
目录

前言	1.1
Frida调试Android概览	1.2
经验心得	1.3
ctor函数	1.3.1
带overload的ctor函数	1.3.1.1
hook类的ctor返回空值	1.3.1.2
\$new函数	1.3.2
借助崩溃找overload型参数定义	1.3.3
找到真正Object数组并打印详情	1.3.4
enum枚举类	1.3.5
打印ByteBuffer数据和字符串值	1.3.6
打印Map的ArrayList	1.3.7
打印函数调用和函数堆栈时支持过滤功能	1.3.8
用logcat辅助找进程	1.3.9
字符串反混淆	1.3.10
找到并调用字符串解密函数m1.e.i	1.3.10.1
用JEB自动反编译字符串	1.3.10.2
常见问题	1.4
Java和js变量的映射关系	1.4.1
找不到类：Error java.lang.ClassNotFoundException	1.4.2
cannot call instance method without an instance	1.4.3
附录	1.5
参考资料	1.5.1

Frida调试Android经验总结

- 最新版本： v1.0.0
- 更新时间： 20251213

简介

整理出之前用安卓逆向期间，用Frida调试Android的app和程序的经验和心得。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

HonKit源码

- [crifan/frida_debug_android_summary: Frida调试Android经验总结](#)

如何使用此HonKit源码去生成发布为电子书

详见：[crifan/honkit_template: demo how to use crifan honkit template and demo](#)

在线浏览

- [Frida调试Android经验总结 book.crifan.org](#)
- [Frida调试Android经验总结 crifan.github.io](#)

离线下载阅读

- [Frida调试Android经验总结 PDF](#)
- [Frida调试Android经验总结 ePub](#)
- [Frida调试Android经验总结 Mobi](#)

版权和用途说明

此电子书教程的全部内容，如无特别说明，均为本人原创。其中部分内容参考自网络，均已备注了出处。如发现有侵权，请通过邮箱联系我 [admin 艾特 crifan.com](#)，我会尽快删除。谢谢合作。

各种技术类教程，仅作为学习和研究使用。请勿用于任何非法用途。如有非法用途，均与本人无关。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 [crifan](#) 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

其他

作者的其他电子书

本人 `crifan` 还写了其他 150+ 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme](#): Crifan的电子书的使用说明

关于作者

关于作者更多介绍，详见：

[关于CrifanLi李茂 – 在路上](#)

crifan.org，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新：
2025-12-13 22:48:33

Frida调试Android概览

之前用Frida调试Android的程序，遇到各种问题和经验心得，整理出来，供参考。

crifan.org，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook最后更新：
2025-12-13 22:29:50

Frida调试Android经验心得

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2025-12-13 22:16:24

ctor函数

概述：

- Java的类的**ctor函数**

- 别名

- = Constructor
 - = 构造函数
 - = 初始化函数
 - = init函数
 - = \$init函数

- 特点

- 没有返回值

- Frida去hook Java的**ctor函数**

- 真正的hook写法是：

- `this.$init()`
 - 而不是： `this.$new()`

- 由于ctor函数没有返回值

- 所以如果用

```
var newXxx = this.$init()  
return newXxx
```

- 则始终返回： `undefined`
 - 想要获取到init初始化后的java类的实例 = 想要打印实例的值：
 - 可以用： `this`

```
this.init()  
var newXxx = this  
console.log(`newXxx=${newXxx}`)  
return
```

crifan.org，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新：
2025-08-20 23:09:42

带overload的ctor函数

举例：

去hook带overload的ctor函数=init函数：

对于：

[String \(Java Platform SE 8 \) \(oracle.com\)](#)

的：

- Constructor=ctor=构造函数=初始化函数=init函数
 - `public String(String original)`
 - `String(byte[] bytes, Charset charset)`
 - `String(byte[] bytes, String charsetName)`

核心写法是：

- `var func_String_ctor = cls_String.$init.overload('java.lang.String')`
- `var func_String_ctor_2pbc = cls_String.$init.overload('[B', 'java.nio.charset.Charset')`
- `var func_String_ctor_2pbs = cls_String.$init.overload('[B', 'java.lang.String')`

核心代码是：

```
static String(func_isShowLog null) {
  /***** java.lang.String *****/
  var className_String = "java.lang.String"
  // FridaAndroidUtil.printClassAllMethodsFields(className_String)

  var cls_String = Java.use(className_String)
  console.log("cls_String=" + cls_String)

  // public String(String original)
  var func_String_ctor = cls_String.$init.overload('java.lang.String')
  // var func_String_ctor = cls_String.getInstance.overload('java.lang.String')
  // var func_String_ctor = cls_String.$new.overload('java.lang.String')
  console.log("func_String_ctor=" + func_String_ctor)
  if (func_String_ctor) {
    func_String_ctor.implementation = function (original) {
      ...
      this.$init(original)
      return
    }
  }

  // String(byte[] bytes, Charset charset)
  //
  var func_String_ctor_2pbc = cls_String.$init.overload('[B', 'java.nio.charset.Charset')
  console.log("func_String_ctor_2pbc=" + func_String_ctor_2pbc)
  if (func_String_ctor_2pbc) {
```

```
func_String_ctor_2pbc.implementation = function (bytes, charset) {  
    ...  
    this.$init(bytes, charset)  
    var newString_2pbc = this  
    console.log(`newString_2pbc=${newString_2pbc}`)  
    return  
}  
}  
  
// String(byte[] bytes, String charsetName)  
//  
var func_String_ctor_2pbs = cls_String.$init.overload('[B', 'java.lang.String')  
console.log("func_String_ctor_2pbs=" + func_String_ctor_2pbs)  
if (func_String_ctor_2pbs) {  
    func_String_ctor_2pbs.implementation = function (bytes, charsetName) {  
        ...  
        this.$init(bytes, charsetName)  
        var newString_2pbs = this  
        console.log(`newString_2pbs=${newString_2pbs}`)  
        return  
    }  
}  
}
```

完整代码详见：

<https://github.com/crifan/JsFridaUtil/blob/main/frida/FridaHookAndroidJava.js>

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2025-08-20 23:00:49

hook类的ctor返回空值

对于java代码

```
public final class C666190Q1w<A, B> {  
    public final A LIZ;  
    public final B LIZIZ;  
    ...  
    public C666190Q1w(A a2, B b) {  
        this.LIZ = a2;  
        this.LIZIZ = b;  
    }  
}
```

去hook了构造函数，最终代码是：

```
function hookDouyinClass_X_0Q1w(ThrowableCls) {  
    /***** X.0Q1w *****/  
  
    var clsName_0Q1w = "X.0Q1w"  
    // printClassAllMethodsFields(clsName_0Q1w)  
  
    var cls_0Q1w = Java.use(clsName_0Q1w)  
    console.log("cls_0Q1w=" + cls_0Q1w)  
  
    // public C666190Q1w(A a2, B b) {  
    // call: C666190Q1w<ByteBuffer, Long> c666190Q1w = new C666190Q1w<>(allocate5, Long.v  
    alueOf(j6))  
    var func_0Q1w_ctor = cls_0Q1w.$init  
    console.log("func_0Q1w_ctor=" + func_0Q1w_ctor)  
    if (func_0Q1w_ctor) {  
        func_0Q1w_ctor.implementation = function (a2, b) {  
            var funcName = "0Q1w(a2,b)"  
            var funcParaDict = {  
                "a2": a2,  
                "b": b,  
            }  
            printFunctionCallAndStack(funcName, funcParaDict, ThrowableCls)  
            // var new0Q1w = this.$init(a2, b)  
            // console.log("new0Q1w=" + new0Q1w)  
            this.$init(a2, b)  
            console.log("this=" + this)  
            console.log("this.LIZ=" + this.LIZ)  
            console.log("this.LIZIZ=" + this.LIZIZ)  
            // var new0Q1w = this  
            // return new0Q1w  
            return  
        }  
    }  
}
```

而其中，之前的写法是：

```
var new0Q1w = this.$init(a2, b)
console.log("new0Q1w=" + new0Q1w)
return new0Q1w
```

但会输出： new0Q1w=undefined

是因为：

- 构造函数=ctor函数=init函数

- 并没有返回值

- 所以（如果用 `return newXxx`，则）返回的值始终是： `undefined`

- 所以后续改为：直接 `return` 返回，效果也是一样的

- 如果想要打印初始化的类的实例值，则可以

- （在初始化后）用：`this`

举例：

```
this.$init(a2, b)
console.log("this=" + this)
console.log("this.LIZ=" + this.LIZ)
console.log("this.LIZIZ=" + this.LIZIZ)
return
```

即可查看到：（内存中的）实例的相关的值：

```
this: X.0Q1w@e1993e0
this.LIZ: Java.Field{holder: X.0Q1w@e1993e0, fieldType: 2, fieldReturnType: [object Object], value: java.nio.HeapByteBuffer[pos=12288 lim=12288 cap=12288]}
this.LIZIZ: Java.Field{holder: X.0Q1w@e1993e0, fieldType: 2, fieldReturnType: [object Object], value: 236556288}
```

以及进一步的要查看属性的值，是加上： `.value`

```
console.log("this.LIZ.value=" + this.LIZ.value)
console.log("this.LIZIZ.value=" + this.LIZIZ.value)
```

输出：

```
this.LIZ.value=java.nio.HeapByteBuffer[pos=12288 lim=12288 cap=12288]
this.LIZIZ.value=236556288
```

crifan.org，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新：
2025-08-20 22:49:37

\$new函数

另外有个（看起来，和 `ctor` 的 `$init` 有点类似的）写法是：

- `$new`
 - 含义：新建一个Java对象的实例

\$new的用法举例

```
var curThrowableCls = Java.use("java.lang.Throwable")
let newThrowable = curThrowableCls.$new()
```

后续即可正常使用，此处新建的 `Throwable` 的实例对象 `newThrowable` 了

比如去打印函数调用堆栈：

```
var stackElements = newThrowable.getStackTrace()
console.log("genStackStr: stackElements=" + stackElements)
```

相关的代码：

```
// generate current stack trace string
static genStackStr(prefix "") {
    // let newThrowable = ThrowableCls.$new()
    // let newThrowable = this.curThrowableCls.$new()
    let newThrowable = FridaAndroidUtil.curThrowableCls.$new()
    // console.log("genStackStr: newThrowable=" + newThrowable)
    var stackElements = newThrowable.getStackTrace()
    // console.log("genStackStr: stackElements=" + stackElements)
    if (!JsUtil.isEmpty(prefix)){
        prefix = prefix + " "
    }
    const linePrefix = "\n "
    var stackStr = prefix + "java Stack:" + linePrefix + stackElements[0] //method//sta
ckElements[0].getMethodName()
    for (var i = 1; i < stackElements.length; i++) {
        stackStr += linePrefix + "at " + stackElements[i]
    }
    // stackStr = "\n\n" + stackStr
    stackStr = stackStr + "\n"
    // console.log("genStackStr: stackStr=" + stackStr)

    return stackStr
}
```

全部代码详见：

<https://github.com/crifan/JsFridaUtil/blob/main/frida/FridaAndroidUtil.js>

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2025-08-20 23:02:52

借助崩溃找overload型参数定义

心得概述

如果要hook的函数的，不会写具体的定义

如果是同名的重载的overload类型，则可以：

故意不写具体定义，只写上implementation

然后让frida报错，而告知你多个函数的完整的定义的定义的写法

举例

对于代码：

```
sources/com/bytedance/retrofit2/client/Request.java
```

```
public Request(Builder builder) {  
    ...  
  
    public Request(String str, String str2, List<Header> list, TypedOutput typedOutput,  
int i, boolean z, int i2, boolean z2, Object obj) {  
    ...  
  
    public Request(String str, String str2, List<Header> list, TypedOutput typedOutput,  
RequestBody requestBody, int i, int i2, boolean z, int i3, boolean z2, Object obj, Stri  
ng str3, Map<Class<?>, Object> map) {  
    ...  
}
```

这个类，有3个构造函数

想要hook，但是对于类的具体定义

尤其是其中一些类，比如：

- `Builder`
 - 是Request的内部的类
- `List<Header> list`
- `Map<Class<?>, Object> map`

等类型，不知道具体的Java的类型如何写

那么就可以故意写成：

```
var func_Request_ctor_b = cls_Request.$init  
...  
func_Request_ctor_b.implementation = function (builder) {
```

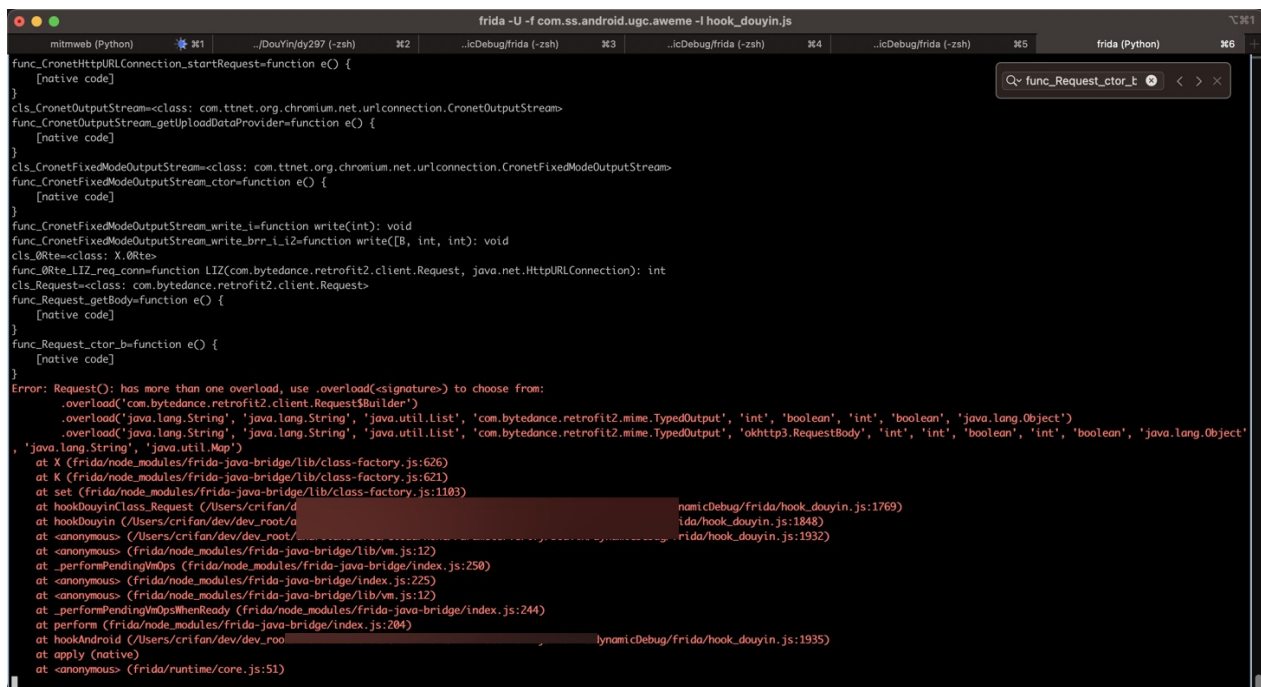
完整代码：

```
// public Request(Builder builder) {
var func_Request_ctor_b = cls_Request.$init
// var func_Request_ctor_b = cls_Request.$init.overload('com.bytedance.retrofit2.client.Request$Builder')
console.log("func_Request_ctor_b=" + func_Request_ctor_b)
if (func_Request_ctor_b) {
  func_Request_ctor_b.implementation = function (builder) {
    var funcName = "Request(builder)"
    var funcParaDict = {
      "builder": builder,
    }
    if (isWeConcernedUrl(this)){
      FridaUtil.printFunctionCallAndStack(funcName, funcParaDict)
    }
    return this.$init(builder)
  }
}
```

运行后触发Frida的崩溃报错，说找不到对应overload的函数

```
Error: Request(): has more than one overload, use .overload(<signature>) to choose from
:
  .overload('com.bytedance.retrofit2.client.Request$Builder')
  .overload('java.lang.String', 'java.lang.String', 'java.util.List', 'com.bytedance.retrofit2.mime.TypedOutput', 'int', 'boolean', 'int', 'boolean', 'java.lang.Object')
  .overload('java.lang.String', 'java.lang.String', 'java.util.List', 'com.bytedance.retrofit2.mime.TypedOutput', 'okhttp3.RequestBody', 'int', 'int', 'boolean', 'int', 'boolean', 'java.lang.Object', 'java.lang.String', 'java.util.Map')
    at X (frida/node_modules/frida-java-bridge/lib/class-factory.js:626)
    at K (frida/node_modules/frida-java-bridge/lib/class-factory.js:621)
    at set (frida/node_modules/frida-java-bridge/lib/class-factory.js:1103)
    at hookyyyClass_Request (/Users/crifan/dev/dev_root/androidReverse/xx/x/yyy/dynamicDebug/frida/hook_yyy.js:1769)
    at hookyyy (/Users/crifan/dev/dev_root/androidReverse/x/x/yyy/dynamicDebug/frida/hook_yyy.js:1848)
    at <anonymous> (/Users/crifan/dev/dev_root/androidReverse/x/x/yyy/dynamicDebug/frida/hook_yyy.js:1932)
    at <anonymous> (frida/node_modules/frida-java-bridge/lib/vm.js:12)
    at _performPendingVmOps (frida/node_modules/frida-java-bridge/index.js:250)
    at <anonymous> (frida/node_modules/frida-java-bridge/index.js:225)
    at <anonymous> (frida/node_modules/frida-java-bridge/lib/vm.js:12)
    at _performPendingVmOpsWhenReady (frida/node_modules/frida-java-bridge/index.js:244)

    at perform (frida/node_modules/frida-java-bridge/index.js:204)
    at hookAndroid (/Users/crifan/dev/dev_root/androidReverse/x/x/yyy/dynamicDebug/frida/hook_yyy.js:1935)
    at apply (native)
    at <anonymous> (frida/runtime/core.js:51)
```



