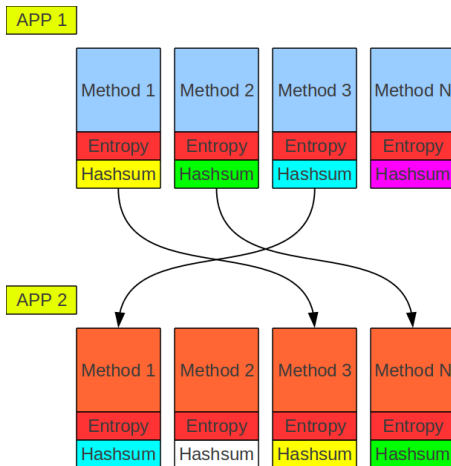
## Signature Example

```
desnos@destiny:~/androguard$ ./androcsign.py -i signatures/droiddream.sign
['B{SP0{Ljava/util/Formatter;}P1{Ljava/util/Formatter;<init>()V}SSP2P2P0{Ljava/lang/StringBuilde
r;}F0P1{Ljava/lang/String;valueOf(Ljava/lang/Object;)Ljava/lang/String;}P1{Ljava/lang/StringBuil
der;<init>(Ljava/lang/String;)V}SP1{Ljava/lang/StringBuilder;append(Ljava/lang/String;)Ljava/lang
/StringBuilder;}F0P1{Ljava/lang/StringBuilder;append(I)Ljava/lang/StringBuilder;}P1{Ljava/lang/
StringBuilder;toString()Ljava/lang/String;}P1{Ljava/util/Formatter;format(Ljava/lang/String; [Lj
ava/lang/Object;)Ljava/util/Formatter;}P1{Ljava/util/Formatter;toString()Ljava/lang/String;}P1{L
java/lang/String;getBytes() [B}P2P0{Ljava/net/URL;}P1{Ljava/net/URL;<init>(Ljava/lang/String;)V}P
1{Ljava/net/URL;openConnection()Ljava/net/URLConnection;}P1{Ljava/net/URLConnection;setDoOut
put(Z)V}P1{Ljava/net/URLConnection;setDoInput(Z)V}SP1{Ljava/net/URLConnection;setRequest
Method(Ljava/lang/String;)V}P1{Ljava/net/URLConnection;getOutputStream()Ljava/io/OutputStrea
m;}P0{Ljava/io/ByteArrayInputStream;}P1{Ljava/io/ByteArrayInputStream;<init>([B)V}B[P1{Ljava/io
/ByteArrayInputStream;read([B I I)I}I}B[P1{Ljava/io/ByteArrayInputStream;close()V}P1{Ljava/io/Out
putStream;close()V}P0{Ljava/io/ByteArrayOutputStream;}P1{Ljava/io/ByteArrayOutputStream;<init>([
)V}P0{Ljava/io/BufferedInputStream;}P1{Ljava/net/URLConnection;getInputStream()Ljava/io/Inpu
tStream;}P1{Ljava/io/BufferedInputStream;<init>(Ljava/io/InputStream;)V}B[P1{Ljava/io/InputStream;
read([B I I)I}I}B[P1{Ljava/io/InputStream;close()V}P1{Ljava/io/ByteArrayOutputStream;size()I}
I}B[SP1{Landroid/content/Context;getSharedPreferences(Ljava/lang/String; I)Landroid/content/Shar
edPreferences;}P1{Landroid/content/SharedPreferences;edit()Landroid/content/SharedPreferences$Ed
itor;}SP1{Landroid/content/SharedPreferences$Editor;putInt(Ljava/lang/String; I)Landroid/content
/SharedPreferences$Editor;}P1{Landroid/content/SharedPreferences$Editor;commit()Z}B[R]B[P1{Lj
ava/io/OutputStream;write([B I I)V}P1{Ljava/io/OutputStream;flush()V}G]B[P1{Ljava/io/ByteArrayOutp
utStream;write([B I I)V}G]', 5.0286870002746582, 4.4915299415588379, 4.9674844741821289, 4.94683
02726745605, 0.0]
```



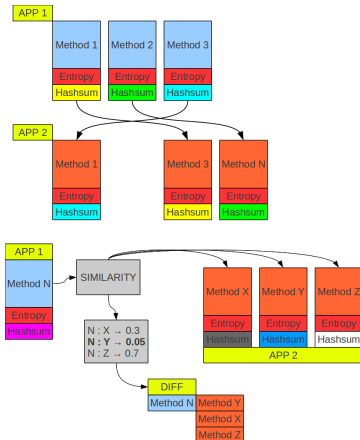
# Analysis

Similarity: remove identical methods by using hash



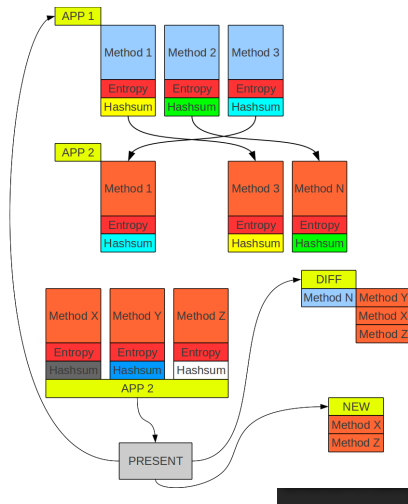
# Analysis

Similarity: find exact/similar methods between two applications



# Analysis

Similarity: Identify new methods between two applications





# Analysis

## Plagiarism/Rip-Off indicator

- ▶ By using previous algorithms:
  - ▶ we can calculate an indicator (between 0.0 to 100.0) to indicate whether the application has been stolen
- ▶ 0.0 to a perfect identical method,
- ▶ value of the NCD for a partial identical method,
- ▶ value of the NCD for the general information of the application (strings, constants, etc.).



