

Android reverse-engineering

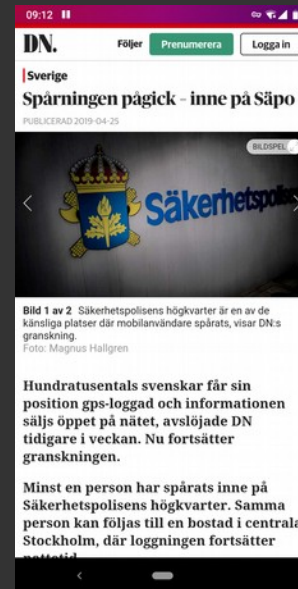
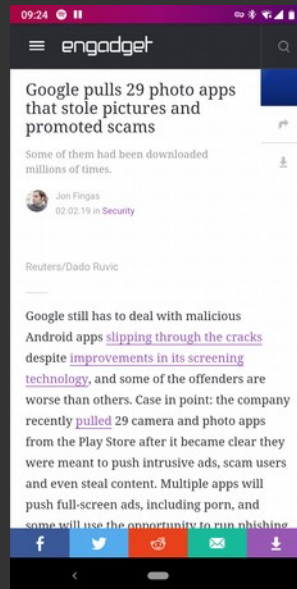
review and modify closed source apps

Magnus, Tech-Evangelist @ 0xFF

Why?

0xFF

review/audit
patch/modify
exploit



Disclaimer

APK

0xFF

Manifest
Resources
Code

Manifest
Resources
Code

Activities
Services
Content providers
Intents

demo

0xFF

Java/Kotlin > DEX > ~~JIT/Dalvik~~ > ARM/x86/etc

Java/Kotlin > DEX > OAT > ART > ARM/x86/etc

DEX > Smali

```
.class public LHelloWorld;

.super Ljava/lang/Object;

.method public static main([Ljava/lang/String;)V
    .registers 2

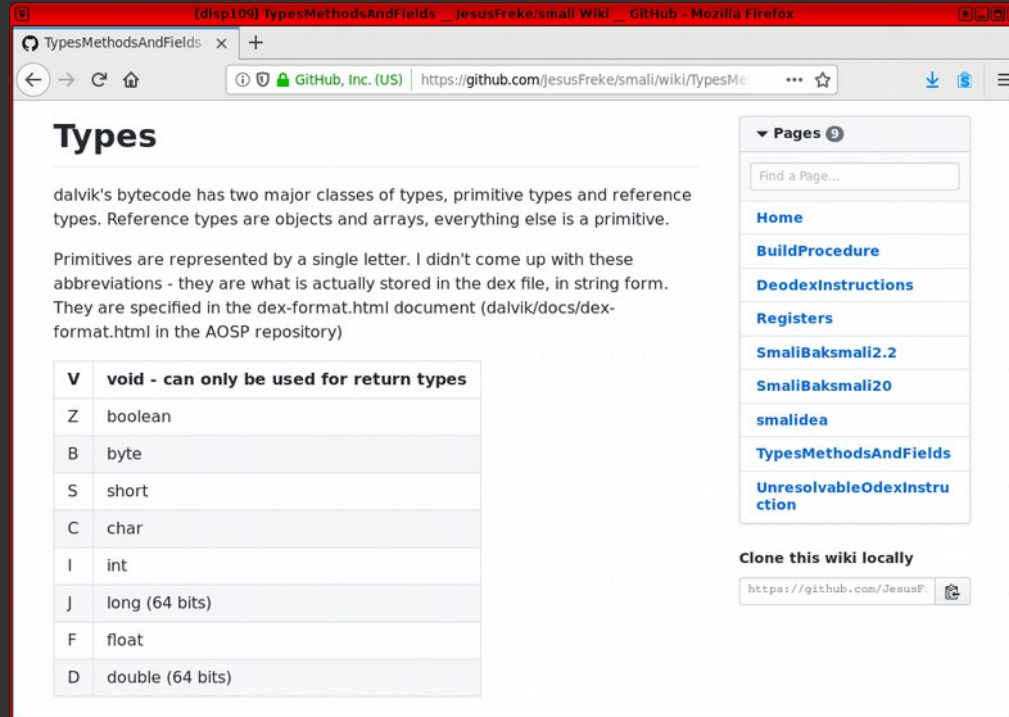
    sget-object v0, Ljava/lang/System;-.>out:Ljava/io/PrintStream;

    const-string v1, "Hello World!"

    invoke-virtual {v0, v1}, Ljava/io/PrintStream;-.>println(Ljava/lang/String;)V

    return-void
.end method
```

Data-types



The screenshot shows a web browser window displaying a GitHub Wiki page. The page title is "Types". The content explains that Dalvik's bytecode has two major classes of types: primitive types and reference types. It lists several primitive types in a table and provides a sidebar with navigation links.

dalvik's bytecode has two major classes of types, primitive types and reference types. Reference types are objects and arrays, everything else is a primitive.