– 其中列出完全的全部的overload的函数的定义，即：

- `.overload('com.bytedance.retrofit2.client.Request$Builder')`
- `.overload('java.lang.String', 'java.lang.String', 'java.util.List', 'com.bytedance.retrofit2.mime.TypedOutput', 'int', 'boolean', 'int', 'boolean', 'java.lang.Object')`
- `.overload('java.lang.String', 'java.lang.String', 'java.util.List', 'com.bytedance.retrofit2.mime.TypedOutput', 'okhttp3.RequestBody', 'int', 'int', 'boolean', 'int', 'boolean', 'java.lang.Object', 'java.lang.String', 'java.util.Map')`

如此，就可以正常去写hook函数了：

```
// public Request(Builder builder) {
// var func_Request_ctor_b = cls_Request.$init
var func_Request_ctor_b = cls_Request.$init.overload('com.bytedance.retrofit2.client.Request$Builder')
console.log("func_Request_ctor_b=" + func_Request_ctor_b)
if (func_Request_ctor_b) {
  func_Request_ctor_b.implementation = function (builder) {
    var funcName = "Request(builder)"
    var funcParaDict = {
      "builder": builder,
    }
    if (isWeConcernedUrl(this)){
      FridaUtil.printFunctionCallAndStack(funcName, funcParaDict)
    }
    return this.$init(builder)
  }
}

// public Request(String str, String str2, List<Header> list, TypedOutput typedOutput, int i, boolean z, int i2, boolean z2, Object obj) {
var func_Request_ctor_9para = cls_Request.$init.overload('java.lang.String', 'java.la
```

```

ng.String', 'java.util.List', 'com.bytedance.retrofit2.mime.TypedOutput', 'int', 'boolean', 'int', 'boolean', 'java.lang.Object')
console.log("func_Request_ctor_9para=" + func_Request_ctor_9para)
if (func_Request_ctor_9para) {
    func_Request_ctor_9para.implementation = function (str, str2, list, typedOutput, i, z, i2, z2, obj) {
        var funcName = "Request(9 Para)"
        var funcParaDict = {
            "str": str,
            "str2": str2,
            "list": list,
            "typedOutput": typedOutput,
            "i": i,
            "z": z,
            "i2": i2,
            "z2": z2,
            "obj": obj,
        }
        if (isWeConcernedUrl(this)){
            FridaUtil.printFunctionCallAndStack(funcName, funcParaDict)
        }
        return this.$init(str, str2, list, typedOutput, i, z, i2, z2, obj)
    }
}

```

```

// public Request(String str, String str2, List<Header> list, TypedOutput typedOutput,
// RequestBody requestBody, int i, int i2, boolean z, int i3, boolean z2, Object obj, String str3, Map<Class<?>, Object> map) {
var func_Request_ctor_13para = cls_Request.$init.overload('java.lang.String', 'java.lang.String', 'java.util.List', 'com.bytedance.retrofit2.mime.TypedOutput', 'okhttp3.RequestBody', 'int', 'int', 'boolean', 'int', 'boolean', 'java.lang.Object', 'java.lang.String', 'java.util.Map')
console.log("func_Request_ctor_13para=" + func_Request_ctor_13para)
if (func_Request_ctor_13para) {
    func_Request_ctor_13para.implementation = function (str, str2, list, typedOutput, requestBody, i, i2, z, i3, z2, obj, str3, map) {
        var funcName = "Request(13 Para)"
        var funcParaDict = {
            "str": str,
            "str2": str2,
            "list": list,
            "typedOutput": typedOutput,
            "requestBody": requestBody,
            "i": i,
            "i2": i2,
            "z": z,
            "i3": i3,
            "z2": z2,
            "obj": obj,
            "str3": str3,
            "map": map,
        }
        if (isWeConcernedUrl(this)){
            FridaUtil.printFunctionCallAndStack(funcName, funcParaDict)
        }
    }
}

```

```
return this.$init(str, str2, list, typedOutput, requestBody, i, i2, z, i3, z2,
obj, str3, map)
}
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2025-12-13 22:17:39

找到真正Object数组并打印详情

之前Frida调试Android时，遇到Java代码相关的，`Object[]` = Object数组方面的问题：

有些值看起来像是Object的数组，但是其实不是（有些是单独（继承自List）的类）

比如之前打印出的值：

```
ankq a => retArrObj-[# anjs 3dc8422b]
```

中的：

```
[# anjs@3dc8422b]
```

本来以为是：

```
Object 的 Array 或 List == Object[]
```

但是其实不是的

-》而其真实的类名，最后是通过

```
var clsName = FridaAndroidUtil.getJavaClassName(retArrObj)
console.log(funcName + " => retArrObj=" + retArrObj + ", clsName=" + clsName)
```

而打印出的：

```
ankq a => retArrObj-[# anjs 3dc8422b], clsName enln
```

中的： `enln`

才明白，其实是：

- `[# anjs@3dc8422b]` 看起来以为类型是 `Object[]`
 - 但其实是：另外一个类： `enln` （的实例）

-》而具体底层细节是：

其中`enln`的代码是：

```
xxx_v250861_JEB/enln.java
```

```
import util.Objects;

public final class enln extends encw {
    public static final encw a;
    final transient Object[] b;
    public final transient int c;
```

```

static {
    enln a = new enln(new Object[0], 0);
}

public enln(Object[] arr_object, int v) {
    this.b = arr_object;
    this.c = v;
}
...

```

以及其所继承的父类是：

xxx_v250861_JEB/encw.java

```

public abstract class encw extends enck implements List, List, RandomAccess {
    private static final enor a = null;
    public static final int d = 0;
    private static final long serialVersionUID = 0xFFFFFFFFCAFEFEBABEL;
    ...
}

```

-》好像由于是：

本身也是属于（父类encw所继承自的）List类型

所以最终打印出来的值，类似于toString()后的值，是：

[# anjs@3dc8422b]

看起来像是对象的数组而已，但其实本身是个单独的类

想要打印对象的数组的详情的核心思路

- （1）先找到，真正的Object[]的对象的数组的变量objArr
- （2）然后再去用for循环+objArr.length，去获取每个Object对象，打印对象的详情

举例：

此处核心代码是：

```

static printClass_enln(inputObj, prefixStr=""){
    ...

    var bValue = curObj.b.value

    console.log(newPrefStr + ClassName + ":"
        + " a=" + curObj.a.value
        + ", b=objArr=" + bValue
        + ", c=size=" + curObj.c.value
    )

    console.log("bValue.length=" + bValue.length)
    for(var idx = 0; idx < bValue.length; idx++) {
        console.log(`----- ${bValue} [${idx}] -----`)
        var curAnjs = bValue[idx]
        console.log("curAnjs=" + curAnjs)
    }
}

```

```
HookAppJava_xxx.printClass_anjs(curAnjs, prefixStr)
}
```

核心输出是：

```
ankq.a enln: a [], b objArr=# anjs 3dc8422b, c=size 1
bValue.length 1

----- # anjs 3dc8422b [0] -----
curAnjs # anjs 3dc8422b
printClass_anjs: inputObj=# anjs 3dc8422b, inputClsName anjs
printClass_anjs: curObj=# anjs 3dc8422b, curClsName anjs
ankq.a anjs: a # anjs 18f4ef8d, b 0xD7, c 45507, d 只能拨打紧急呼救电话 - China Telecom, e
0, f <clsName fjgr>=[object Object], g 455070004000000, h, i clsName fjex = ByteString
@8a9b02a size 2 contents="\026<", j -1, k <clsName null>=null
```

才是我们要的效果：

可以打印出，此处：

对象的数组 `Object[]`，即 `enln.b`，的每个对象元素（其本身是另外一个类 `anjs`）的详情（`anjs` 的属性值）

crifan.org，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook最后更新：
2025-08-22 22:43:21

enum枚举类

- 概述
 - Frida去hook安卓中Java的enum枚举的类
 - Frida去hook enum类，就像正常hook java的类一样
 - 注意：
 - 有个额外的 `.values()` 函数，需要时可以用起来

举例

Java的枚举类：

xxx_v250861_JEB/anjp.java

```
public enum anjp implements fjgu {
    BASE_CLIENT_ID(1),
    SEARCH_CLIENT_ID(2),
    VOICESEARCH_CLIENT_ID(3),
    MAPS_CLIENT_ID(4),
    YOUTUBE_CLIENT_ID(5),
    MARKET_CLIENT_ID(6),
    SHOPPER_CLIENT_ID(7),
    WALLET_CLIENT_ID(8),
    CHROME_CLIENT_ID(9),
    PLAYTX_CLIENT_ID(10),
    PLAYAX_CLIENT_ID(11),
    PROGRAM_CLIENT_ID(12);

    public final int m;

    private anjp(int v1) {
        this.m = v1;
    }

    @Override // fjgu
    public final int a() {
        return this.m;
    }

    @Override
    public final String toString() {
        return Integer.toString(this.m);
    }
}
```

hook代码：

```
static anjp() {
    var clsName_anjp = "anjp"
    // FridaAndroidUtil.printClassAllMethodsFields(clsName_anjp)
```

```

var cls_anjp = Java.use(clsName_anjp)
console.log("cls_anjp=" + cls_anjp)

// private anjp(int v1) {
//
var func_anjp_ctor = cls_anjp.$init
console.log("func_anjp_ctor=" + func_anjp_ctor)
if (func_anjp_ctor) {
    func_anjp_ctor.implementation = function (v1) {
        var funcName = "anjp"
        var funcParaDict = {
            "v1": v1,
        }
        FridaAndroidUtil.printFunctionCallAndStack(funcName, funcParaDict)

        var newAnjp = this.$init(v1)
        console.log(funcName + " => newAnjp=" + newAnjp)
        return newAnjp
    }
}
...
}

```

日志:

```

===== anjp.toString =====
anjp.toString:
anjp.toString java Stack:
  anjp.toString Native Method
  at enls.y Native Method
  at enls.get com.google.android.xxx 250861022 25.08.61 (150400-731857739):8)
  at enls.get Native Method
  at ankb.a com.google.android.xxx 250861022 25.08.61 (150400-731857739):1064)
  at ankb.a Native Method
  at fosv.d com.google.android.xxx 250861022 25.08.61 (150400-731857739):10)
  at fosv.d Native Method
  at fosv.a com.google.android.xxx 250861022 25.08.61 (150400-731857739):7)
  at fosv.a Native Method
  at ankf.a com.google.android.xxx 250861022 25.08.61 (150400-731857739):67)
  at ankf.a Native Method
  at fosv.d com.google.android.xxx 250861022 25.08.61 (150400-731857739):10)
  at fosv.d Native Method
  at fosv.a com.google.android.xxx 250861022 25.08.61 (150400-731857739):7)
  at fosv.a Native Method
  at anmg.init> ( com.google.android.xxx 250861022 25.08.61 (150400-731857739):9)
  at anmg.init> Native Method
  at anmi.a com.google.android.xxx 250861022 25.08.61 (150400-731857739):175)
  at anmi.a Native Method
  at fosv.d com.google.android.xxx 250861022 25.08.61 (150400-731857739):10)
  at fosv.d Native Method
  at fosv.a com.google.android.xxx 250861022 25.08.61 (150400-731857739):7)
  at fosv.a Native Method
  at anmh.b com.google.android.xxx 250861022 25.08.61 (150400-731857739):1)
  at anfl.a com.google.android.xxx 250861022 25.08.61 (150400-731857739):567)

```

```
at anfl.a Native Method
at angv.a com.google.android.xxx 250861022 25.08.61 (150400-731857739) : 850
at angv.a Native Method
at com.google.android.xxx.checkin.CheckinIntentOperation.onHandleIntent com.google.a
ndroid.xxx 250861022 25.08.61 (150400-731857739) : 35
at com.google.android.chimera.IntentOperation.onHandleIntent com.google.android.xxx
250861022 25.08.61 (150400-731857739) : 2
at ansx.onHandleIntent com.google.android.xxx 250861022 25.08.61 (150400-731857739) : 8
)
at rli.run com.google.android.xxx 250861022 25.08.61 (150400-731857739) : 70
at rlh.run com.google.android.xxx 250861022 25.08.61 (150400-731857739) : 152
at eksg.run com.google.android.xxx 250861022 25.08.61 (150400-731857739) : 21
at java.util.concurrent.ThreadPoolExecutor.runWorker ThreadPoolExecutor.java 1145
at java.util.concurrent.ThreadPoolExecutor$Worker.run ThreadPoolExecutor.java 644
at java.lang.Thread.run Thread.java 1012

anjp.toString => retString 1
```

然后其中有个额外特殊的情况：

- enum, 自带有个函数：
 - .values()
 - 此处
 - public static anjp[] anjp.values()
 - 可以用于获取值的列表
 - 需要时可以去调用

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2025-08-22 22:43:16

打印ByteBuffer数据和字符串值

Frida的js中，想要打印Android中Java的ByteBuffer的数据（最好打印出对应字符串）：

工具类函数：

```
// java ByteBuffer to String
static javaByteBufferToStr(byteBufer){
    console.log(`javaByteBufferToStr: byteBufer=${byteBufer}`)
    // javaByteBufferToStr: byteBufer=java.nio.DirectByteBuffer[pos=793 lim=16375 cap=16375]
    // // record current status
    // var oldLimit = byteBufer.limit()
    // var oldPositon = byteBufer.position()
    // console.log(`oldLimit=${oldLimit}, oldPositon=${oldPositon}`)
    byteBufer.flip() // rewind to start position
    console.log(`after ${byteBufer} flip`)
    // after java.nio.DirectByteBuffer[pos=0 lim=793 cap=16375] flip
    // var utf8CharBuffer = FridaAndroidUtil.StandardCharsets.UTF_8.decode(byteBufer)
    // var charsetUtf8 = FridaAndroidUtil.StandardCharsets.UTF_8
    var charsetUtf8 = FridaAndroidUtil.StandardCharsets.UTF_8.value
    console.log("charsetUtf8=" + charsetUtf8)
    // charsetUtf8=UTF-8
    var utf8CharBuffer = charsetUtf8.decode(byteBufer)
    console.log("utf8CharBuffer=" + utf8CharBuffer)
    var utf8BufStr = utf8CharBuffer.toString()
    console.log("utf8BufStr=" + utf8BufStr)
    // // restore status before flip
    // byteBufer.limit(oldLimit)
    // byteBufer.position(oldPositon)

    return utf8BufStr
}
```

调用代码：

```
// public final void onReadSucceeded(boolean z) {
//
var func_CronetUploadDataStream_onReadSucceeded = cls_CronetUploadDataStream.onReadSucceeded
console.log("func_CronetUploadDataStream_onReadSucceeded=" + func_CronetUploadDataStream_onReadSucceeded)
if (func_CronetUploadDataStream_onReadSucceeded) {
    func_CronetUploadDataStream_onReadSucceeded.implementation = function (z) {
        var funcName = "CronetUploadDataStream.onReadSucceeded"
        var funcParaDict = {
            "z": z,
        }
        FridaAndroidUtil.printFunctionCallAndStack(funcName, funcParaDict)

        // var mByteBuffer_before = this.mByteBuffer
        var mByteBuffer_before = this._g
```



```
before CronetUploadDataStream.onReadSucceeded: buffer data java.nio.DirectByteBuffer[pos  
=796 lim=16375 cap=16375]
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2025-08-22 22:44:09

打印Map的ArrayList

举例说明：

对于此处的：

(1) 源码中是：

```
ArrayList<Map.Entry<String, String>> requestHeaders,
```

(2) (jadx、JEB反编译后的) 混淆后代码是：

```
ArrayList mRequestHeaders
```

(3) 后续用

```
FridaAndroidUtil.getJavaClassName(mRequestHeaders)
```

打印出java的类型是：

- `j$.util.m`

其中用jadx搜到的是：

```
xxx_v250861_jadx/sources/j$/util/C0052m.java
```

```
package $ util;

import      SortedMap;

/* renamed from: j$.util.m, reason: case insensitive filesystem */
/* loaded from: classes12.dex */
final class C0052m extends C0049j implements SortedMap {
    . . .
```

其中的：

- `implements SortedMap`
 - 看起来，以为是，很像是：Map相关的类
- `extends C0049j`
 - 找到具体类是
 - `xxx_v250861_jadx/sources/j$/util/C0049j.java`
 - `class C0049j implements java.util.Map, Serializable, Map {`
 - 看起来像是：`java.util.Map = Map` 相关的类
 - -> 但是经过实际测试，其实是：`Array` 类型的类：`ArrayList`

总结起来就是：

此处想要打印此处的：

- java类型
 - 源码中是： `ArrayList`
 - `ArrayList<Map.Entry<String, String>>`
 - 混淆代码中是： `ArrayList mRequestHeaders`
 - `ArrayList`
 - frida中打印出的是
 - `j$.util.m`
 - 从相关源码中看起来，很像是： `Map` 的类
 - 但实际上确是： `Array` 的类