# Analysis

## Plagiarism/Rip-Off indicator: two different applications

```
desnos@destiny:~/androguard$ ./androsim.py -i
examples/obfu/classes_tc.dex apks/classes.dex
DIFF METHODS : 3
NEW METHODS : 199
MATCH METHODS : 0
DELETE METHODS : 4
[0.99816107749938965, 1.0, 1.0, 1.0]
0.0459730625153
```



# Analysis

## Plagiarism/Rip-Off indicator: identical applications

DIFF METHODS : 0

NEW METHODS : 0

MATCH METHODS : 14

DELETE METHODS : 0

[0.08235294371843338, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

99.4509803752



# Analysis

Plagiarism/Rip-Off indicator: quite identical applications

DIFF METHODS : 1

NEW METHODS : 0

MATCH METHODS : 12

DELETE METHODS : 0

[0.14427860081195831, 0.095238097012042999, 0.0,  
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
0.0, 0.0]  
98.2891664441



# Analysis

## Plagiarism/Rip-Off indicator: stolen application

```
desnos@destiny:~/androguard$ ./androsim.py -i apks/  
HolyFuckingBiblev11-market-militia-.apk apks/  
holyfuckingbible.apk  
DIFF METHODS : 1  
NEW METHODS : 81  
MATCH METHODS : 72  
DELETE METHODS : 0  
[0.8460613489151001, 0.091269843280315399, 0.0, 0.0,  
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]  
98.7333362268
```



# Analysis

## Plagiarism/Rip-Off indicator: The Wars



## Plagiarism/Rip-Off indicator: The Wars

[illegible]

# Analysis

## Plagiarism/Rip-Off indicator: DailyMoney(HTCHEN)

- ▶ Timothy Armstrong (Kaspersky Lab):
  - ▶ Pay-Per-Install library was added to the original code,
  - ▶ The library comes as part of an SDK from a company called AirPush.

The screenshot shows the main interface of the 'Earn 10x More on Android' app. At the top, there's a blue header with the Android robot logo and the text 'Android™ Developers: Earn 10x More on Android™'. Below this is a 'Get Started' button. To the right, a circular badge says 'The Future of Mobile Advertising' and 'AndroidAuthority.com'. Below the header, a section titled 'Today's Averages' displays three statistics: 'CPM: \$18.32', 'Fill Rate: 84.41%', and 'Developers: 61,696'. At the bottom, there are three smartphone screens illustrating different ad formats: 'Signup Ads' (showing a list of offers), 'Push Notification Ads' (showing a notification about a new car), and 'Icon Ads' (showing an app icon with a speech bubble). The bottom of the screen has labels for each: 'Signup Ads', 'Push Notification Ads', and 'Icon Ads'.





## Plagiarism/Rip-Off indicator: DailyMoney(HTCHEN)

- ▶ Timothy Armstrong (Kaspersky Lab):
  - ▶ different types of advertisements to end users

- ◆ How much money can I make? What CPM's ?

Airpush developers earn CPM's in the \$6 - \$40 range depending on country mix and the number of ad formats they choose to you. Most importantly however, those CPM's are earned both on active and inactive users.

As a result, most developers are shocked at the actual earnings increase when transitioning from Admob / Inmobi /etc to Airpush. Developers can easily go from making \$30/day on an app, to making \$500 - \$2,000 /day from the same app. If you think that sounds crazy, try us out on one of your smaller apps!

- ▶ The developer is paid every 1.000 impressions (CPM: Cost Per Mille, "It is used in marketing as a benchmark to calculate the relative cost of an advertising campaign or an ad message in a given medium").



## Analysis

## Plagiarism/Rip-Off indicator: DailyMoney(HTCHEN)

[illegible]

## Plagiarism/Rip-Off indicator: DailyMoney(HTCHEN)

### NEW METHODS :

```
Lcom/airpush/android/Airpush; a (Landroid/content/Context; J)V 184
Lcom/airpush/android/Airpush; a (Lcom/airpush/android/Airpush;)V 276
Lcom/airpush/android/Airpush; a (Landroid/content/Context; Ljava/lang/String; Ljava/lang/String;
  Z Z I Z)V 128
Lcom/airpush/android/DeliveryReceiver; onReceive (Landroid/content/Context; Landroid/content/Int
ent;)V 946
Lcom/airpush/android/HttpPostData; a (Ljava/lang/String; Landroid/content/Context;)Ljava/lang/St
ring; 126
Lcom/airpush/android/HttpPostData; a (Ljava/util/List; Z Landroid/content/Context;)Lorg/apache/h
ttp/HttpEntity; 110
Lcom/airpush/android/MessageReceiver; a (J)V 193
Lcom/airpush/android/MessageReceiver; onReceive (Landroid/content/Context; Landroid/content/Inte
nt;)V 184
Lcom/airpush/android/PushAds; onCreate (Landroid/os/Bundle;)V 952
Lcom/airpush/android/PushService; a (J)V 172
Lcom/airpush/android/PushService; a (J)V 162
Lcom/airpush/android/PushService; a (Ljava/lang/String;)V 129
Lcom/airpush/android/PushService; b (J)V 1472
Lcom/airpush/android/PushService; b (Ljava/lang/String;)V 1037
Lcom/airpush/android/PushService; onStart (Landroid/content/Intent; I)V 1377
Lcom/airpush/android/SetPreferences; a (Landroid/content/Context;)Ljava/util/List; 496
Lcom/airpush/android/SetPreferences; a (Landroid/content/Context; Ljava/lang/String; Ljava/lang/
String; Z Z I Z)V 503
```



# Analysis

## Evaluation of Android obfuscators

- ▶ Problem: transformation of the source code in bytecode,
- ▶ Android developers use obfuscators frequently such as proguard or dашo to prevent the reverse engineering of their software,
- ▶ It can be easily reversed by using a classical decompiler like jad, jd-gui or dаva, with varying degrees of reliability,
- ▶ Moreover virtual machines do not allow code modification on the fly (but dynamic code loading) and it is a real problem for classical packers.