Primitives are represented by a single letter. I didn't come up with these abbreviations - they are what is actually stored in the dex file, in string form. They are specified in the dex-format.html document (dalvik/docs/dex-format.html in the AOSP repository)

V	void - can only be used for return types
Z	boolean
B	byte
S	short
C	char
I	int
J	long (64 bits)
F	float
D	double (64 bits)

Pages

- Home
- BuildProcedure
- DeodexInstructions
- Registers
- SmaliBaksmali2.2
- SmaliBaksmali20
- smalidea
- TypesMethodsAndFields**
- UnresolvableOdexInstruction

Clone this wiki locally

<https://github.com/JesusF>

<https://github.com/JesusFreke/smali/wiki/TypesMethodsAndFields>

Instructions

The screenshot shows a web browser displaying the Dalvik bytecode documentation page. The page title is "6d: sput-short". The left sidebar contains a navigation menu with items like Overview, Improvements, Bytecode Format, Dex Format, Instruction Formats, Constraints, Configuration, Garbage Collection, and JIT Compilation. The main content area is divided into columns: the first column lists instruction codes (6e..72, 35c) and their formats (e.g., `invoke-kind (vC, vD, vE, vF, vG), meth@BBBB`); the second column lists parameters (A: argument word count, B: method reference index, C..G: argument registers); and the third column provides detailed explanations of each instruction's behavior and restrictions. A right sidebar contains a "Contents" section with links to various parts of the documentation.

Code	Format	Parameters	Description
6e..72	<code>invoke-kind (vC, vD, vE, vF, vG), meth@BBBB</code>	A: argument word count (4 bits)	Call the indicated method. The result (if any) may be stored with an appropriate <code>move-result*</code> variant as the immediately subsequent instruction.
6e	<code>invoke-virtual</code>	B: method reference index (16 bits)	<code>invoke-virtual</code> is used to invoke a normal virtual method (a method that is not <code>private</code> , <code>static</code> , or <code>final</code> , and is also not a constructor).
6f	<code>invoke-super</code>	C..G: argument registers (4 bits each)	When the <code>method_id</code> references a method of a non-interface class, <code>invoke-super</code> is used to invoke the closest superclass's virtual method (as opposed to the one with the same <code>method_id</code> in the calling class). The same method restrictions hold as for <code>invoke-virtual</code> .
70	<code>invoke-direct</code>		In Dex files version 037 or later, if the <code>method_id</code> refers to an interface method, <code>invoke-super</code> is used to invoke the most specific, non-overridden version of that method defined on that interface. The same method restrictions hold as for <code>invoke-virtual</code> . In Dex files prior to version 037, having an interface <code>method_id</code> is illegal and undefined.
71	<code>invoke-static</code>		<code>invoke-direct</code> is used to invoke a non- <code>static</code> direct method (that is, an instance method that is by its nature non-overridable, namely either a <code>private</code> instance method or a constructor).
72	<code>invoke-interface</code>		<code>invoke-static</code> is used to invoke a <code>static</code> method (which is always considered a direct method).

<https://source.android.com/devices/tech/dalvik/dalvik-bytecode>

Smali registers

v0, v1, v2... - local registers

p0, p1, p2... - method argument alias

(all 32-bit, so how to pass a 64-bit Long?)

```
.class public LHelloWorld;

.super Ljava/lang/Object;

.method public static main([Ljava/lang/String;)V
    .registers 2

    sget-object v0, Ljava/lang/System; ->out:Ljava/io/PrintStream;

    const-string v1, "Hello World!"

    invoke-virtual {v0, v1}, Ljava/io/PrintStream; ->println(Ljava/lang/String;)V

    return-void
.end method
```

Obfuscation

```
Ball.getColor() == a.a()
```

demo

0xFF

debugging smali

0xFF

grep is your friend

grep is your friend

```
grep -inr facebook.com --include=*.smali
```

-i ignores character case

-n display line numbers

-r recursive, search sub folders

--include=*.smali only search files matching

--color=always add coloring

grep is your friend – Trackers

facebook *google.com* *firebase*

urbanairship *crashlytics* *bugfender*

*track** *analytic** *ads*

grep is your friend – Privacy intrusive API calls

QueryIntentActivities *getRunningAppProcesses*

ActivityManager *PackageManager* *WifiManager*

SensorManager *BluetoothAdapter*

Address *LocationManager*

TelephonyManager *AdvertisingIdClient*

grep is your friend – File I/O

file read write

directory sdcard

document

grep is your friend – Net I/O

http http:

connect socket uri address

post .com/.net loadUrl

grep is your friend – Scary stuff

loadLibrary native

install

addJavaScriptInterface

demo

0xFF

OWASP Mobile Top 10

Exploiting GoatDroid

Tools used in this talk

(free and open source)

ADB

ApkTool

uber-apk-signer

ApkStudio

Idea

Ideasmali

Profiling tools

(root required)

Frida

Introspsy

Xposed

Automated tools

Mobile Security Framework (MobSF)

Quick Android Review Kit (QARK)

Drozer

Resources

Smali/Smalidea

<https://github.com/JesusFreke/smali/>

Dalvik instructions

<https://source.android.com/devices/tech/dalvik/dalvik-bytecode>

ADB

<https://developer.android.com/studio/releases/platform-tools>

ApkStudio

<https://github.com/vaibhavpandeyvpz/apkstudio>

OWASP Mobile Top 10

https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10

Thank you!
magnus@0xFF.se
@0xFFse <https://0xff.se/>