的最新代码是：

```
// java ArrayList (byte array / List<Integer> / ArrayList<Map.Entry<String, String>>
) to string
static javaArrayListToStr(javaArraryList){
    // var jsArrayList = FridaAndroidUtil.javaByteArrToJsByteArr(javaArraryList)
    // console.log("jsArrayList=" + jsArrayList)
    // var jsArrayListStr = jsArrayList.toString()
    // console.log("jsArrayListStr=" + jsArrayListStr)
    // return jsArrayListStr

    var javaObjList = javaArraryList.toArray()
    console.log("javaObjList=" + javaObjList)
    var javaObjListStr = javaObjList.toString()
    console.log("javaObjListStr=" + javaObjListStr)
    return javaObjListStr
}

...

var mRequestHeaders = curObj.s.value
console.log(`mRequestHeaders=${mRequestHeaders}, type=${typeof mRequestHeaders}`)
console.log(`mRequestHeaders class=${FridaAndroidUtil.getJavaClassName(mRequestHe
aders)}`)
var mRequestHeadersStr = FridaAndroidUtil.javaArrayListToStr(mRequestHeaders)

// // FridaAndroidUtil.updateClassLoader("java.util.Map")
// var headersStr = mRequestHeaders.toString()
// console.log("headersStr=" + headersStr)
// var headersValues = mRequestHeaders.values()
// console.log("headersValues=" + headersValues)
// var headersEntries = mRequestHeaders.entrySet()
// console.log("headersEntries=" + headersEntries)
// var headersKeys = mRequestHeaders.keySet()
// console.log("headersKeys=" + headersKeys)

// var mRequestHeadersStr = FridaAndroidUtil.mapToStr(mRequestHeaders)

console.log("mRequestHeadersStr=" + mRequestHeadersStr)

console.log("CronetUrlRequest:" + clsNameStr
+ " mInitialUrl=" + curObj.o.value
+ " mInitialMethod=" + curObj.r.value
```



```
// + " mRequestHeaders=" + mRequestHeaders  
+ " mRequestHeaders=" + mRequestHeadersStr
```

相关日志：

```
mRequestHeaders=[object Object], type object  
mRequestHeaders class js.util.m  
javaObjList Content Type application x www form urlencoded, Authorization AidLogin xxx y  
yy, app org.telegram.messenger.web, gcm_ver 250861022, app_ver 56969  
javaObjListStr Content Type application x www form urlencoded, Authorization AidLogin xxx  
yyy, app org.telegram.messenger.web, gcm_ver 250861022, app_ver 56969  
mRequestHeadersStr Content Type application x www form urlencoded, Authorization AidLogi  
n xxx yyy, app org.telegram.messenger.web, gcm_ver 250861022, app_ver 56969
```

就对了。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2025-08-22 22:25:48

打印函数调用和函数堆栈时支持过滤功能

- 需求
 - 想要在Frida去hook代码时，打印 函数调用和函数堆栈时：
 - 支持过滤功能

举例：

```
class HookAppJava_xxx {
  static func_isShowLog(curStr){
    let isShowLog = true
    // let isShowLog = false

    const excludeList = [
      "clearcut",
      "Font",
      "fonts",
      "Google Sans",
      "Google_Sans",
    ]
    // console.log(`excludeList=${excludeList}`)
    for(const eachExclude of excludeList){
      // console.log(`eachExclude=${eachExclude}`)
      if (curStr.includes(eachExclude)){
        // console.log(`eachExclude=${eachExclude} inside curStr=${curStr} => should ex
clude => is not show log`)
        isShowLog = false
        break
      }
    }

    const includeList = [
      "35___40",
      "633___304",
    ]
    for(const eachInclude of includeList){
      if (curStr.includes(eachInclude)){
        isShowLog = true
        break
      }
    }

    return isShowLog
  }
}

static arwe(func_isShowLog null) {
  var clsName_arwe = "arwe"
  // FridaAndroidUtil.printClassAllMethodsFields(clsName_arwe)

  var cls_arwe = Java.use(clsName_arwe)
  console.log("cls_arwe=" + cls_arwe)
```

```

...
    // public static fnlr k(Context context, boolean z, int i, Integer num, arwl arwlVa
r, eogj eogjVar) {
    //
    var func_arwe_k = cls_arwe.k
    console.log("func_arwe_k=" + func_arwe_k)
    if (func_arwe_k) {
        func_arwe_k.implementation = function (context, z, i, num, arwlVar, eogjVar) {
            var funcName = "arwe.k"
            var funcParaDict = {
                "context": context,
                "z": z,
                "i": i,
                "num": num,
                "arwlVar": arwlVar,
                "eogjVar": eogjVar,
            }

            var isShowLog = true
            if (func_isShowLog != null){
                var ctxStr = context.toString()
                isShowLog = func_isShowLog(ctxStr)
            }

            if (isShowLog){
                FridaAndroidUtil.printFunctionCallAndStack(funcName, funcParaDict)
            }

            var retFnlr = this.k(context, z, i, num, arwlVar, eogjVar)
            if (isShowLog){
                console.log(funcName + " => retFnlr=" + retFnlr)
            }
            return retFnlr
        }
    }
}

HookAppJava_xxx.arwe(HookAppJava_xxx.func_isShowLog)

```

- 实现的效果：
 - 如果堆栈日志中
 - 出现（排除列表中的）clearcut 等内容-》就排除掉= 不打印出来
 - 出现（包含列表中的）35___40 等内容 -》就打印出来

举例：

对于这些日志：

```

===== arwe.k =====
arwe.k: context=com.google.android.xxx.clearcut.service.ClearcutLoggerChimeraService@1e
66c6a, z=true, i=0, num=0, arwlVar=null, eogjVar=null
arwe.k java Stack:

```

```
arwe.k Native Method
at apsb.a com.google.android.xxx 250861022 25.08.61 (150400-731857739) 17
at emtf.a com.google.android.xxx 250861022 25.08.61 (150400-731857739) 30
at apsi.c com.google.android.xxx 250861022 25.08.61 (150400-731857739) 51
at apsi.f com.google.android.xxx 250861022 25.08.61 (150400-731857739) 2047
at bwdy.ft com.google.android.xxx 250861022 25.08.61 (150400-731857739) 1
at bwei.run com.google.android.xxx 250861022 25.08.61 (150400-731857739) 54
at eksq.run com.google.android.xxx 250861022 25.08.61 (150400-731857739) 21
at arvc.c com.google.android.xxx 250861022 25.08.61 (150400-731857739) 50
at arvc.run com.google.android.xxx 250861022 25.08.61 (150400-731857739) 76
at java.util.concurrent.ThreadPoolExecutor.runWorker ThreadPoolExecutor.java 1145
at java.util.concurrent.ThreadPoolExecutor.Worker.run ThreadPoolExecutor.java 644
at asaq.run com.google.android.xxx 250861022 25.08.61 (150400-731857739) 8
at java.lang.Thread.run Thread.java 1012

arwe.k => retFnlr== fnlr 7d77978f
```

实现效果：

- 加上过滤代码前 -》会打印这类日志
- 加上过滤代码后 -》不打印这些日志
- == 排除掉arwe.k的context中的带clearcut字眼的内容
- -》就可以排除掉很多触发太多，但是本身不关注的日志内容了
- -》实现最终目的：减少干扰，提高调试效率

注：已把最新代码，同步到Util函数中：

<https://github.com/crifan/JsFridaUtil/blob/main/frida/FridaAndroidUtil.js>

<https://github.com/crifan/JsFridaUtil/blob/main/frida/FridaHookAndroidJava.js>

crifan.org，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新：
2025-08-22 22:43:30

用logcat辅助找进程

心得：当通过app包名找不到进程时，可以借助于 `logcat` 去搜是否有相关信息，或许对于找进程有所帮助

举例：

比如之前对于：

GSF = Google Services Framework (包名: `com.google.android.gsf`)

要去找进程process

但是通过包名始终找不到：

```
redfin:/data/user/0/com.google.android.gms # ps -A | grep gsf
```

后来就是通过：

```
adb logcat | grep "com.google.android.gsf"
```

而（无意间）发现相关日志：

```
05-22 13:47:51.281 6049 6122 I ActivityManager: Start proc 2149 com.google.process.gservices u0a126 for content provider (com.google.android.gsf com.google.android.gsf.gservices.GservicesProvider)
```

从而注意到其中的：

- `Start proc 2149:com.google.process.gservices/`

很明显，其中：

`com.google.process.gservices`

就是对应process进程的名称

然后通过：

```
redfin:/data/user/0/com.google.android.gms # ps -A | grep gservices
u0_a126      5551  5939  14659388  92704 do_epoll_wait      0 S com.google.process.gservices
```

的确找到了对应进程

->从而确定：

- GSF = Google Services Framework (包名: `com.google.android.gsf`)
 - 的进程名称是: `com.google.process.gservices`

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新：

2025-08-22 22:53:07

字符串反混淆

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2025-05-25 11:47:14

找到并调用字符串解密函数m1.e.i

需求

之前用Frida调试某安卓app逆向期间，遇到：

```
/Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/staticAnalysis/360Wallpaper_jadx_showBadCode/sources/com/w/thsz/info/PaperEntry.java
```

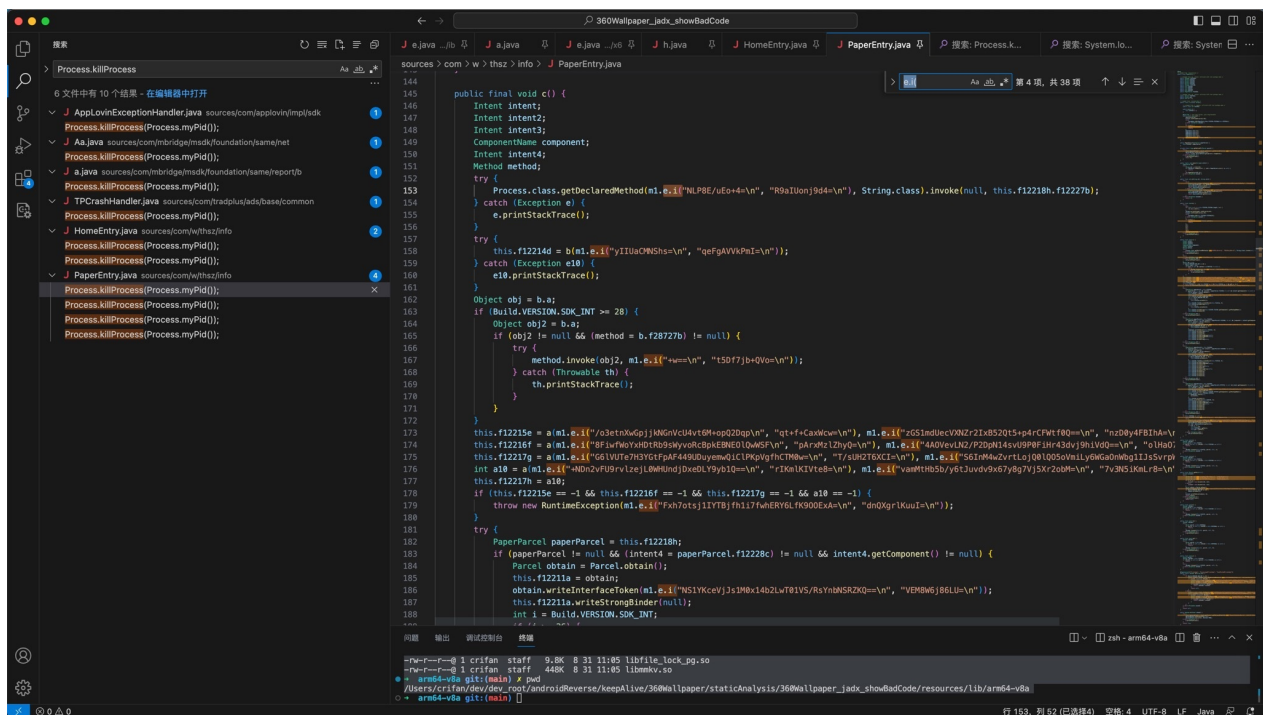
```
Process.class.getDeclaredMethod(m1.e.i("NLP8E/uEo+4=\n", "R9aIUonj9d4=\n"),
String.class).invoke(null, this.f12218h.f12227b);

method.invoke(obj2, m1.e.i("+w==\n", "t5Df7jb+QVo=\n"));

this.f12215e = a(m1.e.i("/o3etnXwGpj jkNGnVcU4vt6M+opQ2Dqp\n", "qt+f+CaxWcw=\n"),
m1.e.i("zGS1mdUecVXNzr2IXB52Qt5+p4rCFwtf0Q==\n", "nzD0y4FBIhA=\n"));

throw new RuntimeException(m1.e.i("Fxh7otsj1IYTBjfh117fwhERY6LfK900ExA=\n",
"dnQXgr1KuuI=\n"));

obtain.writeInterfaceToken(m1.e.i("NS1YKceVjJs1M0x14b2LwT01VS/RsYnbNSRZ
KQ==\n", "VEM8W6j86LU=\n"));
```



发现 `m1.e.i`，是通用的字符串加密解密函数。

而此处，想要搞清楚原始字符串是什么

所以就想要去：给上述加了密的字符串，比如：

- `m1.e.i("/o3etnXwGpj jkNGnVcU4vt6M+opQ2Dqp\n", "qt+f+CaxWcw=\n")`

- `m1.e.i("zGS1mdUecVXNZr2IxB52Qt5+p4rCFWtf0Q==\n", "nzD0y4FBIhA=\n")`
- `m1.e.i("8FiwfWoYxHdRb9sWyvoRcBpkEBNE0lQwWSF\n", "pArxMz1ZhyQ=\n")`
- `m1.e.i("4A0VevLN2/P2DpN14svU9P0FiHr43dvj9hiVdQ==\n", "o1Ha07a0mqA=\n")`
- `m1.e.i("G6lVUTe7H3YgtFpAF449UDuyemwQiClPKpVgfhCTM0w=\n", "T/sUH2T6XCI=\n")`
- `m1.e.i("S6InM4wZvrtLojQ0lQ05oVmily6WGaOnWbg1IJsSvrpw\n", "GPZmYdhG9/U=\n")`
- `m1.e.i("+NDn2vFU9rvlzejL0WHUndjDxeDLy9yb1Q==\n", "rIKm1KIVte8=\n")`
- `m1.e.i("vamMtHb5b/y6tJuvdv9x67y8g7Vj5Xr2obM=\n", "7v3N5iKMLr8=\n")`