# Analysis

## Evaluation of Android obfuscators

- ▶ the obfuscator can use several techniques to protect a Java/Android application:
  1. change names of classes, methods, fields,
  2. modify the control flow,
  3. code optimization,
  4. dynamic code loading,
  5. change instructions with metamorphic technique.



# Analysis

## Evaluation of Android obfuscators

- ▶ Blackbox evaluation with our previous similarity algorithms
- ▶ If this distance is close to 100 then the obfuscator did a poor job ...



# Analysis

## Evaluation of Android obfuscators

```
desnos@destiny:~/androguard$ ./androsim.py -i
examples/obfu/classes_tc.dex examples/obfu/
classes_tc_proguard.dex
DIFF METHODS : 7
NEW METHODS : 4
MATCH METHODS : 0
DELETE METHODS : 0
[0.47394958138465881, 0.040816325694322586,
0.059999998658895493, 0.040816325694322586,
0.059999998658895493, 0.13333334028720856,
0.040816325694322586, 0.095238097012042999]
88.1878750864
desnos@destiny:~/androguard$ ./androsim.py -i
examples/obfu/classes_tc.dex examples/obfu/
classes_tc_dasho.dex
DIFF METHODS : 2
NEW METHODS : 0
MATCH METHODS : 10
DELETE METHODS : 0
[0.50084036588668823, 0.13114753365516663,
0.1428571492433548, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0]
94.0396534709
```



# Analysis

## Malware

- ▶ We can extract automatically new methods: it is the case of an injected malware in the Android official or unofficial markets,
- ▶ The malware writer injects his "evil" code in the application and propagates the new application in different markets.
- ▶ It is possible to isolate the malware quickly if we know the original application, which is an easy task because the malware writer does not generally modify it.





# Analysis

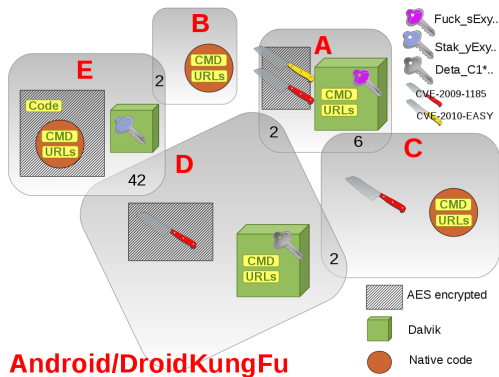
## Malware

```
desnos@destiny:~/androguard$ ./androsim.py -i apks/
com.swampy.sexpos_162.apk apks/com.swampy.sexpos
.apk-GEINIMI-INFECTED.apk
DIFF METHODS : 0
NEW METHODS : 51
MATCH METHODS : 218
DELETE METHODS : 0
[1.0, 0.0, [...]]
99.5433789954
desnos@destiny:~/androguard$ ./androsim.py -i apks/
TAT-LWP-Mod-Dandelion-orig.apk apks/TAT-LWP-Mod-
Dandelion.apk
DIFF METHODS : 0
NEW METHODS : 31
MATCH METHODS : 18
DELETE METHODS : 0
[0.68480598926544189, 0.0, [...]]
96.3957579512
```



# Analysis

## Axelle Apvrille(Fortinet): Clarifying Android DroidKungFu variants



**Android/DroidKungFu**



# Analysis

## Diffing

- ▶ Calculate the differences between two versions of an application to identify modifications:
  - ▶ security bugfix,
  - ▶ reverse engineering.
- ▶ The idea is to detect classical modifications in a method including:
  - ▶ modification of codes in a basic block,
  - ▶ addition of new basic blocks.
- ▶ Bindiff, patchdiff2, ...



# Analysis

## Diffing

- ▶ Isomorphism problem: graph comparing
- ▶ Find identical/similar methods in order to extract modifications of instructions from basic blocks
  - ▶ Identification of identical basic blocks by using NCD,
  - ▶ Extraction of added/removed instructions by using the longest common subsequence algorithm.



