Disassembling Dalvik Bytecode

...

Alain Leon

Background

What is Android?

Android is an operating system by Google that uses a Linux kernel and runs its applications on a VM, formerly known as **Dalvik**

The programs that run on Android are packaged and distributed as **APK** files

Inside each APK file, there is an executable **DEX** file which is what actually gets run when the program starts

Android has the largest installed base of all operating systems of any kind

What is Dalvik?

It's a VM but it's **not** the Java VM

Register-based VM made more efficient when running on battery-powered, relatively low CPU/RAM smartphones

You write Java source that compiles to Java bytecode which then gets translated to Dalvik bytecode

Successor is Android Runtime (ART), introduced in KitKat (4.4+), completely replaced Dalvik in Lollipop (5.0+), which compiles-on-install rather than JIT

What is an APK?

Android Package

This is what you download and install from the Google Play store

It's really just a zip file containing an app

Holds the app's assets and Dalvik bytecode (in .dex or .odex format)

```
~/tmp ➤ unzip kh3.apk
Archive:  kh3.apk
  inflating: META-INF/MANIFEST.MF
  inflating: META-INF/CERT.SF
  inflating: META-INF/CERT.DSA
  inflating: AndroidManifest.xml
 extracting: assets/misc.mp4
 extracting: assets/misc.png
 extracting: assets/op_movie.mp4
  inflating: assets/sdkbox_config.json
 extracting: assets/tutorial_movie.mp4
  inflating: res/color/common_signin_btn_text_dark.xml
  inflating: res/color/common_signin_btn_text_light.xml
  inflating: res/color/wallet_primary_text_holo_light.xml
  inflating: res/color/wallet_secondary_text_holo_dark.xml
```

What is bytecode?

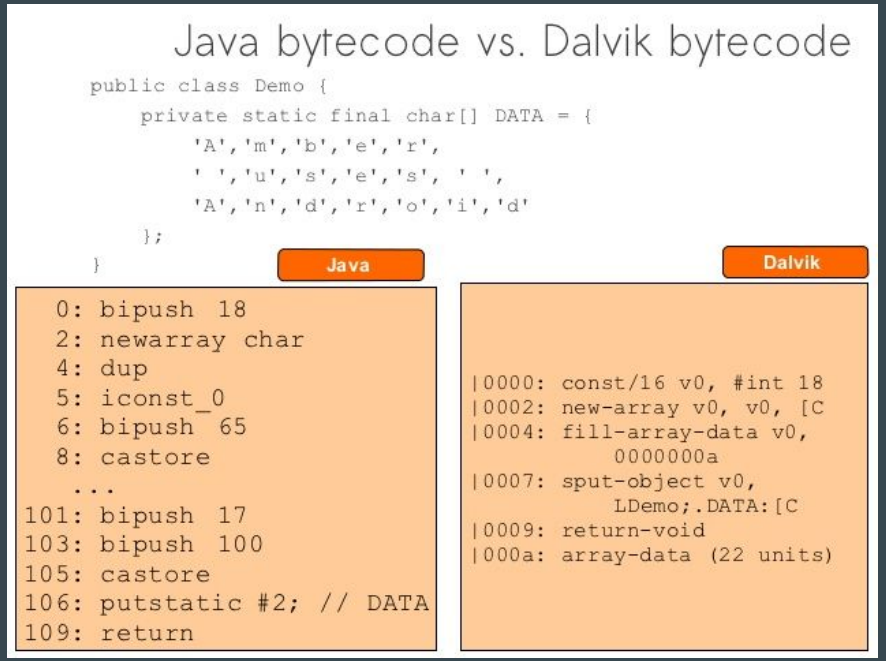
Not machine code

DEX = Dalvik Executable

Intermediate found in Java .class files and Dalvik .dex files

Translated between .dex and .class using the dx tool

Machine code is only created at runtime by the Just-In-Time (JIT) compiler



What is JIT compilation?