去解密字符串

== 字符串反混淆

具体实现：解密字符串

`/Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/dynamicDebug/Frida/frida/hook_360Wallpaper.js`

```
// convert Object(dict/list/...) to JSON string
function toJsonStr(curObj, singleLine false, space 2){
    // console.log("toJsonStr: singleLine=" + singleLine)
    // var jsonStr = JSON.stringify(curObj, null, 2)
    var jsonStr = JSON.stringify(curObj, null, space)
    if(singleLine) {
        // jsonStr = jsonStr.replace(/\n/g, '')
        jsonStr = jsonStr.replace(/\n/g, '')
    }
    return jsonStr
    // return curObj.toString()
}

var decryptedStrDict = {}

// example:
// str1=n+IwzFs/Og7X4DzE\n, str2=+YNTqTlQVWU=, decryptedStr=facebook.com
// str1=C0Y033gie94ALSCKcnQj2g5kDJVwaiqx\n, str2=WhVY8DINF4w= --> decryptedStr=QSV/J
// lRZ8xT8y4VTqTeBg==
function saveDecryptedString(str1, str2, decryptedStr){
    var delimChar = ","
    var curToDecryptStrGroup = str1 + delimChar + str2
    // console.log("curToDecryptStrGroup=" + curToDecryptStrGroup)
    // if (!(curToDecryptStrGroup in decryptedStrDict)){
    if (!decryptedStrDict.hasOwnProperty(curToDecryptStrGroup)){
        decryptedStrDict[curToDecryptStrGroup] = decryptedStr
        console.log("Updated -> decryptedStrDict=" + toJsonStr(decryptedStrDict))
    }
}

// manual decrypt obfuscated string
function manualDecryptString(m1ECls, m1EIFunc){
    // /Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/staticAnalysis/360Wallpaper_jadx_showBadCode/sources/com/w/thsz/info/HomeEntry.java
    // /Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/staticAnalysis/36
```

```

@wallpaper_jadx_showBadCode/sources/com/w/thsz/info/PaperEntry.java
let toDecryptStrArr = [
    // ["o3etnXwGpjKNGnVcU4vt6M+opQ2Dqp\n", "qt+f+CaxWcw=\n"],
    // ["zGS1mdUecVXNZr2IxBS2Qt5+p4rCFWtf0Q==\n", "nzD0y4FBIhA=\n"],
    // ["8F1wfWoYxHdTRb9swyvoRcBpkEBNE0lQwwSF\n", "pArxMz1ZhyQ=\n"],
    // ["4AOVevLN2/P2DpN14svU9P0FiHr43dvj9hiVdQ==\n", "o1Ha07a0mqA=\n"],
    // ["G61VUTe7H3YgtFpAF449UDuyemwQ1ClPKpVgfhCTM0w=\n", "T/sUH2T6XCI=\n"],
    // ["S6InM4wZvrtLojQ0lQ05oVmiLy6WGa0nWbg1IJsSvrpW\n", "GPZmYdhG9/U=\n"],
    // ["+NDn2vFU9rv1zeJL0WHUndjDxeDLY9yb1Q==\n", "rIKm1KIVte8=\n"],
    // ["vamMtHb5b/y6tJuvdv9x67y8g7Vj5Xr2obM=\n", "7v3N5iKMLr8=\n"],

    ["dKEMTz+a/hF0vxgTgbL5S3y5AUkpvvtRdKgNT3Sg7kp3\n", "Fc9oPVDzmj8=\n"],
    ["G96QImCjHHQbwIR+RosblhPGnSR2hXk0G9eRIg==\n", "erD0UA/KeFo=\n"],
    ["QA==\n", "DHIuWtcRHPM=\n"],
    ["97HxJBf07nI=\n", "hNSFZWwpuEI=\n"],
    ["hQG5PXN9gEE=\n", "5GLNVAUU9Dg=\n"],
    ["6ibSaFhtSRDrJNp5SW10B/g8wHtPZlMa9w==\n", "uXKT0gwyG1U=\n"],
    ["qbf04YgsTom/usjUkapBjrSx0+GLvE6Zv6z07g==\n", "6+wBoMXvD9o=\n"],
    ["RGOVY5WGeT9EY4ZkjJx+JVZjnX6PhmQjVnmHcIKNeT5Z\n", "FzfUMcHZMHE=\n"],
    ["a/ZMVX/pp3Zv6AAWcuSsMm3/VFV74aB+b/4=\n", "CpogdR2AyRI=\n"],
    ["L1IU+CJsgCkvTACKBESHcydKGf40SIVpL1sV+A==\n", "Tjxw1k0F5Ac=\n"],
    ["23Kbgc7H69HbbI/d60/si9NqlOfY4+6R23uagQ==\n", "uhz/86Guj/8=\n"],
    ["pMYKuJux1Tck2B7kvZnSaqzeB76NldBwpM8LuA==\n", "xahuyvTYsR4=\n"],
    ["a8JgfYSH7R9y/g==\n", "BowBCe3xiE8=\n"],
    ["jn6X/WJ9JewOYIOhRFUiv4Zmmvt0WSC1jnew/S1HNb6N\n", "7xDzjw0UQcs=\n"],
    ["dx2TBGOGukN3A4dYRa69GX8FngJ1or8DdxSSBA==\n", "FnP3dgzv3m0=\n"],
    ["NLP8E/uEo+4=\n", "R9aIUonj9d4=\n"],
    ["yI IUaCMNShs=\n", "qeFgAVVkpMI=\n"],
    ["+w==\n", "t5Df7jb+QVo=\n"],
    ["zGS1mdUecVXNZr2IxBS2Qt5+p4rCFWtf0Q==\n", "nzD0y4FBIhA=\n"],
    ["4AOVevLN2/P2DpN14svU9P0FiHr43dvj9hiVdQ==\n", "o1Ha07a0mqA=\n"],
    ["S6InM4wZvrtLojQ0lQ05oVmiLy6WGa0nWbg1IJsSvrpW\n", "GPZmYdhG9/U=\n"],
    ["vamMtHb5b/y6tJuvdv9x67y8g7Vj5Xr2obM=\n", "7v3N5iKMLr8=\n"],
    ["Fhx7otsj1IYTbjfh1i7fwhERY6Lfk900ExA=\n", "dnQXgrlKuuI=\n"],
    ["NS1YKceVjJs1M0x14b2LwT01VS/RsYnbNSRZKQ==\n", "VEM8W6j86LU=\n"],
    ["Kb+u3htr/+MpobqCPUP4uSGno9gNT/qjKbav3g==\n", "SNHkrHQcm80=\n"],
    ["ZikFqddEwPhk1xH18WzHomyRCK/BYMW4ZIAEqQ==\n", "Bedh27gtPNY=\n"],
    ["m7XwCB9NrZabq+RU0WwqzJ0t/Q4JaaJwm7zxCA==\n", "+tuUenAkybg=\n"],
    ["DdaB2+RT5g==\n", "bLXi1tJE9kn4=\n"],
    ["Mea6/fs0QEux673g4Qm4GH7Bn+z3CLkFJMW/4fUAqr1026r69g==\n", "UIjeJ5RnzGs=\n"],
    ["wcIRcGqP4zrc3x5hS6vNAeP1EV1aodUA4dUoTlWnwfh/Ck=\n", "lZBQPjn0oG4=\n"],
    ["Pzv0yGsqC/UUP+fPZJk=\n", "S10GrQH1apY=\n"],
    ["5NI4MmJGDgjP1is1YnU=\n", "kLpKVwcZb2s=\n"],
    ["qZTwT62fXkCClPtArQ=\n", "3fyCKsjAPyM=\n"],
    ["u+oaes1wGJO75x1n13cIzvTNP2vBdgnTrskfZsN+Gc8=\n", "2oR+CKIZfL0=\n"],
    ["7EI3G+06AmTzUww66DEVdQ==\n", "iydDWIJuDgE=\n"],
    ["j80jMFDZCyYB3ukRWsIZa43Ii1NR0Qhnn00mN1bGCg==\n", "7q3HQj+wbwI=\n"],
    ["whQiaETnSy/CBA4=\n", "o2drBjCCOUk=\n"],
    ["ee/hgMqVxoN97w==\n", "HoqV06/ns0o=\n"],
    ["vX0MZNDuohKzYEZF2vWwVb92JXfR5qFZr10JYtbxow==\n", "3BNoFr+Hxjw=\n"],
    ["TMkZbwldkwpt2CJMbFaEGw==\n", "K6xtLgYz528=\n"],
    ["fkp5AgYdMph+WlU=\n", "HzkwbHJ4QP4=\n"],
    ["4gCYURyyJQDmAA==\n", "hWXSAnnAU2k=\n"],
    ["7Mysg12Jd1Xo4b1xMA==\n", "gYXf1ET/GAI=\n"],
]
if (m1EIFunc) {

```

```

toDecryptStrArr.forEach((curDecryptRow) => {
    // console.log("curDecryptRow=" + curDecryptRow)
    var str1 = curDecryptRow[0]
    var str2 = curDecryptRow[1]
    // console.log("str1=" + str1 + ", str2=" + str2)
    var str1Stripped = str1.trim()
    var str2Stripped = str2.trim()
    // var decryptdStr = m1EIFunc(str1, str2)
    var decryptdStr = m1ECls.i(str1, str2)
    // console.log("str1=" + str1 + ", str2=" + str2 + " -->> " + decryptdStr)
    console.log(decryptdStr + " <<<-- str1Stripped=" + str1Stripped + ", str2Stripped=" + str2Stripped)
})

// toDecryptStrArr.forEach((item, index, arr) => {
// console.log("item=" + item + ", index=" + index + ", arr=" + arr)

// toDecryptStrArr.forEach((curDecryptRow, curIdx) => {
// console.log("curDecryptRow=" + curDecryptRow + ", curIdx=" + curIdx)
// var str1 = curDecryptRow[0]
// var str2 = curDecryptRow[1]
// console.log("str1=" + str1 + ", str2=" + str2)
// // var decryptdStr = m1EIFunc(str1, str2)
// var decryptdStr = m1ECls.i(str1, str2)
// console.log("str1=" + str1 + ", str2=" + str2 + " -->> " + decryptdStr)
// })
}

}

// 360Wallpaper_jadx_showBadCode/sources/m1/e.java
var m1EClassName = "m1.e"
var m1ECls = Java.use(m1EClassName)
console.log("m1ECls=" + m1ECls)
// printClassAllMethodsFields(m1EClassName)

// print static property value
var m1EC = m1ECls.c
var m1ED = m1ECls.d
var m1EE = m1ECls.e
console.log("m1EC=" + m1EC + ", m1ED=" + m1ED + ", m1EE=" + m1EE)
var m1ECValue = m1EC.value
var m1EDValue = m1ED.value
var m1EEValue = m1EE.value
console.log("m1ECValue=" + m1ECValue + ", m1EDValue=" + m1EDValue + ", m1EEValue=" + m1EEValue)

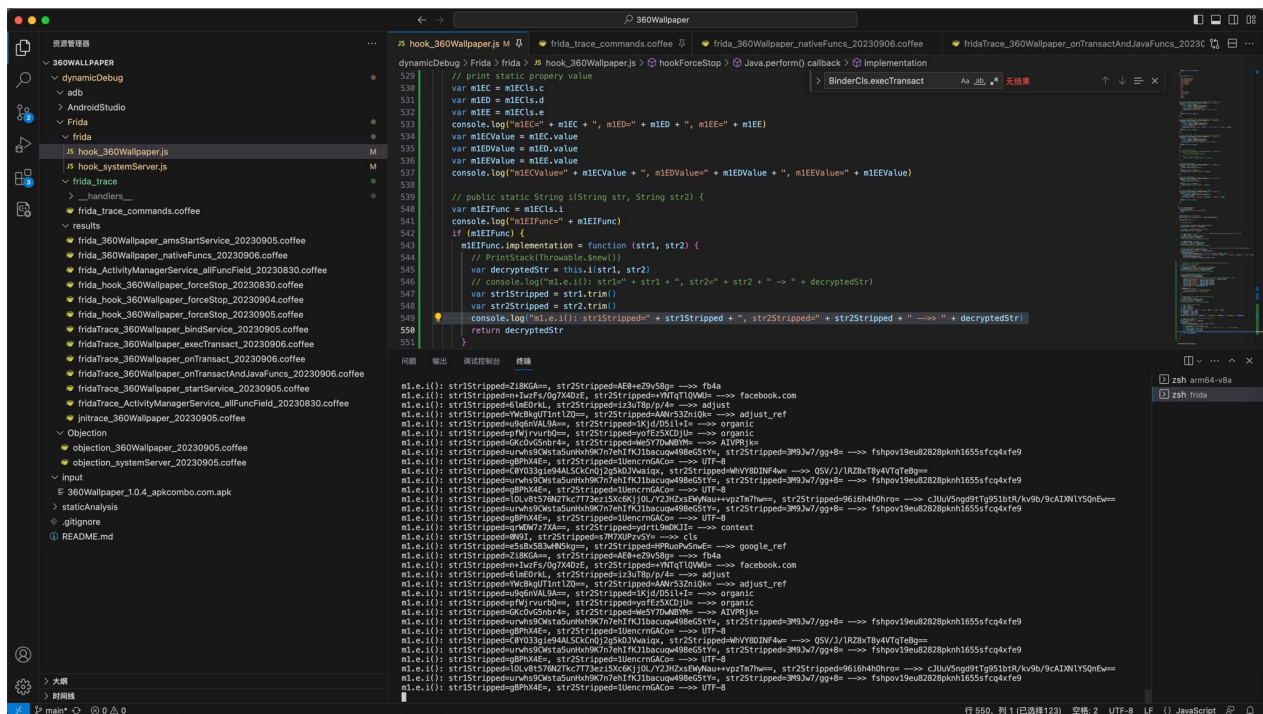
// public static String i(String str, String str2) {
var m1EIFunc = m1ECls.i
console.log("m1EIFunc=" + m1EIFunc)

manualDecryptString(m1EIFunc)

```

```
// hook function
if (m1EIFunc) {
    m1EIFunc.implementation = function (str1, str2) {
        // PrintStack(Throwable.$new())
        var decryptedStr = this.i1(str1, str2)
        // console.log("m1.e.i(): str1=" + str1 + ", str2=" + str2 + " -> " + decryptedStr)
        var str1Stripped = str1.trim()
        var str2Stripped = str2.trim()
        console.log("m1.e.i(): str1Stripped=" + str1Stripped + ", str2Stripped=" + str2Stripped + " -->> " + decryptedStr)
        // saveDecryptedString(str1, str2, decryptedStr)
        return decryptedStr
    }
}
```

效果



Stack: m1.e.i Native Method

```
at a1.c.e Unknown Source 4
at a1.c.d.onReceive OutHelper kt 4
at android.app.LoadedApk ReceiverDispatcher Args lambda getRunnable 0 android.app.L
oadedApk ReceiverDispatcher Args LoadedApk.java 1790
at android.app.LoadedApk ReceiverDispatcher Args ExternalSyntheticLambda0.run Unkn
own Source 2
at android.os.Handler.handleCallback Handler.java 942
at android.os.Handler.dispatchMessage Handler.java 99
at android.os.Looper.loopOnce Looper.java 201
at android.os.Looper.loop Looper.java 288
at android.app.ActivityThread.main ActivityThread.java 7918
at java.lang.reflect.Method.invoke Native Method
at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run RuntimeInit.java 548
```

```

    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:936)
m1.e.i(): str1 e5sBx5B3wHN5kg =
, str2 HPRuoPwSnwE =

...

m1.e.i(): str1Stripped n IwzFs Og7X4DzE, str2Stripped YNTqTlQVWU --> facebook.com
curToDecryptStrGroup n IwzFs Og7X4DzE
, YNTqTlQVWU =

m1.e.i(): str1Stripped 6lmeOrkL, str2Stripped iz3uT8p p/4 --> adjust
curToDecryptStrGroup 6lmeOrkL
, iz3uT8p/p/4 =

m1.e.i(): str1Stripped YwCBkgUT1ntlZQ =, str2Stripped AANr53ZniQk --> adjust_ref
curToDecryptStrGroup YwCBkgUT1ntlZQ =
, AANr53ZniQk =

m1.e.i(): str1Stripped u9q6nVAL9A =, str2Stripped 1Kjd D5il I --> organic
curToDecryptStrGroup u9q6nVAL9A =
, 1Kjd D5il I =

m1.e.i(): str1Stripped pfWjrvurbQ =, str2Stripped yofEz5XCDjU --> organic
curToDecryptStrGroup pfWjrvurbQ =
, yofEz5XCDjU =

m1.e.i(): str1Stripped Zi8KGA =, str2Stripped AE0 eZ9v58g --> fb4a
m1.e.i(): str1Stripped n IwzFs Og7X4DzE, str2Stripped YNTqTlQVWU --> facebook.com
m1.e.i(): str1Stripped 6lmeOrkL, str2Stripped iz3uT8p p/4 --> adjust
m1.e.i(): str1Stripped YwCBkgUT1ntlZQ =, str2Stripped AANr53ZniQk --> adjust_ref
m1.e.i(): str1Stripped u9q6nVAL9A =, str2Stripped 1Kjd D5il I --> organic
m1.e.i(): str1Stripped pfWjrvurbQ =, str2Stripped yofEz5XCDjU --> organic
m1.e.i(): str1Stripped GKcOvG5nhr4 =, str2Stripped We5Y7DwnBYM --> AIVPRjk
m1.e.i(): str1Stripped urwhs9CWsta5unHxh9K7n7ehIfKJ1bacuqw498eG5tY, str2Stripped 3M9Jw7
/gg 8 --> fshpov19eu82828pknh1655sfcq4xfe9
m1.e.i(): str1Stripped gBPhX4E, str2Stripped 1UencrnGACo --> UTF 8
m1.e.i(): str1Stripped C0Y033gie94ALSCKCnQj2g5kDJVwaiqx, str2Stripped WhVY8DINF4w -->
QSV J 1RZ8xT8y4VTqTeBg =
m1.e.i(): str1Stripped urwhs9CWsta5unHxh9K7n7ehIfKJ1bacuqw498eG5tY, str2Stripped 3M9Jw7
/gg 8 --> fshpov19eu82828pknh1655sfcq4xfe9
m1.e.i(): str1Stripped gBPhX4E, str2Stripped 1UencrnGACo --> UTF 8
m1.e.i(): str1Stripped 10Lv8t576N2Tkc7T73ezi5Xc6Kjj0L Y2JHZxsEWyNau vpzTm7hw =, str2St
ripped 96i6h4h0hro --> cJUuV5ngd9tTg951btR kv9b 9cAIXNlYSQnEw =
m1.e.i(): str1Stripped urwhs9CWsta5unHxh9K7n7ehIfKJ1bacuqw498eG5tY, str2Stripped 3M9Jw7
/gg 8 --> fshpov19eu82828pknh1655sfcq4xfe9
m1.e.i(): str1Stripped gBPhX4E, str2Stripped 1UencrnGACo --> UTF 8
m1.e.i(): str1Stripped qrWDW7z7XA =, str2Stripped ydrTL9mDKJI --> context
m1.e.i(): str1Stripped 0N9I, str2Stripped s7M7XUPzvSY --> cls
m1.e.i(): str1Stripped e5sBx5B3wHN5kg =, str2Stripped HPRuoPwSnwE --> google_ref
m1.e.i(): str1Stripped Zi8KGA =, str2Stripped AE0 eZ9v58g --> fb4a

```



```

m1.e.i : str1Stripped n IwzFs Og7X4DzE, str2Stripped YNTqTlQVWU --> facebook.com
m1.e.i : str1Stripped 6lmEOrkL, str2Stripped lz3uT8p p /4 --> adjust
m1.e.i : str1Stripped YwCBkgUT1nt1ZQ, str2Stripped AANr53ZniQk --> adjust_ref
m1.e.i : str1Stripped u9q6nVAL9A, str2Stripped 1Kjd D51l I --> organic
m1.e.i : str1Stripped pFwJrvurbQ, str2Stripped yofEz5XCDjU --> organic
m1.e.i : str1Stripped GKcOvG5nBr4, str2Stripped We5Y7DwNBYM --> AIVPRjk
m1.e.i : str1Stripped urwhs9CWsta5unHxh9K7n7ehIfKJ1bacuqw498eG5tY, str2Stripped 3M9Jw7
/gg 8 --> fshpov19eu82828pknh1655sfcq4xfe9
m1.e.i : str1Stripped gBPhX4E, str2Stripped 1UencrnGACo --> UTF 8
m1.e.i : str1Stripped C0Y033gie94ALSCkCnQj2g5kDJVwaiqx, str2Stripped WhVY8DINF4w -->
QSV J lRZ8xT8y4VTqTeBg
m1.e.i : str1Stripped urwhs9CWsta5unHxh9K7n7ehIfKJ1bacuqw498eG5tY, str2Stripped 3M9Jw7
/gg 8 --> fshpov19eu82828pknh1655sfcq4xfe9
m1.e.i : str1Stripped gBPhX4E, str2Stripped 1UencrnGACo --> UTF 8
m1.e.i : str1Stripped lOLv8t576N2Tkc7T73ezi5Xc6KjjOL Y2JHZxsEWyNau vpzTm7hw, str2St
ripped 96i6h4h0hro --> cJUuV5ngd9tTg951btR kv9b 9cAIXNlYSQnEw
m1.e.i : str1Stripped urwhs9CWsta5unHxh9K7n7ehIfKJ1bacuqw498eG5tY, str2Stripped 3M9Jw7
/gg 8 --> fshpov19eu82828pknh1655sfcq4xfe9
m1.e.i : str1Stripped gBPhX4E, str2Stripped 1UencrnGACo --> UTF 8

```

```

Updated -> decryptedStrDict={"@mIyyw==\n,pQNbvxzsUjA=\n":"wait","qrWDW7z7XA==\n,ydrTL9m
DKJI=\n":"context"}

```

```

Updated -> decryptedStrDict={"@mIyyw==\n,pQNbvxzsUjA=\n":"wait","qrWDW7z7XA==\n,ydrTL9m
DKJI=\n":"context","@N9I\n,s7M7XUPzvSY=\n":"cls"}

```

```

...