# Analysis

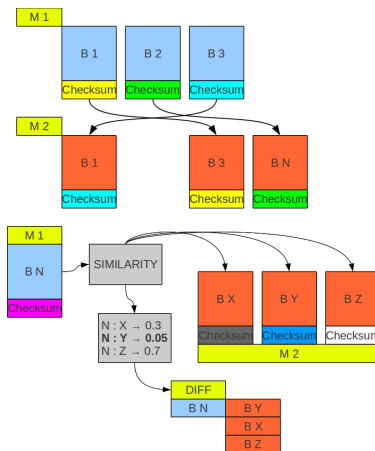
## Diffing: Identification of basic blocks

- It is the similarity algorithms but it is just a different level of granularity



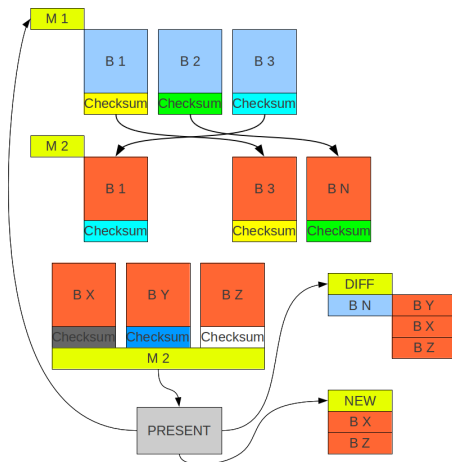
# Analysis

Diffing: Find exactly/partially the same basic blocks between two methods



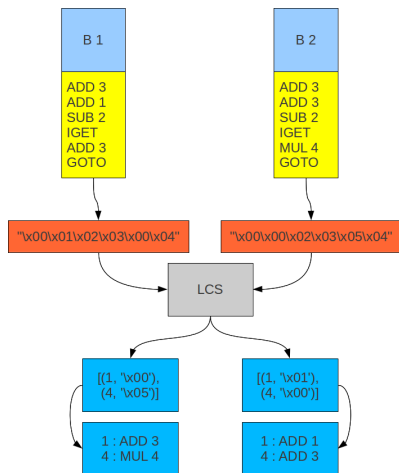
## Analysis

## Diffing: Find new basic blocks between two methods



# Analysis

Diffing: Find added/removed instructions from a basic block





# Analysis

## Diffing: Skype android application

- ▶ The 15th April 2011, AndroidPolice released a new security vulnerability in Skype (version 1.0.0.831) for Android,
- ▶ This vulnerability exposes the users' name, phone number, and chat logs to all installed applications,
- ▶ The security bug is very simple, it is an incorrect usage of permissions to open files,
- ▶ A few days after this vulnerability, Skype release a new version (1.0.0.983) which fixed this security bug.



# Analysis

## Diffing: Skype android application

- ▶ exactly identical: 8038,
- ▶ partialy identical: 165,
- ▶ new: 14,
- ▶ delete: 7.



## Diffing: Skype android application

- ▶ searching methods related to file permissions (by using the Java API or directly with chmod program)
- ▶ most of them are related to simple constant modification but we can identify a method really close to another one (with the same name) which manipulate files:
  - ▶ Lcom/skype/ipc/SkypeKitRunner; run ()V with  
Lcom/skype/ipc/SkypeKitRunner; run ()V 0.269383959472



# Analysis

## Diffing: Skype android application

- ▶ This method has four modified basic blocks, but only three basic blocks merit further investigation.



# Analysis

## Diffing: Skype android application

- An integer value (it is the operating mode) of the method *openFileOutput*, *public abstract FileOutputStream openFileOutput (String name, int mode)* has been changed from 3 to 0

```
DIFF run-BB@0x316 :  
[...]  
220(324) const-string v7 , [string@ 2998 'csf']  
221(328) + const/4 v8 , [#+ 0] , {0}  
222(328) - const/4 v8 , [#+ 3] , {3}  
223(328) invoke-virtual v5 , v7 , v8 , [meth@ 120  
    Landroid/content/Context; (Ljava/lang/String; I)  
    Ljava/io/FileOutputStream; openFileOutput]  
[...]
```



# Analysis

## Diffing: Skype android application

- In another basic block, the first argument of chmod has been changed from 777 to 750

```
DIFF run-BB@0x348 :
229(346) invoke-static [meth@ 5805 Ljava/lang/
        Runtime; () Ljava/lang/Runtime; getRuntime]
230(34c) move-result-object v2
231(34e) new-instance v4 , [type@ 899 Ljava/lang/
        StringBuilder;]
232(352) invoke-direct v4 , [meth@ 5848 Ljava/lang/
        StringBuilder; () V <init>]
233(358) + const-string v5 , [string@ 2921 'chmod
        750 ']
234(358) - const-string v5 , [string@ 2904 'chmod
        777 ']
235(358) invoke-virtual v4 , v5 , [meth@ 5855 Ljava/
        lang/StringBuilder; (Ljava/lang/String;) Ljava/
        lang/StringBuilder; append]
236(35e) move-result-object v4
237(360) invoke-virtual v3 , [meth@ 5719 Ljava/io/
        File; () Ljava/lang/String; getCanonicalPath]
```



# Analysis

## Diffing: Skype android application

- ▶ And in the last modified basic block, there is a new call to a new method which fixes all files in the context directory of the application:
  - ▶ `Lcom/skype/ipc/SkypeKitRunner; ([Ljava/io/File;) V fixPermissions]`
- ▶ which fixes all permissions (patch permissions from the previous version) to:
  - ▶ `RWX` — — for a directory,
  - ▶ `RW-` — — for a file.

```
417(5c8) + move-object/from16 v0 , v19
418(5c8) invoke-virtual v4 , v3 , v2 , v5 , [meth@
      5804 Ljava/lang/Runtime; (Ljava/lang/String; [
      Ljava/lang/String; Ljava/io/File;) Ljava/lang/
      Process; exec]
419(5ce) + move-object v1 , v4
420(5ce) move-result-object v2
421(5d0) + invoke-direct v0 , v1 , [meth@ 1923 Lcom
      /skype/ipc/SkypeKitRunner; ([Ljava/io/File;) V
      fixPermissions]
```