Mix between traditional ahead-of-time compiling and interpreting

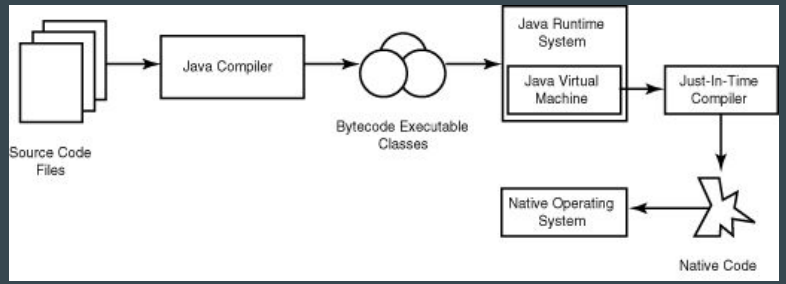
Machine code is generated during runtime

Combines the speed of compiled code with the flexibility of interpretation

At the cost of overhead of an interpreter + the additional overhead of compiling

Allows for adaptive optimization such as dynamic recompilation

Think re.compile() from Python



What is the Android NDK?

Android Native Development Kit

A set of tools that allow you to leverage C and C++ code in your Android apps

Uses the Java Native Interface (JNI) to expose Java calls to underlying system

Used by Cocos2d-x, game development tools written in C++

Cocos is compiled as a shared library and shipped inside the APK

Hacking at the Surface Level

Use a Macro to “Bot” the Game

Was the goal of my last talk

Use macros or scripts to automate some repeatable circuit to gain in-game currencies all day every day

Prone to errors

Slow, human level gain

Too Bad It’s Not Really That Cool

Hacking at the REST Level

Wireshark

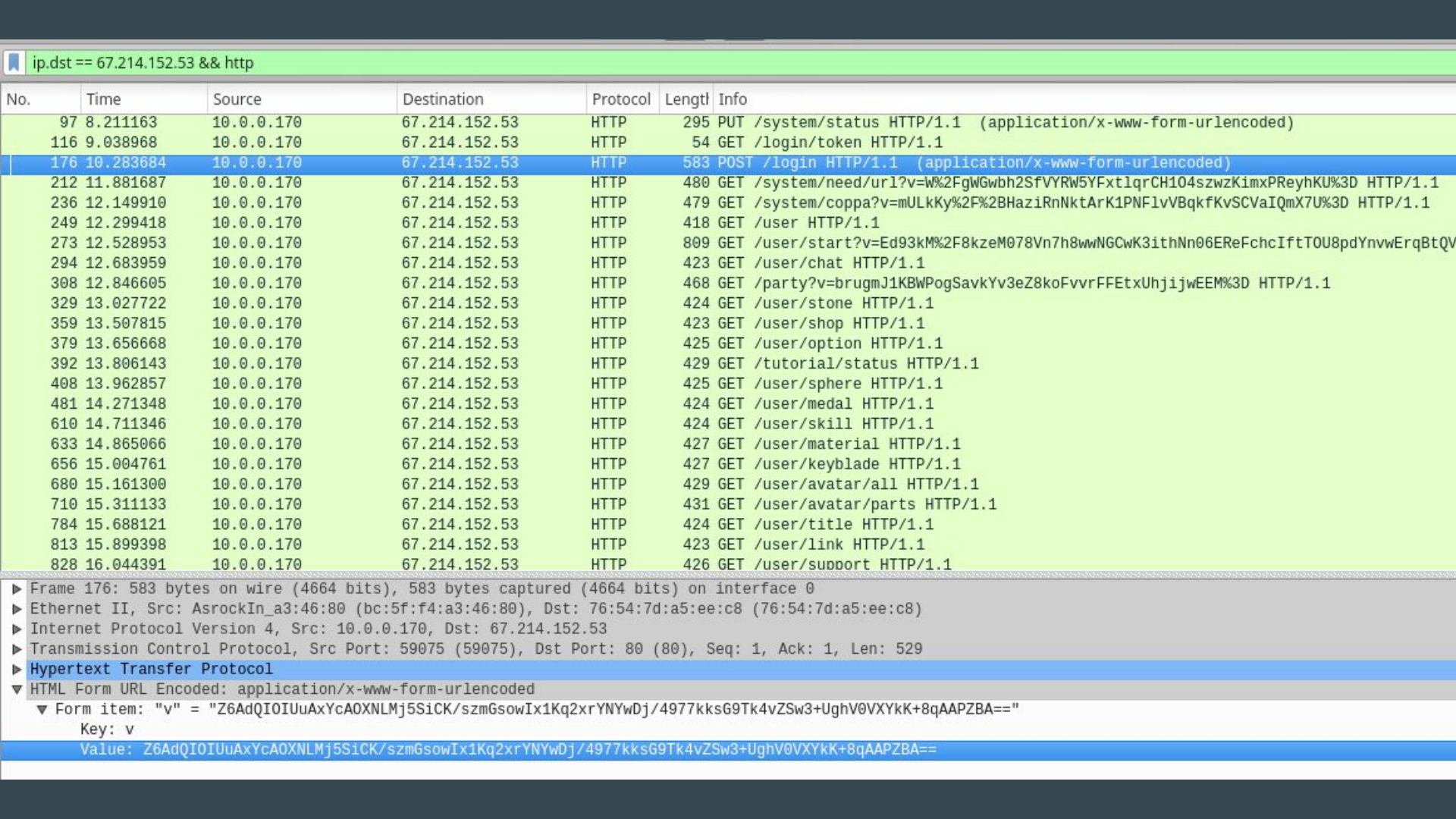
Sniff the traffic to and from an Android emulator