```

```

Updated -> decryptedStrDict={
  "RHk=\n,NEk26SxLQpY=\n": "p0",
  "Ctg=\n,euj9oN1089Q=\n": "p0",
  "WNoEx1td\n,ZK5sr1hjJHA=\n": "<this>",
  "6to15TJn\n,xK1VtlcTvZ4=\n": ".wpSet",
  "yBAkZWj1VXzbuzRjbuJWIYaNBUNTxx8V+w==\n,qN5AFweMMVI=\n": "android.settings.SETTINGS",
  "iznC7BGVypmSDsLzOg==\n,/1Gx1k7zuPY=\n": "thsz_from_set",
  "MzhD1Fyu+7sqD0PLdw==\n,R1AwrGPIIdQ=\n": "thsz_from_set",
  "TD0nYsjkP49U0AxS1vQ8\n,O014DbiBUdA=\n": "wp_open_out_num",
  "NIU=\n,RLXuws1cso4=\n": "p0",
  "v68=\n,z59QGAfzgVI=\n": "p0",
  "10g=\n,p3jUwVR1T1E=\n": "p0",
  "bTY=\n,HQYlg2SZsI8=\n": "p0",
  "+pM=\n,iqNagnhpDJw=\n": "p0",
  "+mc=\n,ilcKLu3y9Hc=\n": "p0",
  "qfk=\n,2ckbQ5GN/s8=\n": "p0",
  "aog=\n,GrhJtEmf1VA=\n": "p0",
  "qjI=\n,2gK64i8EPUY=\n": "p0",
  "ZNg=\n,FOgMtYm5CFA=\n": "p0",
  "+5/dnwJs0ULhmcau\n,iPqpWGMcoR0=\n": "set_app_icon",
  "b4w=\n,H7wXwoPm++I=\n": "p0",
  "Z3U=\n,F0R7J/92o8c=\n": "p1",
  "@mIyyw==\n,pQNbvxzsUjA=\n": "wait",
  "qrWDW7z7XA==\n,ydrTL9mDKJI=\n": "context",
  "@N9I\n,s7M7XUPzvSY=\n": "cls",
  "e5sBx5B3wHN5kg==\n,HPRuoPwSnwE=\n": "google_ref",

```

```

"Zi8KGA==\n,AE0+eZ9v58g=\n": "fb4a",
"n+IwzFs/Og7X4DzE\n,+YNTqTlQVWU=\n": "facebook.com",
"6ImE0rkL\n,iz3uT8p/p/4=\n": "adjust",
"YwCbkgUT1ntlZQ==\n,AANr53ZniQk=\n": "adjust_ref",
"u9q6nVAL9A==\n,1Kjd/D5i1+I=\n": "organic",
"pfWjrvurbQ==\n,yofEz5XCDjU=\n": "organic",
"GKcOvG5nbr4=\n,we5Y7DwNBYM=\n": "AIVPRjk=",
"urwhs9CWsta5unHxh9K7n7ehIfKJ1bacuqw498eG5tY=\n,3M9Jw7/gg+8=\n": "fshpov19eu82828pknh1655sfcq4xfe9",
"gBPhX4E=\n,1UencrnGACo=\n": "UTF-8",
"COY033gie94ALSCKCnQj2g5kDJVwaiqx\n,WhVY8DINF4w=\n": "QSV/J/1RZ8xT8y4VTqTeBg==",
"lOLv8t576N2Tkc7T73ezi5Xc6KjjOL/Y2JHZxsEWyNau++vpzTm7hw==\n,96i6h4h0hro=\n": "cJUuV5ngd9tTg951btR/kv9b/9cAIXNlYSQnEw==",
"lGseWqSRpxScawZtpYztW5ZxG2e110B2u1Y3V5ihkG6wSC1MgrmPdbJW\n,9QVyCMv4wzo=\n": "android.intent.action.CLOSE_SYSTEM_DIALOGS"
}

```

» 手动解密:

```

m1EIFunc=function e() {
    [native code]
}

TRANSACTION_startService <<<--- str1Stripped /o3etnXwGpjkkNGnVcU4vt6M-opQ2Dqp, str2Stripped qt f CaxWcw
START_SERVICE_TRANSACTION <<<--- str1Stripped zGS1mdUecVXNZr2IxBS2Qt5 p4rCFWtf0Q==, str2Stripped nzD0y4FBIhA
TRANSACTION_broadcastIntent <<<--- str1Stripped 8F1wfWoYxHdTRb9sWyvoRcBpkEBNE0lQwWSF, str2Stripped pArxMz1ZhyQ
BROADCAST_INTENT_TRANSACTION <<<--- str1Stripped 4A0VevLN2 P2Dpn14svU9P0FiHr43dvj9hiVdQ==, str2Stripped olHa07a0mqA
TRANSACTION_startInstrumentation <<<--- str1Stripped G6lVUTE7H3YGtFpAF449UDuyemwQic1PKpVgfhCTM0w, str2Stripped T sUH2T6XCI
START_INSTRUMENTATION_TRANSACTION <<<--- str1Stripped S6InM4wZvrtLojQ0lQ05oVm1Ly6WGa0Nwbgi1JsSvrpw, str2Stripped GPZmYdhG9 U
TRANSACTION_startActivity <<<--- str1Stripped NDn2vFU9rvlzejL0WHUndjDxeDLy9yb1Q==, str2Stripped rIKmlKIVte8
START_ACTIVITY_TRANSACTION <<<--- str1Stripped vamMtHb5b y6tJuvdv9x67y8g7Vj5Xr2obM, str2Stripped 7v3N5iKmlr8

```

/Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/staticAnalysis/360Wallpaper_jadx_showBadCode/sources/com/w/thsz/info/

- PaperEntry.java
- HomeEntry.java

» 字符串解密解码结果:

```

m1EIFunc=function e() {
    [native code]
}

android.app.IActivityManager.Stub <<<--- str1Stripped dKEMTz a hF0vxgTGBL5S3y5AUKpvvtRdKgNT3Sg7kp3, str2Stripped Fc9oPVDzmj8
android.app.IActivityManager <<<--- str1Stripped G96QImCjHHQbwIR RosbLhPGnSR2hXk0G9eRig==, str2Stripped erD0UA KeFo
L <<<--- str1Stripped QA==, str2Stripped DHIuwtCRHPM

```

```

setArgv0 <<<--- str1Stripped 97HxJBf07nI, str2Stripped hNSFZWwpuEI
activity <<<--- str1Stripped hQG5PXN9gEE, str2Stripped 5GLNVAUU9Dg
START_SERVICE_TRANSACTION <<<--- str1Stripped 6ibSaFhtSRDrJNp5SW10B g8wHtPZlMa9w, str
2Stripped uXKT0gwyG1U
BROADCAST_INTENT_TRANSACTION <<<--- str1Stripped qbf04YGsTom usjukapBjrSx0 GLvE6Zv6z07g
, str2Stripped 6 WBoMXvD9o
START_INSTRUMENTATION_TRANSACTION <<<--- str1Stripped RGOVY5WGeT9EY4ZkjJx JVZjnX6PhmQjV
nmHcIKNeT5Z, str2Stripped FzfUMchZMHE
all binder code get failed <<<--- str1Stripped a ZMVX pp3Zv6AAWcuSsMm3 VFV74aB b 4, st
r2Stripped CpogdR2AyRI
android.app.IActivityManager <<<--- str1Stripped L1IU CJsGckvTACKBESHcydKGf40SIVpL1sV A
, str2Stripped Tjxw1k0F5Ac
android.app.IActivityManager <<<--- str1Stripped 23Kbgc7H69HbbI d60 si9Nql0fY4 6R23uagQ
, str2Stripped uhz 86GuJ 8
android.app.IActivityManager <<<--- str1Stripped pMYKuJux1TCk2B7kvZnSaqzeB76NldBwpM8LuA
, str2Stripped xahuYvTYsR4
mNativePtr <<<--- str1Stripped a8JgfYSH7R9y g, str2Stripped BowBCe3xiE8
android.app.IActivityManager.Stub <<<--- str1Stripped jn6X WJ9JewOYiOhRFUiv4Zmmvt0WSClj
new S1HNB6N, str2Stripped 7xDzjw0UQcs
android.app.IActivityManager <<<--- str1Stripped dx2TBG0GukN3A4dYRa69GX8FngJ1or8DdxSSBA
, str2Stripped FnP3dgzv3m0
setArgv0 <<<--- str1Stripped NLP8E uEo 4, str2Stripped R9aIUonj9d4
activity <<<--- str1Stripped yIIUaCMNShs, str2Stripped qeFgAVVkpMI
L <<<--- str1Stripped w, str2Stripped t5Df7jb QVo
START_SERVICE_TRANSACTION <<<--- str1Stripped zGS1mdUecVXNZr2IxBS2Qt5 p4rCFWtf0Q, str
2Stripped nzD0y4FBIhA
BROADCAST_INTENT_TRANSACTION <<<--- str1Stripped 4A0VevLN2 P2DpN14svU9P0FiHr43dvj9hiVdQ
, str2Stripped olHa07a0mqA
START_INSTRUMENTATION_TRANSACTION <<<--- str1Stripped S6InM4wZvrtLojQ0lQ05oVm1Ly6WGa0nW
bg1IJsSvrpw, str2Stripped GPZmYdhG9 U
START_ACTIVITY_TRANSACTION <<<--- str1Stripped vamMtHb5b y6tJuvdv9x67y8g7Vj5Xr2obM, st
r2Stripped 7v3N5iKmlr8
all binder code get failed <<<--- str1Stripped Fxh7otsj1IYTBjfh1i7fwhERY6LfK900ExA, st
r2Stripped dnQXgrlKuuI
android.app.IActivityManager <<<--- str1Stripped NS1YKceVjJs1M0x14b2LwT01VS RsYnbNSRZKQ
, str2Stripped VEM8W6j86LU
android.app.IActivityManager <<<--- str1Stripped Kb u3htr MpobqCPUP4uSGno9gNT qjKbav3g
, str2Stripped SNHkrHQCm80
android.app.IActivityManager <<<--- str1Stripped ZikFqddEwPhklxH18WzHomyRCK BYMW4ZIAEqQ
, str2Stripped Bedh27gtpNY
android.app.IActivityManager <<<--- str1Stripped m7XwCB9NrZabq RUOWWqzJ0t Q4JaaJWm7zxCA
, str2Stripped tuUenAkybg
account <<<--- str1Stripped DdaB2 RT5g, str2Stripped bLXitJE9kn4
android.accounts.IAccountManager.Stub <<<--- str1Stripped Mea6 fsOqEUx673g4Qm4GH7Bn z3C
LkFJMW 4fUAqRl026r69g, str2Stripped UIjeJ5RnzGs
TRANSACTION_removeAccountExplicitly <<<--- str1Stripped wcIRcGqP4zrc3x5hS6vNAeP1EV1aodu
A4dUoTlwnwwfh Ck, str2Stripped lZBQPjn0oG4
three_ac_label <<<--- str1Stripped Pzv0yGSqC UUP fPZJk, str2Stripped S10GrQH1apY
three_ac_label <<<--- str1Stripped 5NI4MmJGDgjP1is1YnU, str2Stripped kLPKvWcZb2s
three_ac_type <<<--- str1Stripped qZTwT62fXkCCiPtArQ, str2Stripped 3fyCKsjAPyM
android.accounts.IAccountManager <<<--- str1Stripped u oaes1wGJO75x1n13cIzvTNP2vBdgnTrs
kfZsN Gc8, str2Stripped 2oR CKIZfL0
getContextObject <<<--- str1Stripped 7EI3G 06AmTzUww66DEVdQ, str2Stripped iydDWIJudGE
android.os.ServiceManagerNative <<<--- str1Stripped j80jMFDZCyyB3ukRwsIZa43Ii1NR0qhnn00
mNlbGCg, str2Stripped 7q3HQj wbwI

```



```
asInterface <<<--- str1Stripped whQiaETnSy CBA4, str2Stripped o2drBjCCOUk
getService <<<--- str1Stripped ee hgMqVxoN97w, str2Stripped HoqV06 ns0o
android.os.ServiceManagerNative <<<--- str1Stripped vx0MZNDuohKzYEF2vWwVb92JXfR5qFZr10
JYtbxow, str2Stripped 3BNoFr Hxjw
getContextObject <<<--- str1Stripped TMkZbwldkwpT2CJMbFaEGw, str2Stripped K6xtLgYZ528

asInterface <<<--- str1Stripped fkp5AgYdMph w1U, str2Stripped HzkwbHJ4QP4
getService <<<--- str1Stripped 4gCYURyyJQDMAA, str2Stripped hWXsAnnAU2k
mIsVivowidget <<<--- str1Stripped 7Mysgi2Jd1Xo4bixMA, str2Stripped gYXf1ET GAI
m1.e: Found instance object m1.e b55a893
m1.e: Found instance object m1.e 187c0d0
m1.e: Found instance object m1.e 46ca8c9
m1.e: onComplete
```

即可自动解密，得到我们要的原始的字符串的值了。

crifan.org，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新：
2025-12-13 22:25:49