# Analysis

## Decompilation

- ▶ Useful for static source code analysis.
- ▶ Current ways to decompile are not efficient enough.
  - ▶ Source code unreadable
  - ▶ Doesn't compile back
  - ▶ Decompilation fail





# Analysis

```
public static boolean isPackageInstalled(Context paramContext, String paramString)
{
    List localList = paramContext.getPackageManager().getInstalledPackages(0);
    int i = 0;
    while (true)
    {
        int j = localList.size();
        if (i >= j);
        for (int k = 0; ; k = 1)
        {
            return k;
            if (!((PackageInfo)localList.get(i)).packageName.equals(paramString))
                break;
        }
        i += 1;
    }
}
```



# Analysis

```
public void run()
{
    byte[] arrayOfByte = new byte[4096];
    int i = 0;
    while (true)
    {
        if (i < 0);
        String str;
        while (true)
        {
            return;
            try
            {
                i = this.val$in.read(arrayOfByte);
                str = new String(arrayOfByte, 0, i);
                if (!str.contains("Forked"))
                    break label183;
            }
        }
    }
}
```



# Analysis

```
// ERROR //  
private String getMountPoint(InputStream paramInputStream, String paramString)  
{  
    // Byte code:  
    // 0: aconst_null  
    // 1: astore_3  
    // 2: aconst_null  
    // 3: astore 4  
    // 5: new 132 java/io/InputStreamReader  
    // 8: dup  
    // 9: aload_1  
    // 10: invokespecial 135 java/io/InputStreamReader:<init> (Ljava/io/InputStream;)V  
    // 13: astore 5  
    // 15: new 137 java/io/BufferedReader  
    // 18: dup  
    // 19: aload 5  
    // 21: sipush 1024  
    // 24: invokespecial 140 java/io/BufferedReader:<init> (Ljava/io/Reader;I)V  
    // 27: astore 6  
    // 29: aload 6
```



# Analysis

- ▶ Bytecode contains high level information:
  - ▶ operators are typed
  - ▶ different functions calls depending on the method “type”
  - ▶ ...
- ▶ Code rewriting is not allowed.
  - ▶ Once the code is analysed, we know it will not change during execution.



# Analysis

## Decompilation

Different phases (optimizations/compilation) :

- ▶ Intermediate representation
- ▶ Semantic analysis
- ▶ CFG generation
- ▶ Dataflow analysis
- ▶ Control flow analysis
- ▶ Code generation



# Analysis

## Decompilation

- ▶ Intermediate representation
  - ▶ Bytecode is already a kind of IR
  - ▶ We “abstract” instructions with python objects
  - ▶ Kind of SSA (Static Single Assignment)
- ▶ Semantic analysis
- ▶ CFG generation
- ▶ Dataflow analysis
- ▶ Control flow analysis
- ▶ Code generation



# Analysis

## Decompilation

- ▶ Intermediate representation
- ▶ Semantic analysis
  - ▶ Data type propagation
- ▶ CFG generation
- ▶ Dataflow analysis
- ▶ Control flow analysis
- ▶ Code generation



# Analysis

## Decompilation

- ▶ Intermediate representation
- ▶ Semantic analysis
- ▶ CFG generation
  - ▶ method divided into basic blocks
  - ▶ each node of the graph represent a basic block
- ▶ Dataflow analysis
- ▶ Control flow analysis
- ▶ Code generation





# Analysis

## Decompilation

- ▶ Intermediate representation
- ▶ Semantic analysis
- ▶ CFG generation
- ▶ Dataflow analysis
  - ▶ refine the IR of the method
- ▶ Control flow analysis
- ▶ Code generation



# Analysis

## Decompilation

- ▶ Intermediate representation
- ▶ Semantic analysis
- ▶ CFG generation
- ▶ Dataflow analysis
- ▶ Control flow analysis
  - ▶ detect the high level constructs of the method
- ▶ Code generation



# Analysis

## Decompilation

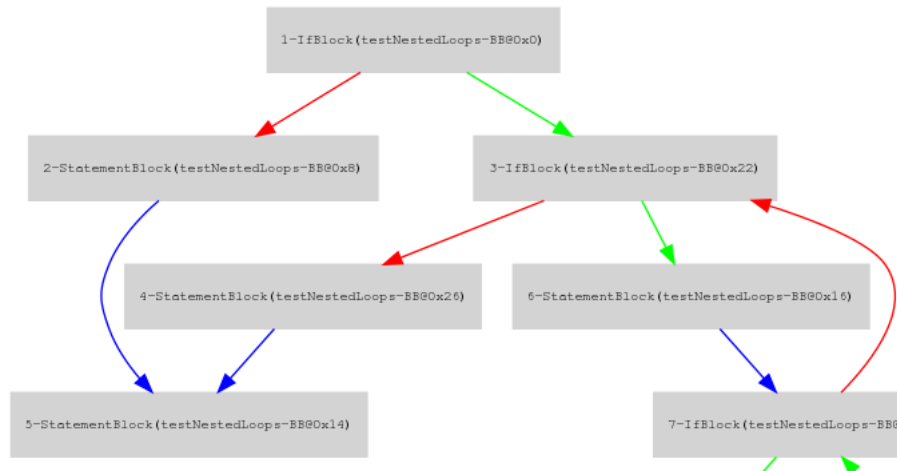
- ▶ Intermediate representation
- ▶ Semantic analysis
- ▶ CFG generation
- ▶ Dataflow analysis
- ▶ Control flow analysis
- ▶ Code generation
  - ▶ write the source by traversing the AST



# Analysis

## Control flow analysis

- ▶ Number nodes of graph in reverse post-order:
  - ▶ number given when visited for the last time