Make a malicious imposter client

Replay the get/put/posts using curl or python

Fail: Google Play Services uses OAuth 2.0

Sends ephemeral Base64-URL-encoded token


ip.dst == 67.214.152.53 && http
No. Time Source Destination Protocol Lengtl Info
97 8.211163 10.0.0.170 67.214.152.53 HTTP 295 PUT /system/status HTTP/1.1 (application/x-www-form-urlencoded)
116 9.038968 10.0.0.170 67.214.152.53 HTTP 54 GET /login/token HTTP/1.1
176 10.283684 10.0.0.170 67.214.152.53 HTTP 583 POST /login HTTP/1.1 (application/x-www-form-urlencoded)
212 11.881687 10.0.0.170 67.214.152.53 HTTP 480 GET /system/need/url?v=W%2FgWGwbh2SfVYRW5YFxtlqrCH1O4szwzKimxPReyhKU%3D HTTP/1.1
236 12.149910 10.0.0.170 67.214.152.53 HTTP 479 GET /system/coppa?v=mULkKy%2F%2BHaziRnNktArK1PNFlvVBqkfKvSCVaIQmX7U%3D HTTP/1.1
249 12.299418 10.0.0.170 67.214.152.53 HTTP 418 GET /user HTTP/1.1
273 12.528953 10.0.0.170 67.214.152.53 HTTP 809 GET /user/start?v=Ed93kM%2F8kzeM078Vn7h8wwNGCwK3ithNn06EReFchcIftTOU8pdYnvwErqBtQV
294 12.683959 10.0.0.170 67.214.152.53 HTTP 423 GET /user/chat HTTP/1.1
308 12.846605 10.0.0.170 67.214.152.53 HTTP 468 GET /party?v=brugmJ1KBWPogSavkYv3eZ8koFvvrFFEtxUhjijwEEM%3D HTTP/1.1
329 13.027722 10.0.0.170 67.214.152.53 HTTP 424 GET /user/stone HTTP/1.1
359 13.507815 10.0.0.170 67.214.152.53 HTTP 423 GET /user/shop HTTP/1.1
379 13.656668 10.0.0.170 67.214.152.53 HTTP 425 GET /user/option HTTP/1.1
392 13.806143 10.0.0.170 67.214.152.53 HTTP 429 GET /tutorial/status HTTP/1.1
408 13.962857 10.0.0.170 67.214.152.53 HTTP 425 GET /user/sphere HTTP/1.1
481 14.271348 10.0.0.170 67.214.152.53 HTTP 424 GET /user/medal HTTP/1.1
610 14.711346 10.0.0.170 67.214.152.53 HTTP 424 GET /user/skill HTTP/1.1
633 14.865066 10.0.0.170 67.214.152.53 HTTP 427 GET /user/material HTTP/1.1
656 15.004761 10.0.0.170 67.214.152.53 HTTP 427 GET /user/keyblade HTTP/1.1
680 15.161300 10.0.0.170 67.214.152.53 HTTP 429 GET /user/avatar/all HTTP/1.1
710 15.311133 10.0.0.170 67.214.152.53 HTTP 431 GET /user/avatar/parts HTTP/1.1
784 15.688121 10.0.0.170 67.214.152.53 HTTP 424 GET /user/title HTTP/1.1
813 15.899398 10.0.0.170 67.214.152.53 HTTP 423 GET /user/link HTTP/1.1
828 16.044391 10.0.0.170 67.214.152.53 HTTP 426 GET /user/support HTTP/1.1
Frame 176: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface 0
Ethernet II, Src: AsrockIn_a3:46:80 (bc:5f:f4:a3:46:80), Dst: 76:54:7d:a5:ee:c8 (76:54:7d:a5:ee:c8)
Internet Protocol Version 4, Src: 10.0.0.170, Dst: 67.214.152.53
Transmission Control Protocol, Src Port: 59075 (59075), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 529
Hypertext Transfer Protocol
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "v" = "Z6AdQIOIUuAxYcAOXNLMj5SiCK/szmGsowIx1Kq2xrYNYwDj/4977kksG9Tk4vZSw3+UghV0VXYkK+8qAAPZBA=="
Key: v
Value: Z6AdQIOIUuAxYcAOXNLMj5SiCK/szmGsowIx1Kq2xrYNYwDj/4977kksG9Tk4vZSw3+UghV0VXYkK+8qAAPZBA==