用JEB自动反编译字符串

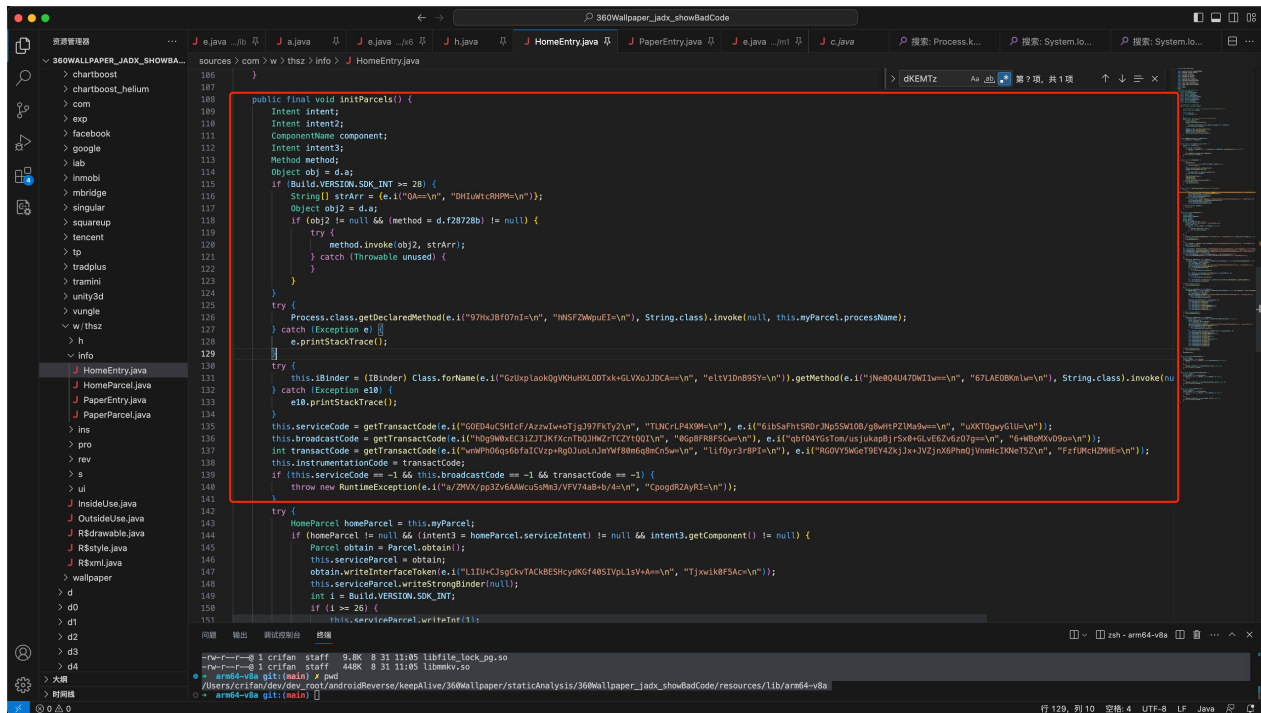
用JEB反编译代码时，默认自动已开启：字符串反混淆=字符串反编译

举例：

效果对比

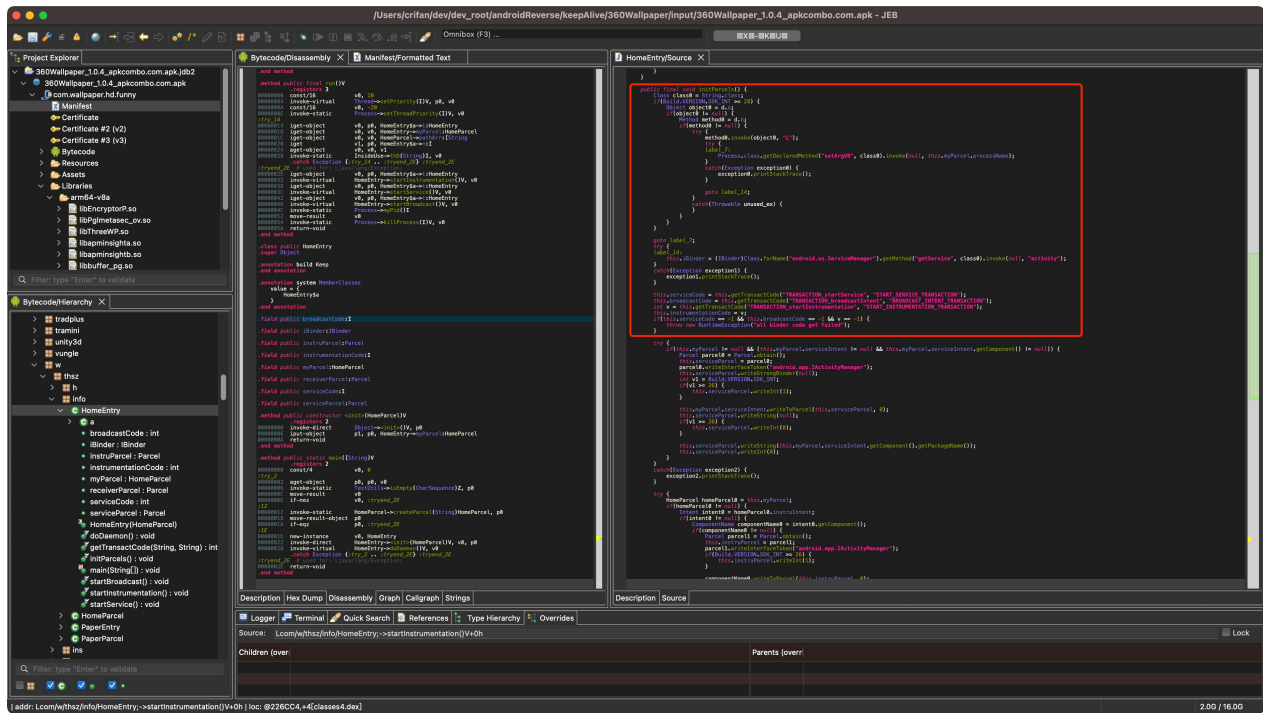
jadx：无法自动反混淆字符串

```
public final void initParcels() {  
    ...  
        Process.class.getDeclaredMethod(e.i("97HxJBf07nI==\n", "hNSFZWwpuEI==\n"), String.class).invoke(null, this.myParcel.processName);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    try {  
        this.iBinder = (IBinder) Class.forName(e.i("GzUxplaokQgVKHuHXL0DTxk+GLVXoJJ  
DCA==\n", "eltV1DnB9SY==\n")).getMethod(e.i("jNe0Q4U47DWI1w==\n", "67LAE0BKmlw==\n"), String.class).invoke(null, e.i("hQG5PXN9gEE==\n", "5GLNVAUU9Dg==\n"));  
    } catch (Exception e10) {  
        e10.printStackTrace();  
    }  
    this.serviceCode = getTransactCode(e.i("GOED4uC5HIcF/AzzwIw+oTjgJ97FkTy2\n", "T  
LNCrLP4X9M==\n"), e.i("61bSaFhtSRDrJNp5SW10B/g8wHtPZlMa9w==\n", "uXKT0gwyG1U==\n"));  
    this.broadcastCode = getTransactCode(e.i("hDg9W0xEC3iZJTJKfXcnTbQJHWZrTCZYtQQI\  
n", "0Gp8FR8FSCw==\n"), e.i("qbf04YGsTom/usjukapBjrSx0+GLvE6Zv6z07g==\n", "6+wBoMXvD9o=\  
n"));  
    int transactCode = getTransactCode(e.i("wnWPh06qs6bfaICVzp+RgOJuoLnJmYwf80m6q8m  
Cn5w==\n", "lif0yr3r8PI==\n"), e.i("RG0VY5WGeT9EY4ZkjJx+JVZjnX6PhmQjVnmHcIKNeT5Z\n", "Fzf  
UMcHZMHE==\n"));  
}
```



JEB：可以自动反混淆字符串 = 还原出原始字符串

```
public final void initParcels() {  
    ...  
    Process.class.getDeclaredMethod("setArgv0", class0).invoke(  
null, this.myParcel.processName);  
    ...  
}  
...  
this.iBinder = (IBinder)Class.forName("android.os.ServiceManager").getMethod(  
("getService", class0).invoke(null, "activity");  
...  
this.serviceCode = this.getTransactCode("TRANSACTION_startService", "START_SERV  
ICE_TRANSACTION");  
this.broadcastCode = this.getTransactCode("TRANSACTION_broadcastIntent", "BROAD  
CAST_INTENT_TRANSACTION");  
int v = this.getTransactCode("TRANSACTION_startInstrumentation", "START_INSTRUM  
ENTATION_TRANSACTION");  
}
```



crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2025-05-25 11:55:40

Frida调试Android常见问题

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2025-12-13 21:50:16

Frida中的Java和js变量类型的映射关系

此处Java中的变量类型，转换成Frida中的类型的写法是：

Frida（的js）中的变量类型，其底层来自于**Dex**

所以其实就是：

- Frida的类型 == Dex中的类型 的语法

即：

- Java变量类型 -> Frida=Dex变量类型
 - 非数组列表
 - 简单类型
 - 完整写法
 - `int` `int`
 - `byte` `byte`
 - `short` `short`
 - `long` `long`
 - `float` `float`
 - `double` `double`
 - `char` `char`
 - -> 缩写 ShortyDescriptor
 - 缩写 说明
 - `V` void; 仅对返回类型有效
 - `Z` boolean
 - `B` byte
 - `S` short
 - `C` char
 - `I` int
 - `J` long
 - `F` float
 - `D` double
 - 特殊的对象Object == 带父类，带路径的Java的类：前缀L + 点. 变成 斜杠/ + 后缀分号;
 - 语法： `Lfully/qualified/Name;`
 - 说明：类 `fully.qualified.Name`
 - 举例
 - `Ljava/lang/String;`
 - `Ljava/lang/Object;`
 - `Ljava/lang/System;`
 - `Ljava/io/PrintStream;`
 - `Ljava/net/InetAddress;`
 - `Ljava/time/LocalDateTime;`
 - `Ljava/lang/ClassLoader;`
 - 列表=数组：前面再加个：左中括号 `[`

- 语法
 - [descriptor
- 说明
 - descriptor 的数组，可递归地用于“数组的数组”，但维数不能超过 255
- 举例
 - 普通类型：字母缩写
 - `int[]` -> `[I`
 - `byte[]` -> `[B`
 - `short[]` -> `[S`
 - `long[]` -> `[J`
 - `float[]` -> `[F`
 - `double[]` -> `[D`
 - `char[]` -> `[C`
 - 特殊类型：用（Frida=Dex中的）完整类名写法
 - `[Ljava/lang/String;`
 - `[Ljava/lang/Object;`
 - `[Ldalvik/system/DexPathList$Element;`

且已整理到相关工具类代码中：

- <https://github.com/crifan/JsFridaUtil/blob/main/frida/FridaAndroidUtil.js>

```
// https://source.android.com/docs/core/runtime/dex-format?hl=zh-cn
// https://cmrodriguez.me/blog/methods/
static FridaDexTypeMappingDict = {
  "void": "V",

  "boolean": "Z",
  "char": "C",
  "byte": "B",
  "short": "S",
  "int": "I",
  "long": "J",
  "float": "F",
  "double": "D",

  // from FridaDexTypeMappingDict_list

  // TODO: add more type
}

static FridaDexTypeMappingDict_list = {
  "char[]": "[C",
  "byte[]": "[B",
  "short[]": "[S",
  "int[]": "[I",
  "long[]": "[J",
  "float[]": "[F",
  "double[]": "[D",

  "String[]": "[Ljava/lang/String;",
  "Object[]": "[Ljava/lang/Object;",
```

```
}
```

含义 = 使用场景举例：

(1) Java相关：函数的签名=定义

举例1：

[runtime/native/dalvik_system_DexFile.cc - platform/art - Git at Google \(googlesource.com\)](https://source.android.com/platform/art)

```
NATIVE_METHOD(DexFile, createCookieWithArray, "([BII)Ljava/lang/Object;"),
```

->

- ([BII)Ljava/lang/Object;
 - ([BII)
 - 括号内是函数参数
 - [BII
 - [B = byte[] = 字节码数组列表
 - I = int
 - I = int
 - Ljava/lang/Object;
 - 类型： Object

-> 所以函数的完整定义 类型 就是这样的：

```
Object createCookieWithArray(byte[], int, int)
```

举例2：

```
NATIVE_METHOD(DexFile, isProfileGuidedCompilerFilter, "(Ljava/lang/String;)Z"),
```

->

- (Ljava/lang/String;)Z
 - (Ljava/lang/String;)
 - Ljava/lang/String;
 - Z

->

```
void isProfileGuidedCompilerFilter(String)
```

(2) Frida中

Frida中，定义Object数组的写法：

```
var JavaObjArr = Java.use("[Ljava.lang.Object;")
console.log("JavaObjArr=" + JavaObjArr)
```


输出：

```
JavaObjArr=<class: [Ljava.lang.Object;>
```

把变量转换为Object的数组：

```
var objArr = Java.cast(obj, JavaObjArr)
console.log("objArr=" + objArr)
```

输出：

```
objArr=[Ljava.lang.Object;@fc9e9af
```

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2025-12-13 21:51:34

Error java.lang.ClassNotFoundException Didn't find class on path DexPathList zip file

情况1：类名错了

- 现象

hook报错：

```
Error: java.lang.ClassNotFoundException: Didn't find class zzz.xxx on path: DexPathList[zip file]
```

即：找不到类名

- 原因：类名写错了
- 根本原因：
 - 之前Jadx反编译出的源码中的文件名 `sources/zzz/xxx.java` 中的 `xxx` 不是真正的类名
 - 而注释中 `renamed from: yyy` 才是真正的类名
- 解决办法：把名字改为真正的Java的类名： `yyy`

情况2：类不在当前hook的二进制中

- 背景

安卓逆向期间，去根据之前Logcat日志：

```
2023-08-31 10:26:23.911 6481-6481 AppButtonsPrefCtl com.android.settings
D Stopping package com.wallpaper.hd.funny
```

找到了安卓内部的类： `com.android.server.pm.Settings` 中的函数： `createNewSetting` ，输出了上述的log

- 现象

去Frida中用js代码：

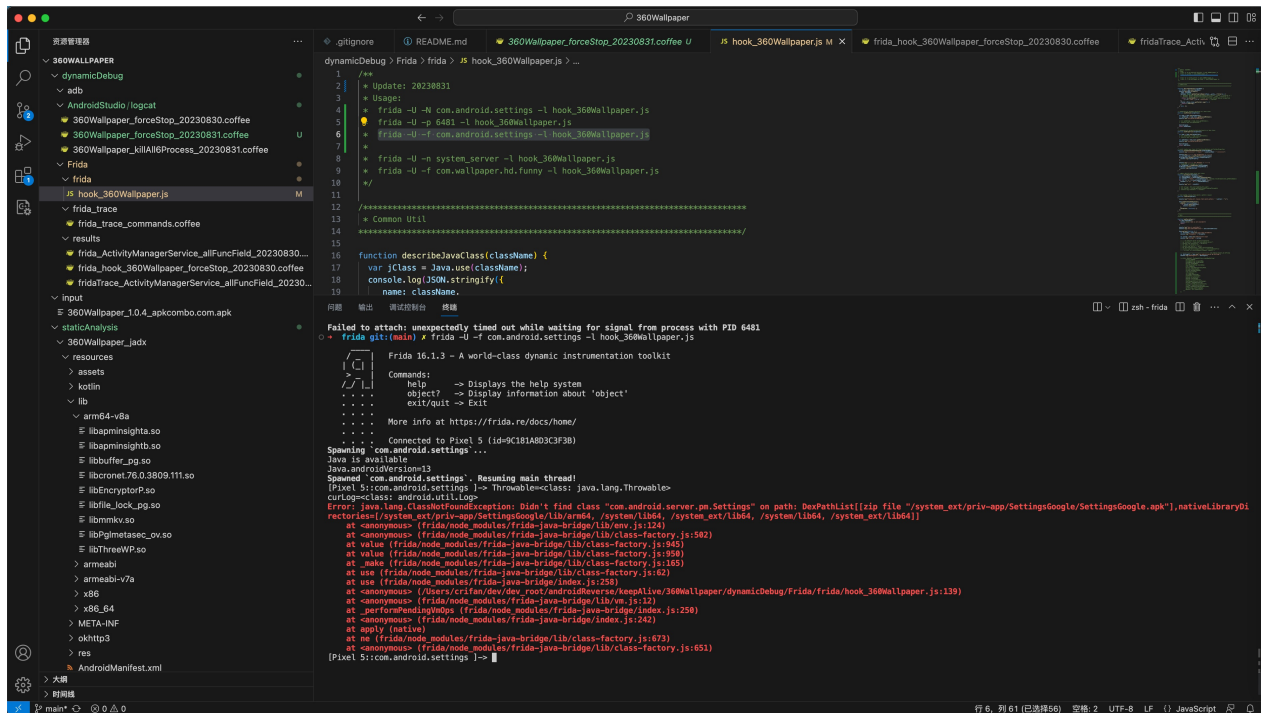
```
var SettingsCls = Java.use("com.android.server.pm.Settings")
```

去hook安卓内部的类： `com.android.server.pm.Settings`

```
frida -U -f com.android.settings -l hook_360Wallpaper.js
```

报错：

```
Error: java.lang.ClassNotFoundException: Didn't find class "com.android.server.pm.Settings" on path: DexPathList[[zip file "/system_ext/priv-app/SettingsGoogle/SettingsGoogle.apk"],nativeLibraryDirectories=[/system_ext/priv-app/SettingsGoogle/lib/arm64, /system_ext/priv-app/SettingsGoogle/lib/arm64, /system_ext/lib64, /system/lib64, /system_ext/lib64]]
    at <anonymous> (frida/node_modules/frida-java-bridge/lib/env.js:124)
    at <anonymous> (frida/node_modules/frida-java-bridge/lib/class-factory.js:502)
    at value (frida/node_modules/frida-java-bridge/lib/class-factory.js:945)
    at value (frida/node_modules/frida-java-bridge/lib/class-factory.js:950)
    at _make (frida/node_modules/frida-java-bridge/lib/class-factory.js:165)
    at use (frida/node_modules/frida-java-bridge/lib/class-factory.js:62)
    at use (frida/node_modules/frida-java-bridge/index.js:258)
    at <anonymous> (/Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/dynamicDebug/Frida/frida/hook_360Wallpaper.js:139)
    at <anonymous> (frida/node_modules/frida-java-bridge/lib/vm.js:12)
    at _performPendingVmOps (frida/node_modules/frida-java-bridge/index.js:250)
    at <anonymous> (frida/node_modules/frida-java-bridge/index.js:242)
    at apply (native)
    at ne (frida/node_modules/frida-java-bridge/lib/class-factory.js:673)
    at <anonymous> (frida/node_modules/frida-java-bridge/lib/class-factory.js:651)
```



- 原因
 - 此处安卓系统的app: 设置, 包名: `com.android.settings`
 - 好像没包含, 此处输出的Stopping package的日志的代码:
 - 类 `com.android.server.pm.Settings` 的函数 `createNewSetting`
 - 因为是另外的二进制程序 `system_server` 才包含此函数
- 解决办法

经过实测, 换 `system_server` 去hook, 即可找到。

- 具体步骤

```
frida -U -n system_server -l hook_360Wallpaper.js
```

即可找到该类，正常输出：

```
SettingsCls=>class: com.android.server.pm.Settings
```

情况3：不能用当前ClassLoader，要换ClassLoader才能找到

问题=现象

Frida的hook代码：

```
static CronetURLRequest() {
    var clsName_CronetURLRequest = "org.chromium.net.impl.CronetURLRequest"
    FridaAndroidUtil.printClassAllMethodsFields(clsName_CronetURLRequest)

    var cls_CronetURLRequest = Java.use(clsName_CronetURLRequest)
    console.log("cls_CronetURLRequest=" + cls_CronetURLRequest)
    ...
}
```