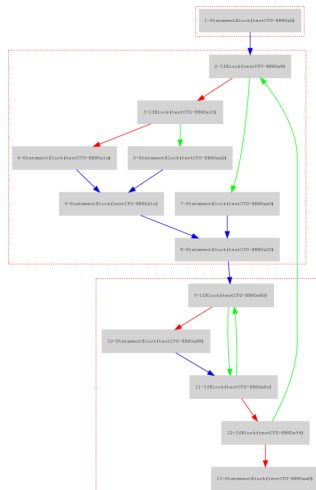
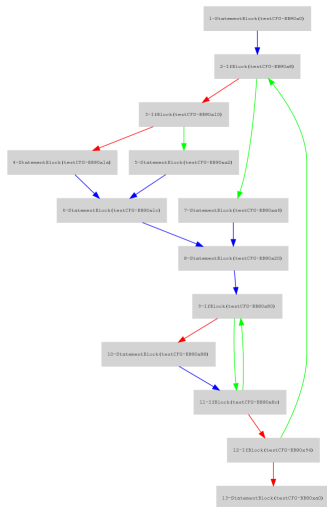


## Control flow analysis

- ▶ We want to identify structures
- ▶ Build intervals to detect loops
- ▶ Nodes are flagged accordingly
- ▶ Switch and Conditionnal structures detected by traversing the graph in reverse (from last to first node)



# Analysis



# Analysis

- ▶ Need to find the next element of a structure
  - ▶ E.g: next of a conditionnal structure is the first common node of both branches
    - ▶ Special case with short circuit
- ▶ Write the code of the nodes by traversing it
  - ▶ nodes are flagged : type of node, of loop, head of loop, ...



# Outline

Android

Analysis

Static Analysis

Visualization

Demos

Conclusion





# Application

## Control Flow Graph

- ▶ Export like a classical graphviz picture,
- ▶ Export the CFG in Cytoscape.