Hacking at the APK/DEX level

Get the APK

Find on Google Play and use that URL at an APK Downloader website or

Enable USB Debugging, install Android SDK, connect your smartphone and:

```
adb shell pm list packages | grep khux

adb shell pm path com.square_enix.android_googleplay.khuxww

adb pull /data/app/com.square_enix.android_googleplay.khuxww-1/base.apk
```

DEX Bytecode Disassembling (Baksmaling)

Two ways, recommend doing both:

Directly: Convert to bytecode to a readable format (Baksmali, Jasmine, etc.)

```
apktool d -f “khux.apk” -o smali
```

Indirectly: Convert to Java first, then use Java’s decompiling tools

dex2jar -> Java Decompiler (JD-Core, JD-GUI, etc.)

Smali Dalvik Bytecode Representation

```
.method public abstract zza(Lcom/google/android/gms/common/api/zza$zza;)Lcom/google/android/gms/common/api/zza$zza;
    .annotation system Ldalvik/annotation/Signature;
        value = {
            "<A::",
            "Lcom/google/android/gms/common/api/Api$zza;",
            "R::",
            "Lcom/google/android/gms/common/api/Result;",
            "T:",
            "Lcom/google/android/gms/common/api/zza$zza",
            "<TR;TA;>;>(TT;)TT;"
        }
    .end annotation
.end method

.method public abstract zza(Lcom/google/android/gms/common/api/Api;)Z
    .annotation system Ldalvik/annotation/Signature;
        value = {
            "(",
            "Lcom/google/android/gms/common/api/Api",
            "<*>;)Z"
        }
    .end annotation
.end method
```

Apply Changes

Change variables, convert to hex first!

```
const/16 v0, 9bff
```

Output variables to the Android log

```
const-string v0, "grep_for_this_breh:"
invoke-static {v0, p1}, Landroid/util/Log;->e(Ljava/lang/String;Ljava/lang/String;)I
```

APK Reassembling

apktool b -f smali/ -o khux_rekt.apk

jarsigner (Android SDK) - sign the apk with your own keystore or..

https://github.com/appium/sign

```
java -jar sign.jar modded.apk
```

zipalign (Android SDK) - (optional) ensures that all uncompressed data starts with a particular alignment relative to the start of the file, reducing app's RAM footprint

```
zipalign 4 modded.s.apk aligned.apk
```

Reinstall the APK

Uninstall the original APK if it's still on the device

Install the modded APK

```
adb install aligned.apk
```

Disable or uninstall Facebook if you're having problems with Facebook login

Watch the logs

```
adb logcat | grep grep_for_this_breh
```

Hacking at the Shared Object Level

Shared Object Analysis

libcocos2dcpp.so was the only meaningful difference

When diff tells you “Binary files differ”, you can convert to hex and try again.

```
xxd hacked.so > hacked.hex
```

```
vimdiff hacked.hex unhacked.hex
```

You can also try a byte-for-byte comparison

```
cmp -l file1.so file2.so
```

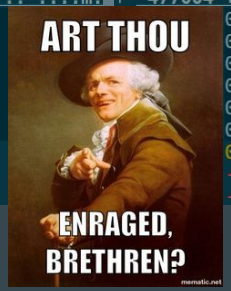
This prints out the line number of the changes and their differences in octal