但是报错，找不到类：

```
Error: java.lang.ClassNotFoundException
```

```
Error: java.lang.ClassNotFoundException: Didn't find class "CronetURLRequest" on path:
DexPathList[[zip file "/data/app/~~12Pi7u0lbbj7Dqzrw11Eww==/com.google.android.xxx-zSF0
DEP1E-FZv8eQIo1iGQ==/base.apk"],nativeLibraryDirectories=[/data/app/~~12Pi7u0lbbj7Dqzrw
11Eww==/com.google.android.xxx-zSF0DEP1E-FZv8eQIo1iGQ==/lib/arm64, /data/app/~~12Pi7u0l
bbj7Dqzrw11Eww==/com.google.android.xxx-zSF0DEP1E-FZv8eQIo1iGQ==/base.apk!/lib/arm64-v8
a, /system/lib64, /system_ext/lib64]]
    at <anonymous> (frida/node_modules/frida-java-bridge/lib/env.js:124)
    at <anonymous> (frida/node_modules/frida-java-bridge/lib/class-factory.js:502)
    at value (frida/node_modules/frida-java-bridge/lib/class-factory.js:949)
    at value (frida/node_modules/frida-java-bridge/lib/class-factory.js:954)
    at _make (frida/node_modules/frida-java-bridge/lib/class-factory.js:165)
    at use (frida/node_modules/frida-java-bridge/lib/class-factory.js:62)
    at use (frida/node_modules/frida-java-bridge/index.js:256)
    at enumProperties (/Users/crifan/dev/dev_root/androidReverse/xxx/xxx/xxx_v250861022
/dynamicDebug/frida/frida_js/hook_xxx.js:2335)
    at printClassAllMethodsFields (/Users/crifan/dev/dev_root/androidReverse/xxx/xxx/xx
x_v250861022/dynamicDebug/frida/frida_js/hook_xxx.js:2353)
    at CronetURLRequest (/Users/crifan/dev/dev_root/androidReverse/xxx/xxx/xxx_v2508610
22/dynamicDebug/frida/frida_js/hook_xxx.js:5886)
    at hookApp_AndroidJava (/Users/crifan/dev/dev_root/androidReverse/xxx/xxx/xxx_v2508
61022/dynamicDebug/frida/frida_js/hook_xxx.js:30797)
    at hookApp (/Users/crifan/dev/dev_root/androidReverse/xxx/xxx/xxx_v250861022/dynami
cDebug/frida/frida_js/hook_xxx.js:30975)
    at <anonymous> (/Users/crifan/dev/dev_root/androidReverse/xxx/xxx/xxx_v250861022/dy
namicDebug/frida/frida_js/hook_xxx.js:30995)
    at <anonymous> (frida/node_modules/frida-java-bridge/lib/vm.js:12)
    at _performPendingVmOps (frida/node_modules/frida-java-bridge/index.js:248)
    at <anonymous> (frida/node_modules/frida-java-bridge/index.js:223)
    at <anonymous> (frida/node_modules/frida-java-bridge/lib/vm.js:12)
    at _performPendingVmOpsWhenReady (frida/node_modules/frida-java-bridge/index.js:242)
```

```

    at perform (frida/node_modules/frida-java-bridge/index.js:202)
    at hookAndroid (/Users/crifan/dev/dev_root/androidReverse/xxx/xxx/xxx_v250861022/dynamicDebug/frida/frida_js/hook_xxx.js:30998)
    at apply (native)
    at <anonymous> (frida/runtime/core.js:51)

```

原因

此处内部是用了另外的机制classLoader，去加载此类，导致此处默认的方式，找不到此类

解决办法

去换用别的特定的classLoader，即可正常加载该类，即可找到该类

具体代码

```

class FridaAndroidUtil{

    static findClassLoader(className){
        var foundClassLoader = null

        const classLoaders = Java.enumerateClassLoadersSync()
        // console.log("classLoaders=" + classLoaders + ", type=" + (typeof classLoaders))

        for (const loaderIdx in classLoaders) {
            var curClassLoader = classLoaders[loaderIdx]
            // var loaderClsName = FridaAndroidUtil.getJavaClassName(curClassLoader)
            // console.log(`[${loaderIdx}] loaderClsName=${loaderClsName}, curClassLoader=${curClassLoader}`)

            try {
                if (curClassLoader.findClass(className)){
                    // var loaderStr = curClassLoader.toString()
                    // console.log(`Found ${className} in loader ${loaderStr}`)
                    // Found org.chromium.net.impl.CronetURLRequest in loader dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk"],nativeLibraryDirectories=[/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]]

                    // if (loaderStr.includes(apkName)){
                    //     // console.log(`Update current Java.classFactory.loader to ${loaderStr}`)

                    //     Java.classFactory.loader = curClassLoader
                    //     break
                    // }

                    console.log(`Found ${className} in loader ${curClassLoader}`)
                    foundClassLoader = curClassLoader
                    break
                }
            }
        }
    }
}

```

```

    } catch (err){
        // console.log(`${err}`)
    }
}

console.log(`findClassLoader: className=${className} => foundClassLoader=${foundClassLoader}`)
return foundClassLoader
}

static setClassLoder(newClassLoader){
    var oldClassLoader = Java.classFactory.loader
    console.log(`oldClassLoader=${oldClassLoader}`)
    Java.classFactory.loader = newClassLoader
    console.log(`Set ClassLoader to ${newClassLoader}`)
}

static updateClassLoader(className){
    var foundClassLoader = FridaAndroidUtil.findClassLoader(className)
    // console.log(`foundClassLoader=${foundClassLoader}`)
    if(foundClassLoader) {
        FridaAndroidUtil.setClassLoder(foundClassLoader)
    }
}

static CronetURLRequest() {
    var clsName_CronetURLRequest = "org.chromium.net.impl.CronetURLRequest"
    FridaAndroidUtil.updateClassLoader(clsName_CronetURLRequest)
    // FridaAndroidUtil.printClassAllMethodsFields(clsName_CronetURLRequest)

    var cls_CronetURLRequest = Java.use(clsName_CronetURLRequest)
    console.log("cls_CronetURLRequest=" + cls_CronetURLRequest)

    // FridaHookAndroidJava.CronetURLRequest_origCode(cls_CronetURLRequest)
}

```

日志:

```

Found org.chromium.net.impl.CronetURLRequest in loader dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk"],nativeLibraryDirectories=[/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]]
findClassLoader: className org.chromium.net.impl.CronetURLRequest => foundClassLoader dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk"],nativeLibraryDirectories=[/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]]
foundClassLoader=dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk"],nativeLibraryDirectories=[/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]]

```

```
oldClassLoader dalvik.system.PathClassLoader[DexPathList[[zip file "/data/app/~12Pi7u01bbj7Dqzrw11Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/base.apk"],nativeLibraryDirectories [/data/app/~12Pi7u01bbj7Dqzrw11Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/lib/arm64, /data/app/~12Pi7u01bbj7Dqzrw11Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/base.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]]
Set ClassLoader to dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk"],nativeLibraryDirectories [/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]]
```

附录日志

另外贴上，优化代码期间：

```
static updateClassFactoryLoader(className, apkName){
  console.log(`updateClassFactoryLoader: className=${className}, apkName=${apkName}`)
  const classLoaders = Java.enumerateClassLoadersSync()
  console.log("classLoaders=" + classLoaders + ", type=" + (typeof classLoaders))

  for (const loaderIdx in classLoaders) {
    var curClassLoader = classLoaders[loaderIdx]
    var loaderClsName = FridaAndroidUtil.getJavaClassName(curClassLoader)
    console.log(`[${loaderIdx}] curClassLoader=${curClassLoader}, loaderClsName=${loaderClsName}`)

    try {
      if (curClassLoader.findClass(className)){
        var loaderStr = curClassLoader.toString()
        console.log(`Found ${className} in loader ${loaderStr}`)
        if (loaderStr.includes(apkName)){
          console.log(`Update current Java.classFactory.loader to ${loaderStr}`)
          Java.classFactory.loader = curClassLoader

          break
        }
      }
    } catch (err){
      // console.log(`${err}`)
    }
  }

  var clsName_CronetUrlRequest = "org.chromium.net.impl.CronetUrlRequest"
  FridaHookAndroidJava.updateClassFactoryLoader(clsName_CronetUrlRequest, "CronetDynamite")
  FridaAndroidUtil.printClassAllMethodsFields(clsName_CronetUrlRequest)
}
```

输出的效果：

```
updateClassFactoryLoader: className org.chromium.net.impl.CronetUrlRequest, apkName CronetDynamite
classLoaders dalvik.system.PathClassLoader[DexPathList[[directory "."],nativeLibraryDirectories=[/system/lib64, /system_ext/lib64, /system/lib64, /system_ext/lib64]],java.lang.BootClassLoader bf13c25,dalvik.system.PathClassLoader[DexPathList[[zip file "/data/a
```



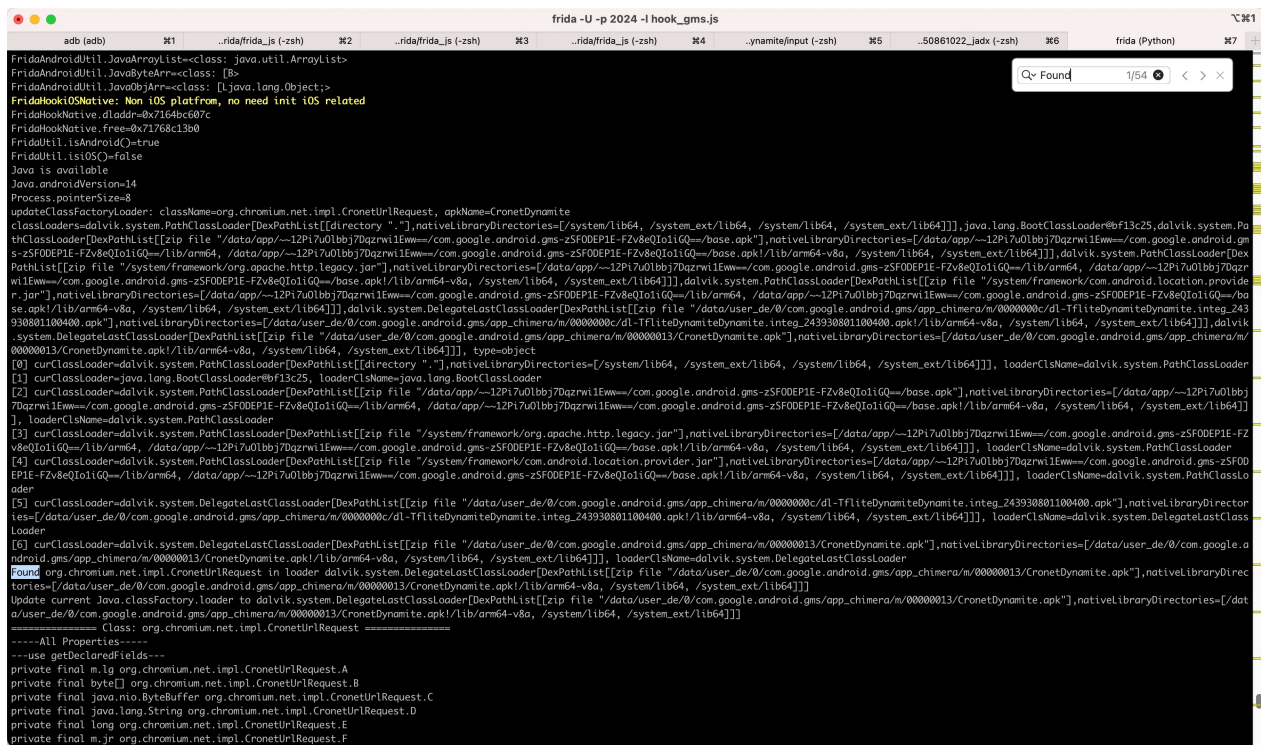
```

pp/~12Pi7u0lbbj7Dqzrw11Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/base.apk"]
, nativeLibraryDirectories=[/data/app/~12Pi7u0lbbj7Dqzrw11Eww==/com.google.android.xxx-
zSFODEP1E-FZv8eQIo1iGQ==/lib/arm64, /data/app/~12Pi7u0lbbj7Dqzrw11Eww==/com.google.and
roid.xxx-zSFODEP1E-FZv8eQIo1iGQ==/base.apk!/lib/arm64-v8a, /system/lib64, /system_ext/l
ib64]]], dalvik.system.PathClassLoader[DexPathList[[zip file "/system/framework/org.apac
he.http.legacy.jar"], nativeLibraryDirectories=[/data/app/~12Pi7u0lbbj7Dqzrw11Eww==/com
.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/lib/arm64, /data/app/~12Pi7u0lbbj7Dqzrw11
Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/base.apk!/lib/arm64-v8a, /system/
lib64, /system_ext/lib64]]], dalvik.system.PathClassLoader[DexPathList[[zip file "/syste
m/framework/com.android.location.provider.jar"], nativeLibraryDirectories=[/data/app/~1
2Pi7u0lbbj7Dqzrw11Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/lib/arm64, /dat
a/app/~12Pi7u0lbbj7Dqzrw11Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/base.a
pk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]], dalvik.system.DelegateLastClassL
oader[DexPathList[[zip file "/data/user_de/0/com.google.android.xxx/app_chimera/m/000000
00c/dl-TfliteDynamiteDynamite.integ_243930801100400.apk"], nativeLibraryDirectories=[/da
ta/user_de/0/com.google.android.xxx/app_chimera/m/00000000c/dl-TfliteDynamiteDynamite.in
teg_243930801100400.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]], dalvik.syst
em.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.xx
x/app_chimera/m/00000013/CronetDynamite.apk"], nativeLibraryDirectories=[/data/user_de/0
/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /syst
em/lib64, /system_ext/lib64]]], type object
[0] curClassLoader dalvik.system.PathClassLoader[DexPathList[[directory "."], nativeLibr
aryDirectories=[/system/lib64, /system_ext/lib64, /system/lib64, /system_ext/lib64]]],
loaderClsName dalvik.system.PathClassLoader
[1] curClassLoader java.lang.BootClassLoader bf13c25, loaderClsName java.lang.BootClass
Loader
[2] curClassLoader dalvik.system.PathClassLoader[DexPathList[[zip file "/data/app/~12P
i7u0lbbj7Dqzrw11Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/base.apk"], native
LibraryDirectories=[/data/app/~12Pi7u0lbbj7Dqzrw11Eww==/com.google.android.xxx-zSFODEP
1E-FZv8eQIo1iGQ==/lib/arm64, /data/app/~12Pi7u0lbbj7Dqzrw11Eww==/com.google.android.xx
x-zSFODEP1E-FZv8eQIo1iGQ==/base.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]],
loaderClsName dalvik.system.PathClassLoader
[3] curClassLoader dalvik.system.PathClassLoader[DexPathList[[zip file "/system/framewo
rk/org.apache.http.legacy.jar"], nativeLibraryDirectories=[/data/app/~12Pi7u0lbbj7Dqzrw
i1Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/lib/arm64, /data/app/~12Pi7u0l
bbj7Dqzrw11Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/base.apk!/lib/arm64-v8a
, /system/lib64, /system_ext/lib64]]], loaderClsName dalvik.system.PathClassLoader
[4] curClassLoader dalvik.system.PathClassLoader[DexPathList[[zip file "/system/framewo
rk/com.android.location.provider.jar"], nativeLibraryDirectories=[/data/app/~12Pi7u0lbb
j7Dqzrw11Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/lib/arm64, /data/app/~1
2Pi7u0lbbj7Dqzrw11Eww==/com.google.android.xxx-zSFODEP1E-FZv8eQIo1iGQ==/base.apk!/lib/a
rm64-v8a, /system/lib64, /system_ext/lib64]]], loaderClsName dalvik.system.PathClassLoa
der
[5] curClassLoader dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/u
ser_de/0/com.google.android.xxx/app_chimera/m/00000000c/dl-TfliteDynamiteDynamite.integ_
243930801100400.apk"], nativeLibraryDirectories=[/data/user_de/0/com.google.android.xxx/
app_chimera/m/00000000c/dl-TfliteDynamiteDynamite.integ_243930801100400.apk!/lib/arm64-v
8a, /system/lib64, /system_ext/lib64]]], loaderClsName dalvik.system.DelegateLastClassL
oader
[6] curClassLoader dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/u
ser_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDynamite.apk"], nativeLibra
ryDirectories=[/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDyna
mite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]], loaderClsName dalvik syst
em.DelegateLastClassLoader
Found org.chromium.net.impl.CronetUrlRequest in loader dalvik.system.DelegateLastClassL
oader[DexPathList[[zip file "/data/user_de/0/com.google.android.xxx/app_chimera/m/000000