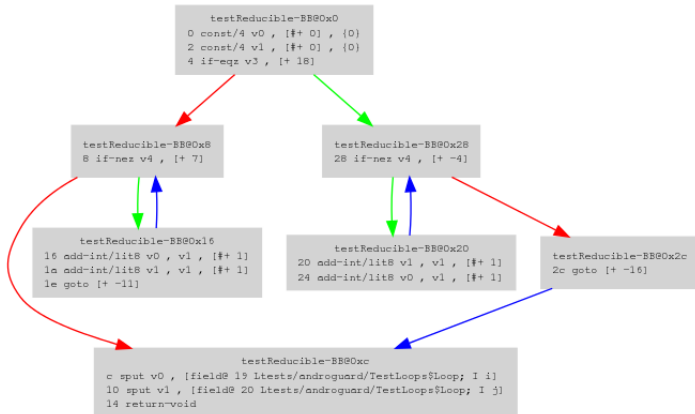


## Control Flow Graph



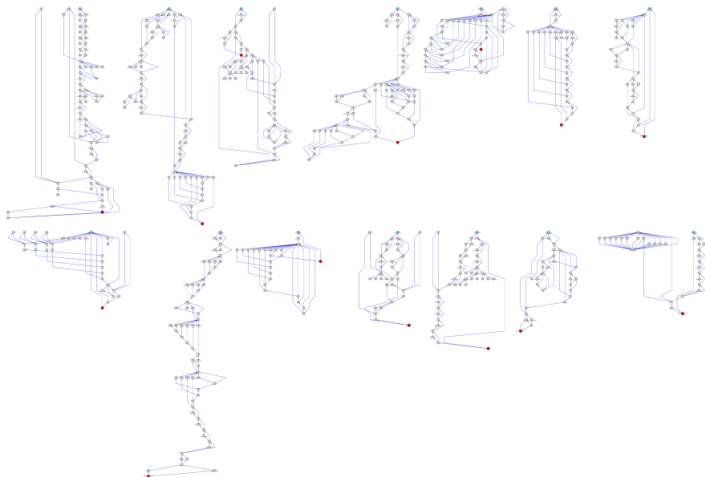
## Application

## Control Flow Graph



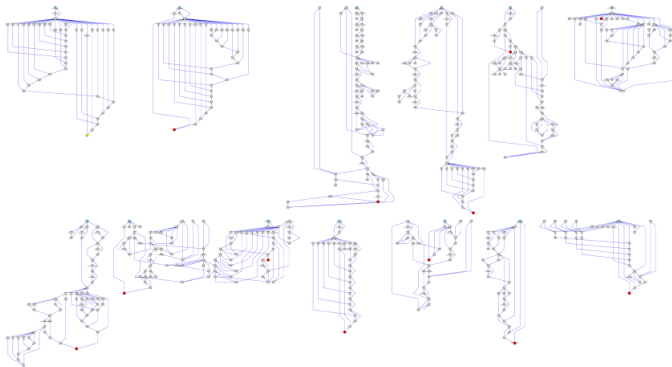
# Application

## Control Flow Graph



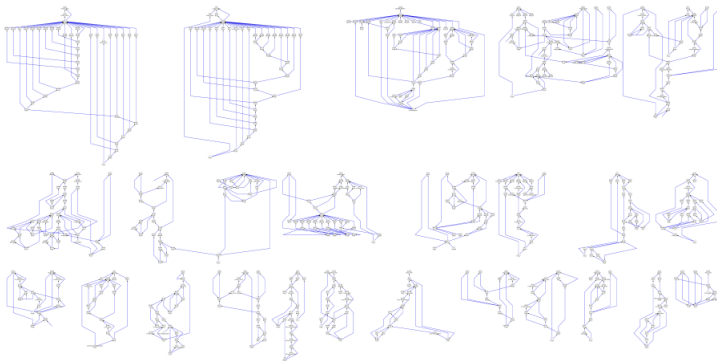
## Application

## Control Flow Graph



# Application

## Control Flow Graph



# Application

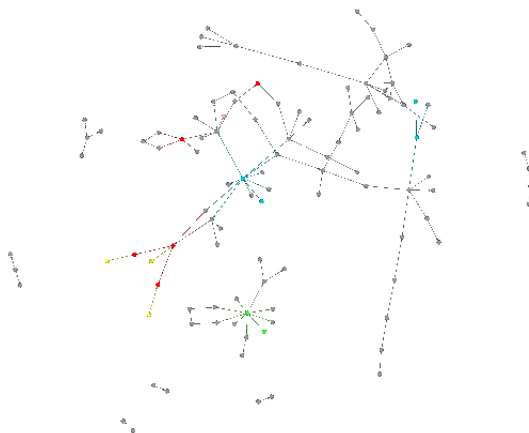
## Methods Call Graph

- ▶ Export methods call graph in .gexf format:
  - ▶ Information about each node
  - ▶ Add specific nodes (permissions, activities, ...)