hackedso.hex | notso.hex

```
+      1 +--476946 lines: 00000000: 7f45 4c46 0101 0100 0000 0000 0000 0000  .ELF....
  476947 00747120: 10bc 08e0 0d97 01b4 10bc 0d9f f86a 0028  ...........j.(
  476948 00747130: 01d0 46f1 4df5 786b 0028 01d0 46f1 48f5  ..F.M.xk.(..F.H.
  476949 00747140: 1fa8 daf4 6cea 22a8 daf4 80ea 10b4 01bc  ....l.".........
  476950 00747150: c6f4 eced e7e7 e6e7 e5e7 e4e7 e3e7 e2e7  ................
  476951 00747160: e1e7 e0e7 dfe7 dee7 dde7 dce7 dbe7 dae7  ................
  476952 00747170: d9e7 d8e7 01b4 10bc e2e7 c6f4 e4ed c046  ...............F
  476953 00747180: f8cb ffff 0040 1c46 6623 8900 ff20 4043  .....@.Ff#... @C
  476954 00747190: 7047 16ab 1393 04b4 80bc 1091 01b4 40bc  pG............@.
  476955 007471a0: 0a96 9b48 7844 9b49 0818 0068 0068 e062  ...HxD.I...h.h.b
  476956 007471b0: ad20 8000 3a58 1020 6946 0860 5520 c000  . ..:X. iF.`U ..
  476957 007471c0: 3818 1032 1ba9 0423 c7f4 44e8 a069 0f90  8..2...#..D..i..
  476958 007471d0: 0121 1191 0025 0995 40b4 01bc 20b4 04bc  .!...%..@... ...
  476959 007471e0: 80b4 08bc daf4 3eea 0e90 5920 c000 3e18  ......>...Y ..>.
  476960 007471f0: 3858 0368 1ba8 40b4 02bc 20b4 04bc 9847  8X.h..@... ....G
+  476961 +-- 30 lines: 00747200: 206a 0028 0dd1 3068 0368 1ba8 0322 40b4   j.(..0h.h..
  476991 007473e0: 002e 10d0 40b4 01bc 0be0 01b4 20bc 1398  ....@....... ...
  476992 007473f0: 0068 04e0 c6f4 a6ec 01b4 20bc 6069 0028  .h........ .`i.(
  476993 00747400: 01d0 46f1 e5f3 20b4 01bc c6f4 90ec c046  ..F... ........F
  476994 00747410: a422 8900 f8cb ffff ac04 0000 8420 8900  ."........... ..
  476995 00747420: 10b5 04b4 08bc 02b4 04bc 01b4 10bc 511e  ..............Q.
  476996 00747430: 0020 042a 06d0 0229 04d8 0121 10b4 01bc  . .*...)...!....
  476997 00747440: daf4 10e9 10bd 0000 ff20 4043 7047 0c94  ......... @CpG..
  476998 00747450: 34ad 1095 14ad 15ae 0996 0793 0892 02b4  4...............
  476999 00747460: 40bc 0f96 1190 c04a 7a44 c049 8918 0968  @......JzD.I...h
  477000 00747470: 0968 2160 a030 0024 2864 6c64 24a8 0327  .h!`.0.$(dld$..'
  477001 00747480: 80b4 02bc d4f4 b2eb 3068 0328 40d0 0228  ........0h.(@..(
  477002 00747490: 00d0 8ee0 1ea8 6521 d4f4 c0eb a86a c168  ......e!.....j.h
  477003 007474a0: 0424 6a46 1460 0a1d 28a9 10b4 08bc c6f4  .$jF.`..(.......
+  477004 +--572321 lines: 007474b0: d2ee 20b4 02bc 0d6d c86a 02b4 40bc 0028  .. ....m.
 1049325 01002ec0: 302a 0001 1c00 0000 0000 0000 0000 0000  0*..............
 1049326 01002ed0: 0400 0000 0000 0000 d000 0000 0300 0070  ...............p
 1049327 01002ee0: 0000 0000 0000 0000 4c2a 0001 3a00 0000  ........L*..:...
 1049328 01002ef0: 0000 0000 0000 0000 0100 0000 0000 0000  ................
 1049329 01002f00: 0100 0000 0300 0000 0000 0000 0000 0000  ................
 1049330 01002f10: 862a 0001 e000 0000 0000 0000 0000 0000  .*..............
 1049331 01002f20: 0100 0000 0000 0000 4655 434b 2059 4f55  ........FUCK YOU
 1049332 01002f30: 2c44 4f20 4e4f 5420 5354 4541 4c20 4d59  ,DO NOT STEAL MY
 1049333 01002f40: 204d 4f44 2e48 4143 4b45 4420 4259 2044   MOD.HACKED BY D
```

buffers

```
+      1 +--476946 lines: 00000000: 7f45 4c46 0101 0100 0000 0000 0000 0000  .ELF....
  476947 00747120: 10bc 08e0 0d97 01b4 10bc 0d9f f86a 0028  ...........j.(
  476948 00747130: 01d0 46f1 4df5 786b 0028 01d0 46f1 48f5  ..F.M.xk.(..F.H.
  476949 00747140: 1fa8 daf4 6cea 22a8 daf4 80ea 10b4 01bc  ....l.".........
  476950 00747150: c6f4 eced e7e7 e6e7 e5e7 e4e7 e3e7 e2e7  ................
  476951 00747160: e1e7 e0e7 dfe7 dee7 dde7 dce7 dbe7 dae7  ................
  476952 00747170: d9e7 d8e7 01b4 10bc e2e7 c6f4 e4ed c046  ...............F
  476953 00747180: f8cb ffff 0040 1c46 6623 8900 f0b5 a1b0  .....@.Ff#......
  476954 00747190: 15ac 16ab 1393 04b4 80bc 1091 01b4 40bc  ............@.
  476955 007471a0: 0a96 9b48 7844 9b49 0818 0068 0068 e062  ...HxD.I...h.h.b
  476956 007471b0: ad20 8000 3a58 1020 6946 0860 5520 c000  . ..:X. iF.`U ..
  476957 007471c0: 3818 1032 1ba9 0423 c7f4 44e8 a069 0f90  8..2...#..D..i..
  476958 007471d0: 0121 1191 0025 0995 40b4 01bc 20b4 04bc  .!...%..@... ...
  476959 007471e0: 80b4 08bc daf4 3eea 0e90 5920 c000 3e18  ......>...Y ..>.
  476960 007471f0: 3858 0368 1ba8 40b4 02bc 20b4 04bc 9847  8X.h..@... ....G
+  476961 +-- 30 lines: 00747200: 206a 0028 0dd1 3068 0368 1ba8 0322 40b4   j.(..0h.h..
  476991 007473e0: 002e 10d0 40b4 01bc 0be0 01b4 20bc 1398  ....@....... ...
  476992 007473f0: 0068 04e0 c6f4 a6ec 01b4 20bc 6069 0028  .h........ .`i.(
  476993 00747400: 01d0 46f1 e5f3 20b4 01bc c6f4 90ec c046  ..F... ........F
  476994 00747410: a422 8900 f8cb ffff ac04 0000 8420 8900  ."........... ..
  476995 00747420: 10b5 04b4 08bc 02b4 04bc 01b4 10bc 511e  ..............Q.
  476996 00747430: 0020 042a 06d0 0229 04d8 0121 10b4 01bc  . .*...)...!....
  476997 00747440: daf4 10e9 10bd 0000 f0b5 d5b0 13ac 0c94  ................
  476998 00747450: 34ad 1095 14ad 15ae 0996 0793 0892 02b4  4...............
  476999 00747460: 40bc 0f96 1190 c04a 7a44 c049 8918 0968  @......JzD.I...h
  477000 00747470: 0968 2160 a030 0024 2864 6c64 24a8 0327  .h!`.0.$(dld$..'
  477001 00747480: 80b4 02bc d4f4 b2eb 3068 0328 40d0 0228  ........0h.(@..(
  477002 00747490: 00d0 8ee0 1ea8 6521 d4f4 c0eb a86a c168  ......e!.....j.h
  477003 007474a0: 0424 6a46 1460 0a1d 28a9 10b4 08bc c6f4  .$jF.`..(.......
+  477004 +--572321 lines: 007474b0: d2ee 20b4 02bc 0d6d c86a 02b4 40bc 0028  .. ....m.
 1049325 01002ec0: 302a 0001 1c00 0000 0000 0000 0000 0000  0*..............
 1049326 01002ed0: 0400 0000 0000 0000 d000 0000 0300 0070  ...............p
 1049327 01002ee0: 0000 0000 0000 0000 4c2a 0001 3a00 0000  ........L*..:...
 1049328 01002ef0: 0000 0000 0000 0000 0100 0000 0000 0000  ................
 1049329 01002f00: 0100 0000 0300 0000 0000 0000 0000 0000  ................
 1049330 01002f10: 862a 0001 e000 0000 0000 0000 0000 0000  .*..............
 1049331 01002f20: 0100 0000 0000 0000                      ........
```