# Application

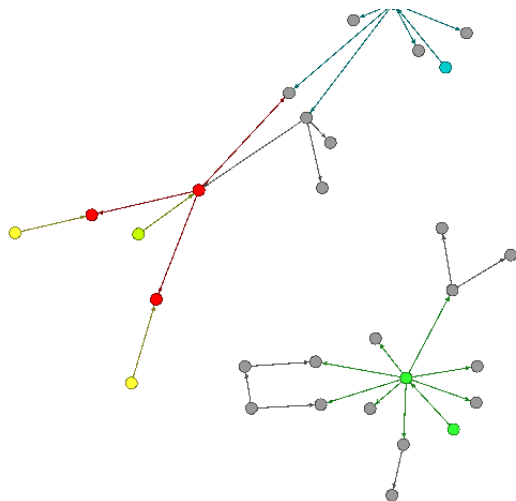
## Methods Call Graph





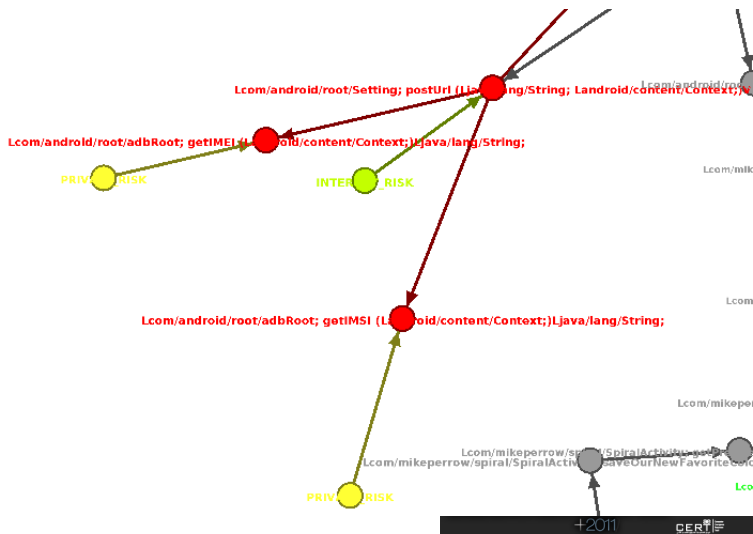
# Application

## Methods Call Graph



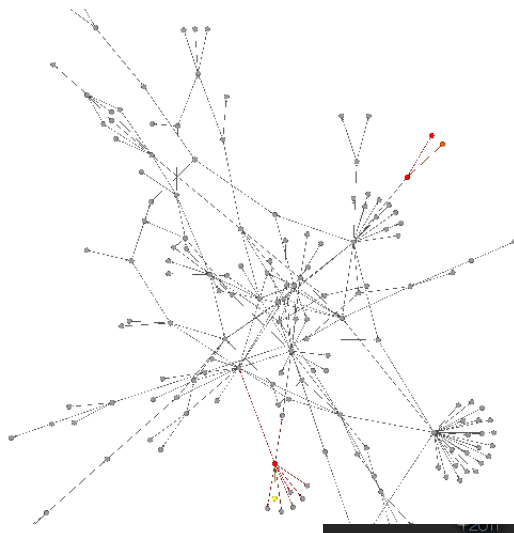
# Application

## Methods Call Graph



# Application

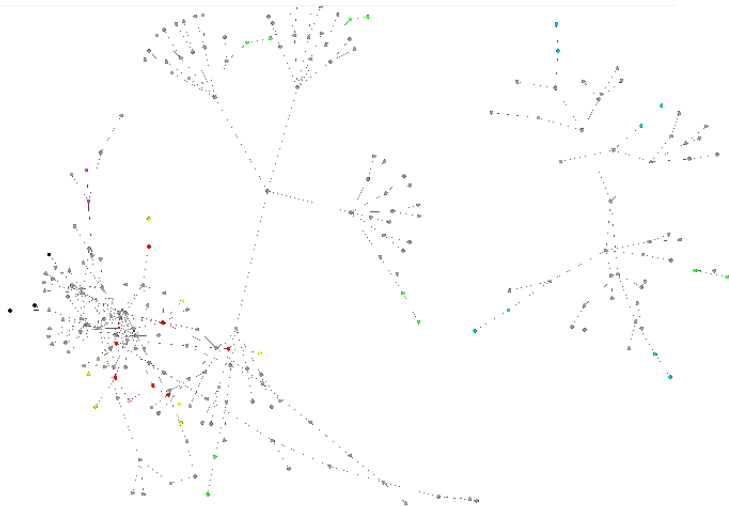
## Methods Call Graph





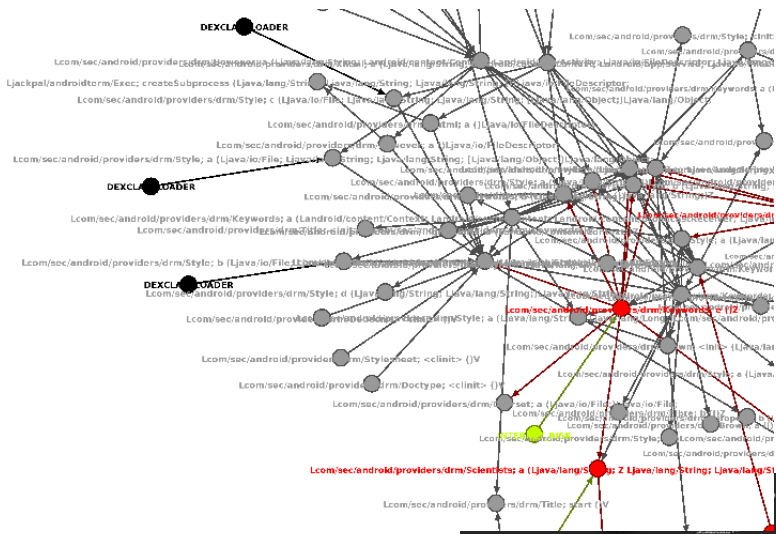
# Application

## Methods Call Graph



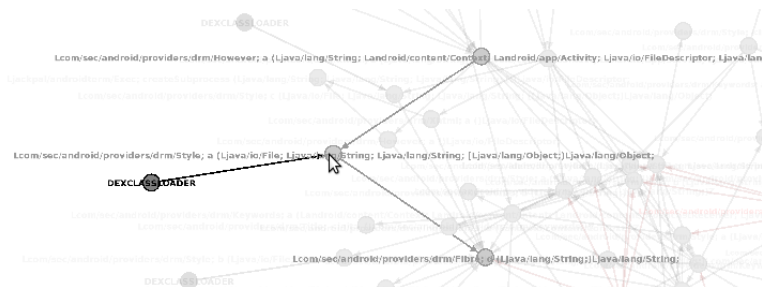
# Application

## Methods Call Graph



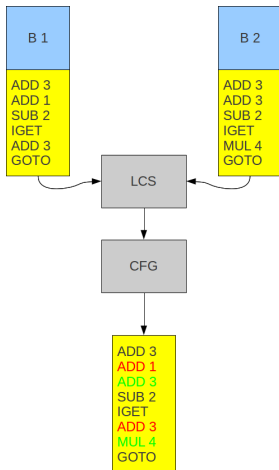
# Application

## Methods Call Graph



# Diffing

- Aureliano Calvo: Showing differences between disassembled functions





# Diffing

```
DIFF run-BB00x348 :
50(b4) invoke-static [meth@ 5805 Ljava/lang/Runtime; () Ljava/lang/Runtime; getRuntime]
51(ba) move-result-object v2
52(bc) new-instance v4 , [type@ 899 Ljava/lang/StringBuilder;]
53(c0) invoke-direct v4 , [meth@ 5848 Ljava/lang/StringBuilder; () V <init>]
54(c6) const-string v5 , [string@ 2921 'chmod 750 ']
55(c6) const-string v5 , [string@ 2904 'chmod 777 ']
56(c6) invoke-virtual v4 , v5 , [meth@ 5855 Ljava/lang/StringBuilder; (Ljava/lang/String;) Ljava/lang/StringBuilder; append]
57(cc) move-result-object v4
58(ce) invoke-virtual v3 , [meth@ 5719 Ljava/io/File; () Ljava/lang/String; getCanonicalPath]
59(d4) move-result-object v5
60(d6) invoke-virtual v4 , v5 , [meth@ 5855 Ljava/lang/StringBuilder; (Ljava/lang/String;) Ljava/lang/StringBuilder; append]
61(dc) move-result-object v4
62(de) invoke-virtual v4 , [meth@ 5857 Ljava/lang/StringBuilder; () Ljava/lang/String; toString]
63(e4) move-result-object v4
64(e6) invoke-virtual v2 , v4 , [meth@ 5803 Ljava/lang/Runtime; (Ljava/lang/String;) Ljava/lang/Process; exec]
65(ec) move-result-object v2
```



# Current section

Android

Analysis

Static Analysis

Visualization

Demos

Conclusion



# Usage of the reversing tools



# Usage of the decompiler



# Current section

Android

Analysis

Static Analysis

Visualization

Demos

Conclusion



# Conclusion

## Androguard

- ▶ LGPL framework/tools<sup>1</sup>
- ▶ Python/C(++)
- ▶ You're Welcome !

---

<sup>1</sup><http://code.google.com/p/androguard/>



# Conclusion

## Future Works

- ▶ Improve plagiarism algorithm,
- ▶ Emulation of android bytecodes,
- ▶ Data tainting,
- ▶ Optimization phases of the decompiler.



# Conclusion

!

- ▶ Thanks to Blackhat
- ▶ Questions ?