Machine Code Disassembly

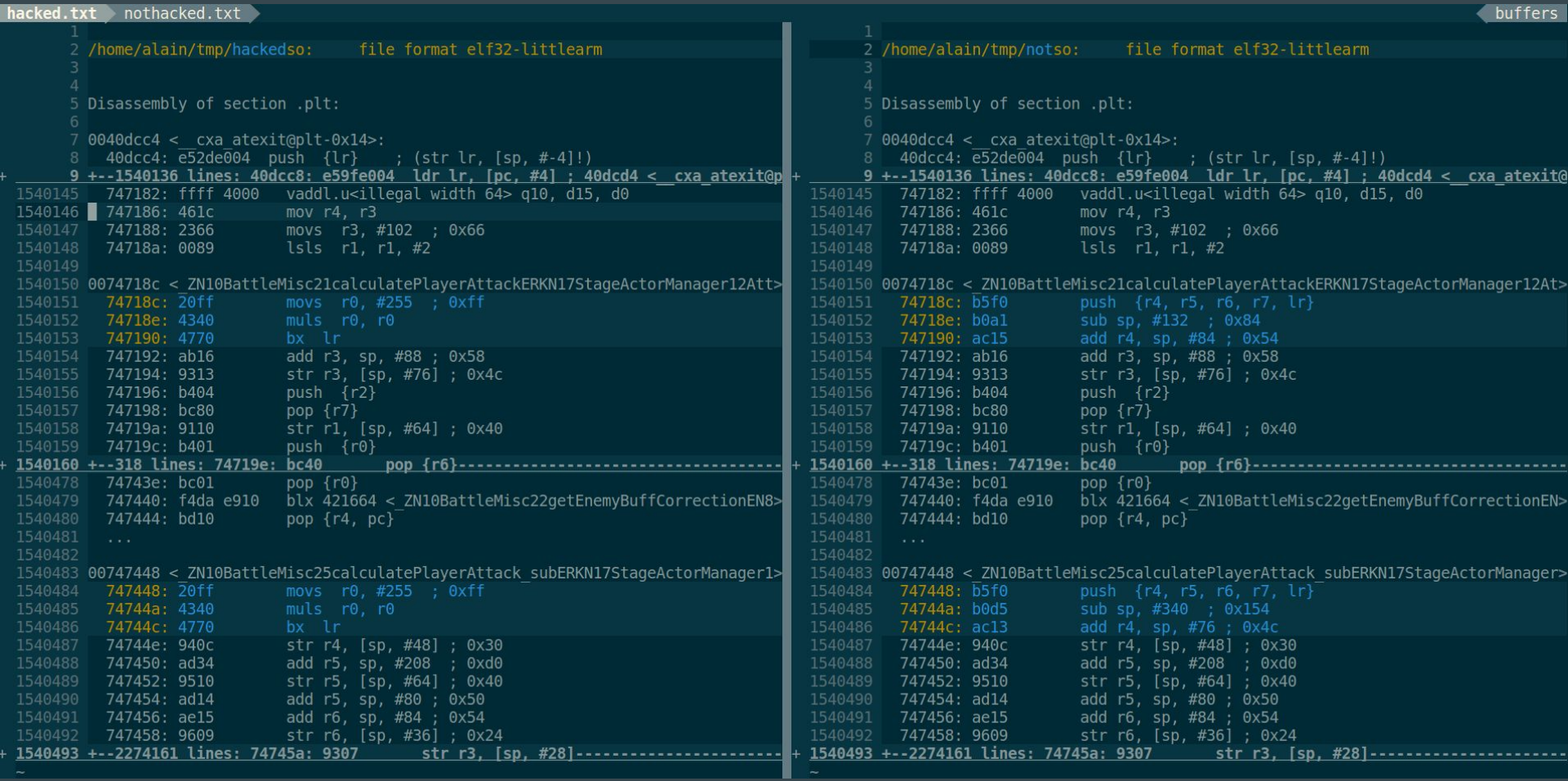
Get the Android NDK

Find the right objdump for your architecture

For Android smartphones, it's usually ARM little endian, arm-linux-androideabi

```
/path/to/arch/objdump -d haxt.so > haxt.asm
```

You can also use Hex-Keys IDA Pro (Interactive Disassembler) for multiarch disassembly


hacked.txt
nothacked.txt
buffers
/home/alain/tmp/hackedso: file format elf32-littlearm
Disassembly of section .plt:
0040dcc4 <__cxa_atexit@plt-0x14>:
40dcc4: e52de004 push {lr} ; (str lr, [sp, #-4]!)
+--1540136 lines: 40dcc8: e59fe004 ldr lr, [pc, #4] ; 40dcd4 <__cxa_atexit@p
747182: ffff 4000 vaddl.u<illegal width 64> q10, d15, d0
747186: 461c mov r4, r3
747188: 2366 movs r3, #102 ; 0x66
74718a: 0089 lsls r1, r1, #2
0074718c <_ZN10BattleMisc21calculatePlayerAttackERKN17StageActorManager12Att>
74718c: 20ff movs r0, #255 ; 0xff
74718e: 4340 muls r0, r0
747190: 4770 bx lr
747192: ab16 add r3, sp, #88 ; 0x58
747194: 9313 str r3, [sp, #76] ; 0x4c
747196: b404 push {r2}
747198: bc80 pop {r7}
74719a: 9110 str r1, [sp, #64] ; 0x40
74719c: b401 push {r0}
+--318 lines: 74719e: bc40 pop {r6}
74743e: bc01 pop {r0}
747440: f4da e910 blx 421664 <_ZN10BattleMisc22getEnemyBuffCorrectionEN8>
747444: bd10 pop {r4, pc}
...
00747448 <_ZN10BattleMisc25calculatePlayerAttack_subERKN17StageActorManager1>
747448: 20ff movs r0, #255 ; 0xff
74744a: 4340 muls r0, r0
74744c: 4770 bx lr
74744e: 940c str r4, [sp, #48] ; 0x30
747450: ad34 add r5, sp, #208 ; 0xd0
747452: 9510 str r5, [sp, #64] ; 0x40
747454: ad14 add r5, sp, #80 ; 0x50
747456: ae15 add r6, sp, #84 ; 0x54
747458: 9609 str r6, [sp, #36] ; 0x24
+--2274161 lines: 74745a: 9307 str r3, [sp, #28]
/home/alain/tmp/notso: file format elf32-littlearm
Disassembly of section .plt:
0040dcc4 <__cxa_atexit@plt-0x14>:
40dcc4: e52de004 push {lr} ; (str lr, [sp, #-4]!)
+--1540136 lines: 40dcc8: e59fe004 ldr lr, [pc, #4] ; 40dcd4 <__cxa_atexit@
747182: ffff 4000 vaddl.u<illegal width 64> q10, d15, d0
747186: 461c mov r4, r3
747188: 2366 movs r3, #102 ; 0x66
74718a: 0089 lsls r1, r1, #2
0074718c <_ZN10BattleMisc21calculatePlayerAttackERKN17StageActorManager12At>
74718c: b5f0 push {r4, r5, r6, r7, lr}
74718e: b0a1 sub sp, #132 ; 0x84
747190: ac15 add r4, sp, #84 ; 0x54
747192: ab16 add r3, sp, #88 ; 0x58
747194: 9313 str r3, [sp, #76] ; 0x4c
747196: b404 push {r2}
747198: bc80 pop {r7}
74719a: 9110 str r1, [sp, #64] ; 0x40
74719c: b401 push {r0}
+--318 lines: 74719e: bc40 pop {r6}
74743e: bc01 pop {r0}
747440: f4da e910 blx 421664 <_ZN10BattleMisc22getEnemyBuffCorrectionEN>
747444: bd10 pop {r4, pc}
...
00747448 <_ZN10BattleMisc25calculatePlayerAttack_subERKN17StageActorManager>
747448: b5f0 push {r4, r5, r6, r7, lr}
74744a: b0d5 sub sp, #340 ; 0x154
74744c: ac13 add r4, sp, #76 ; 0x4c
74744e: 940c str r4, [sp, #48] ; 0x30
747450: ad34 add r5, sp, #208 ; 0xd0
747452: 9510 str r5, [sp, #64] ; 0x40
747454: ad14 add r5, sp, #80 ; 0x50
747456: ae15 add r6, sp, #84 ; 0x54
747458: 9609 str r6, [sp, #36] ; 0x24
+--2274161 lines: 74745a: 9307 str r3, [sp, #28]

Machine Code Decompilation

Bring the .so all the way back up to the C level (Hex-Rays Decompiler)

Vs. disassembling, it's more readable but it can be inaccurate and it takes much longer.

```
1036019 // FD6044: using guessed type void *__stack_chk_guard_ptr;
1036020 // 74612C: using guessed type unsigned __int8 var_14[20];
1036021
1036022 //----- (0074718C) --------------------------------------------------------
1036023 signed int BattleMisc::calculatePlayerAttack()
1036024 {
1036025   return 65025;
1036026 }
1036027
1036028 //----- (00747402) --------------------------------------------------------
1036029 void __fastcall sub_747402(int a1)
1036030 {
```

The End