```

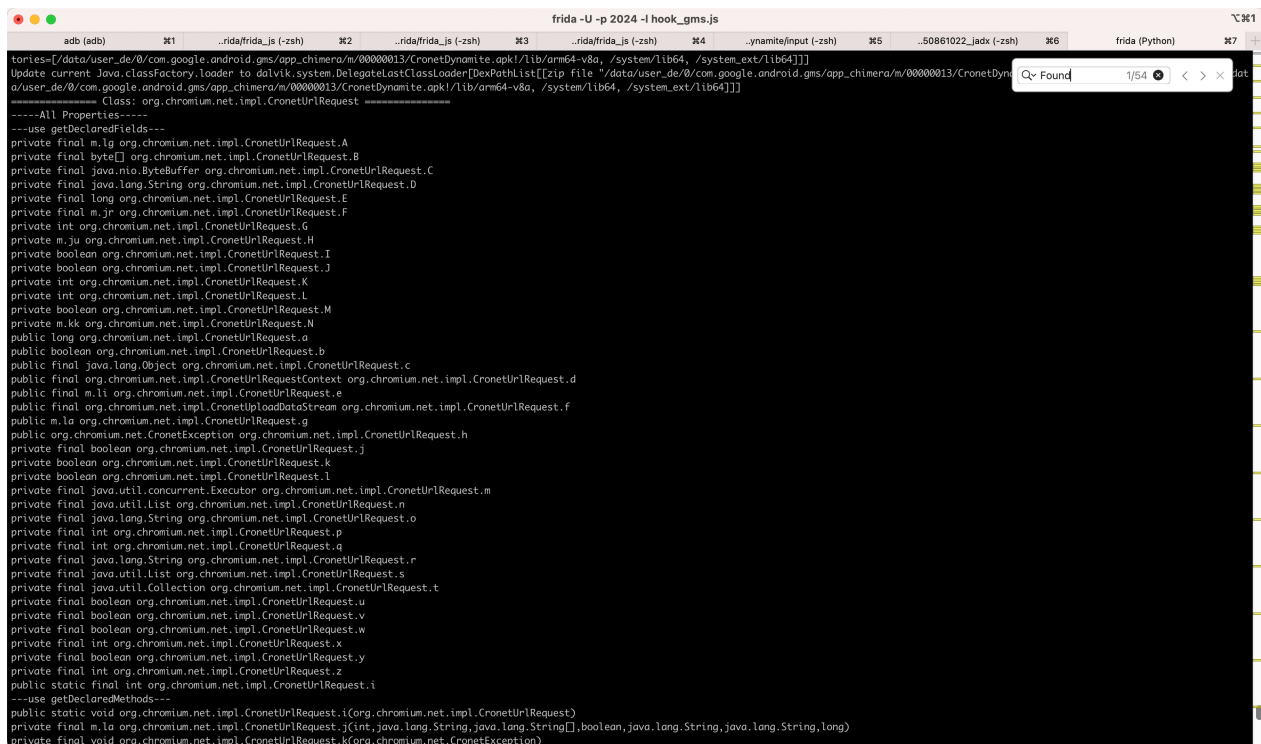


```
013/CronetDynamite.apk"], nativeLibraryDirectories ["/data/user_de/0/com.google.android.x  
xx/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext  
/lib64]]]  
Update current Java classFactory.loader to dalvik system DelegateLastClassLoader[DexPat  
hList[[zip file "/data/user_de/0/com.google.android.xxx/app_chimera/m/00000013/CronetDy  
namite.apk"], nativeLibraryDirectories ["/data/user_de/0/com.google.android.xxx/app_chime  
ra/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]]
```



```
adb (adb) X1 .rida/frida.js (-zsh) X2 .rida/frida.js (-zsh) X3 .rida/frida.js (-zsh) X4 .ynamite/input (-zsh) X5 .50861022_jadx (-zsh) X6 frida (Python) X7  
FridaAndroidUtil1.java:ArrayList<T> class: java.util.ArrayList  
FridaAndroidUtil1.java:ByteArray<T> class: [B  
FridaAndroidUtil1.java:ObjArr<T> class: [Ljava.lang.Object;  
FridaHookIOSNative: Non iOS platform, no need init iOS related  
FridaHookNative_dldaddr=0x7164bc07c  
FridaHookNative_free=0x7168c13b0  
FridaUtil.isAndroidQ=true  
FridaUtil.isIOSQ=false  
Java is available  
Java.androidVersion=14  
Process.pointerSize=8  
updateClassFactoryLoader: className=org.chromium.net.impl.CronetURLRequest, apkName=CronetDynamite  
ClassLoader=dalvik.system.PathClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/base.apk"], nativeLibraryDirectories=[/data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/lib/arm64, /data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/base.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]], dalvik.system.PathClassLoader[DexPathList[[zip file "/system/framework/org.apache.http.legacy.jar"], nativeLibraryDirectories=[/data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/lib/arm64, /data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/base.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]], dalvik.system.PathClassLoader[DexPathList[[zip file "/system/framework/com.android.location.provider.jar"], nativeLibraryDirectories=[/data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/lib/arm64, /data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/base.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]], dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.gms/app_chimera/m/0000000c/dl-TFlliteDynamiteDynamite.integ.243930801100400.apk"], nativeLibraryDirectories=[/data/user_de/0/com.google.android.gms/app_chimera/m/0000000c/dl-TFlliteDynamiteDynamite.integ.243930801100400.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]], dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.gms/app_chimera/m/00000013/CronetDynamite.apk"], nativeLibraryDirectories=[/data/user_de/0/com.google.android.gms/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]], type=object  
[0] curClassLoader=dalvik.system.PathClassLoader[DexPathList[[directory "/"], nativeLibraryDirectories=[/system/lib64, /system_ext/lib64]], loaderClsName=dalvik.system.PathClassLoader  
[1] curClassLoader=java.lang.BootClassLoader@f13c25, loaderClsName=java.lang.BootClassLoader  
[2] curClassLoader=dalvik.system.PathClassLoader[DexPathList[[zip file "/data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/base.apk"], nativeLibraryDirectories=[/data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/base.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]], loaderClsName=dalvik.system.PathClassLoader  
[3] curClassLoader=dalvik.system.PathClassLoader[DexPathList[[zip file "/system/framework/org.apache.http.legacy.jar"], nativeLibraryDirectories=[/data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/lib/arm64, /data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/base.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]], loaderClsName=dalvik.system.PathClassLoader  
[4] curClassLoader=dalvik.system.PathClassLoader[DexPathList[[zip file "/system/framework/com.android.location.provider.jar"], nativeLibraryDirectories=[/data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/lib/arm64, /data/app/~12Pi7u0lbbj7Dqzrw1Ew==/com.google.android.gms-zSF0DEPIE-FZv8eQIoiIQ==/base.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]], loaderClsName=dalvik.system.PathClassLoader  
[5] curClassLoader=dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.gms/app_chimera/m/0000000c/dl-TFlliteDynamiteDynamite.integ.243930801100400.apk"], nativeLibraryDirector  
ies=[/data/user_de/0/com.google.android.gms/app_chimera/m/0000000c/dl-TFlliteDynamiteDynamite.integ.243930801100400.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]], loaderClsName=dalvik.system.DelegateLastClass  
Loader  
[6] curClassLoader=dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.gms/app_chimera/m/00000013/CronetDynamite.apk"], nativeLibraryDirectories=[/data/user_de/0/com.google.a  
ndroid.gms/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]], loaderClsName=dalvik.system.DelegateLastClassLoader  
Found org.chromium.net.impl.CronetURLRequest in loader dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.gms/app_chimera/m/00000013/CronetDynamite.apk"], nativeLibraryDirec  
tories=[/data/user_de/0/com.google.android.gms/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]]  
Update current Java classFactory.loader to dalvik system DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.gms/app_chimera/m/00000013/CronetDynamite.apk"], nativeLibraryDirectories=[/dat  
a/user_de/0/com.google.android.gms/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]]  
Class: org.chromium.net.impl.CronetURLRequest  
-----All Properties-----  
--use getDeclaredFields--  
private final m.lg org.chromium.net.impl.CronetURLRequest.A  
private final byte[] org.chromium.net.impl.CronetURLRequest.B  
private final java.nio.ByteBuffer org.chromium.net.impl.CronetURLRequest.C  
private final java.lang.String org.chromium.net.impl.CronetURLRequest.D  
private final long org.chromium.net.impl.CronetURLRequest.E  
private final m.jr org.chromium.net.impl.CronetURLRequest.F  
private int org.chromium.net.impl.CronetURLRequest.G  
private m.jr org.chromium.net.impl.CronetURLRequest.H  
private boolean org.chromium.net.impl.CronetURLRequest.I  
private boolean org.chromium.net.impl.CronetURLRequest.J  
private int org.chromium.net.impl.CronetURLRequest.K  
private int org.chromium.net.impl.CronetURLRequest.L  
private boolean org.chromium.net.impl.CronetURLRequest.M  
private m.kk org.chromium.net.impl.CronetURLRequest.N  
public long org.chromium.net.impl.CronetURLRequest.a  
public boolean org.chromium.net.impl.CronetURLRequest.b  
public final java.lang.Object org.chromium.net.impl.CronetURLRequest.c  
public final org.chromium.net.impl.CronetURLRequestContext org.chromium.net.impl.CronetURLRequest.d  
public final m.ll org.chromium.net.impl.CronetURLRequest.e  
public final org.chromium.net.impl.CronetUploadDataStream org.chromium.net.impl.CronetURLRequest.f  
public m.la org.chromium.net.impl.CronetURLRequest.g  
public org.chromium.net.CronetException org.chromium.net.impl.CronetURLRequest.h  
private final boolean org.chromium.net.impl.CronetURLRequest.j  
private boolean org.chromium.net.impl.CronetURLRequest.k  
private boolean org.chromium.net.impl.CronetURLRequest.l  
private final java.util.concurrent.Executor org.chromium.net.impl.CronetURLRequest.m  
private final java.util.List org.chromium.net.impl.CronetURLRequest.n  
private final java.lang.String org.chromium.net.impl.CronetURLRequest.o  
private final int org.chromium.net.impl.CronetURLRequest.p  
private final int org.chromium.net.impl.CronetURLRequest.q  
private final java.lang.String org.chromium.net.impl.CronetURLRequest.r  
private final java.util.List org.chromium.net.impl.CronetURLRequest.s  
private final java.util.Collection org.chromium.net.impl.CronetURLRequest.t  
private final boolean org.chromium.net.impl.CronetURLRequest.u  
private final boolean org.chromium.net.impl.CronetURLRequest.v  
private final boolean org.chromium.net.impl.CronetURLRequest.w  
private final int org.chromium.net.impl.CronetURLRequest.x  
private final boolean org.chromium.net.impl.CronetURLRequest.y  
private final int org.chromium.net.impl.CronetURLRequest.z  
public static final int org.chromium.net.impl.CronetURLRequest.i  
--use getDeclaredMethods--  
public static void org.chromium.net.impl.CronetURLRequest.i(org.chromium.net.impl.CronetURLRequest)  
private final m.la org.chromium.net.impl.CronetURLRequest.j(int, java.lang.String, java.lang.String[], boolean, java.lang.String, java.lang.String, long)  
private final void org.chromium.net.impl.CronetURLRequest.k(org.chromium.net.CronetException)
```

后续即可正常的，找到类之后，继续打印出类的属性和函数：



```
adb (adb) X1 .rida/frida.js (-zsh) X2 .rida/frida.js (-zsh) X3 .rida/frida.js (-zsh) X4 .ynamite/input (-zsh) X5 .50861022_jadx (-zsh) X6 frida (Python) X7  
Found org.chromium.net.impl.CronetURLRequest in loader dalvik.system.DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.gms/app_chimera/m/00000013/CronetDynamite.apk"], nativeLibraryDirector  
ies=[/data/user_de/0/com.google.android.gms/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]]  
Update current Java classFactory.loader to dalvik system DelegateLastClassLoader[DexPathList[[zip file "/data/user_de/0/com.google.android.gms/app_chimera/m/00000013/CronetDynamite.apk"], nativeLibraryDirector  
ies=[/data/user_de/0/com.google.android.gms/app_chimera/m/00000013/CronetDynamite.apk!/lib/arm64-v8a, /system/lib64, /system_ext/lib64]]]  
Class: org.chromium.net.impl.CronetURLRequest  
-----All Properties-----  
--use getDeclaredFields--  
private final m.lg org.chromium.net.impl.CronetURLRequest.A  
private final byte[] org.chromium.net.impl.CronetURLRequest.B  
private final java.nio.ByteBuffer org.chromium.net.impl.CronetURLRequest.C  
private final java.lang.String org.chromium.net.impl.CronetURLRequest.D  
private final long org.chromium.net.impl.CronetURLRequest.E  
private final m.jr org.chromium.net.impl.CronetURLRequest.F  
private int org.chromium.net.impl.CronetURLRequest.G  
private m.jr org.chromium.net.impl.CronetURLRequest.H  
private boolean org.chromium.net.impl.CronetURLRequest.I  
private boolean org.chromium.net.impl.CronetURLRequest.J  
private int org.chromium.net.impl.CronetURLRequest.K  
private int org.chromium.net.impl.CronetURLRequest.L  
private boolean org.chromium.net.impl.CronetURLRequest.M  
private m.kk org.chromium.net.impl.CronetURLRequest.N  
public long org.chromium.net.impl.CronetURLRequest.a  
public boolean org.chromium.net.impl.CronetURLRequest.b  
public final java.lang.Object org.chromium.net.impl.CronetURLRequest.c  
public final org.chromium.net.impl.CronetURLRequestContext org.chromium.net.impl.CronetURLRequest.d  
public final m.ll org.chromium.net.impl.CronetURLRequest.e  
public final org.chromium.net.impl.CronetUploadDataStream org.chromium.net.impl.CronetURLRequest.f  
public m.la org.chromium.net.impl.CronetURLRequest.g  
public org.chromium.net.CronetException org.chromium.net.impl.CronetURLRequest.h  
private final boolean org.chromium.net.impl.CronetURLRequest.j  
private boolean org.chromium.net.impl.CronetURLRequest.k  
private boolean org.chromium.net.impl.CronetURLRequest.l  
private final java.util.concurrent.Executor org.chromium.net.impl.CronetURLRequest.m  
private final java.util.List org.chromium.net.impl.CronetURLRequest.n  
private final java.lang.String org.chromium.net.impl.CronetURLRequest.o  
private final int org.chromium.net.impl.CronetURLRequest.p  
private final int org.chromium.net.impl.CronetURLRequest.q  
private final java.lang.String org.chromium.net.impl.CronetURLRequest.r  
private final java.util.List org.chromium.net.impl.CronetURLRequest.s  
private final java.util.Collection org.chromium.net.impl.CronetURLRequest.t  
private final boolean org.chromium.net.impl.CronetURLRequest.u  
private final boolean org.chromium.net.impl.CronetURLRequest.v  
private final boolean org.chromium.net.impl.CronetURLRequest.w  
private final int org.chromium.net.impl.CronetURLRequest.x  
private final boolean org.chromium.net.impl.CronetURLRequest.y  
private final int org.chromium.net.impl.CronetURLRequest.z  
public static final int org.chromium.net.impl.CronetURLRequest.i  
--use getDeclaredMethods--  
public static void org.chromium.net.impl.CronetURLRequest.i(org.chromium.net.impl.CronetURLRequest)  
private final m.la org.chromium.net.impl.CronetURLRequest.j(int, java.lang.String, java.lang.String[], boolean, java.lang.String, java.lang.String, long)  
private final void org.chromium.net.impl.CronetURLRequest.k(org.chromium.net.CronetException)
```

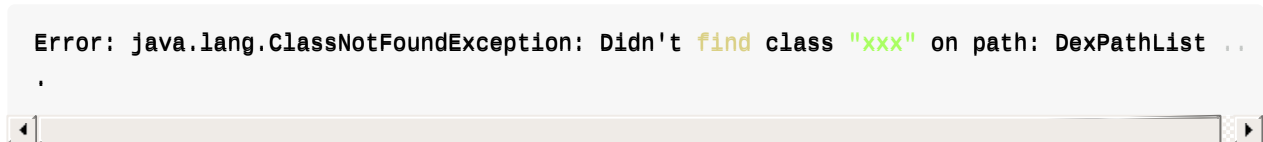
进一步优化：工具类函数

最新整理出工具类函数：

- `FridaAndroidUtil.updateClassLoader`
 - 已发布到：
 - <https://github.com/crifan/JsFridaUtil/blob/main/frida/FridaAndroidUtil.js>

心得

如果去hook其他类，也出现类似错误：



那么也可以尽快去试试：

用上面的

```
FridaAndroidUtil.updateClassLoader("xxx")
```

看看是否有效。

crifan.org，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook最后更新：
2025-12-13 21:54:08

Error cannot call instance method without an instance

现象

对于Java的类 `java.util.HashMap` 的 `put` 函数：

```
public Object HashMap.put(Object, Object)
```

去hook：

```
var func_HashMap_put = cls_HashMap.put('java.lang.Object', 'java.lang.Object')
```

但报错：

```
Error: put: cannot call instance method without an instance
```

原因

此处语法错误 == 笔误，写成：重载overload函数的写法（加上函数参数类型）

```
cls_HashMap.put('java.lang.Object', 'java.lang.Object')
```

但该写法其实是函数调用的写法

而如果去调用函数的话，的确需要：Instance实例

但是此处是hook，不是函数调用

此处只是需要去hook该函数，所以只需要

解决办法

只写出函数名（而无需额外的，类似overload的参数类型）

具体写法

```
var func_HashMap_put = cls_HashMap.put
```

crifan.org，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新：
2025-05-25 10:07:15

附录

下面列出相关参考资料。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2022-03-17 20:39:28

参考资料

- 【已解决】安卓保活逆向360Wallpaper：字符串解密函数e.i相关代码
- 【已解决】安卓保活逆向360Wallpaper：jadx反编译代码中e.i的通用字符串加密解密函数
- 【已解决】安卓保活逆向360Wallpaper：去使用m1.e.i去解密已混淆加密字符串得到原始字符串
- 【已解决】安卓保活逆向360Wallpaper：用JEB反编译
- 【已解决】安卓保活逆向360Wallpaper：用JEB反编译得到java代码
- 【记录】安卓apk反编译java代码效果对比：JEB对比jadx
- 【心得】Frida调试安卓：java的对象数组类型如何打印数组内的对象的值的详情
- 【整理】Frida调试安卓Java类心得：hook类的初始化构造函数中返回值为空undefined
- 【整理】Frida调试安卓心得：hook安卓java类的初始化构造函数的hook写法
- 【无需解决】Frida去hook调试java类的函数中：始终无法获取属性值始终都是undefined
- 【已解决】Frida去hook调试java的Enum枚举的类
- 【记录】安卓逆向GMS：Frida中打印ByteBuffer的内存数据
- 【已解决】Frida的js中打印ByteBuffer的数据和打印字符串值
- 【已解决】Frida中的Java和js变量类型的映射关系
- 【已解决】Frida的js中如何打印Java的Map的ArrayList
- 【记录】安卓逆向GMS：类org.chromium.net.impl.CronetURLRequest
- 【记录】安卓逆向GMS：根据源码去hook类org.chromium.net.impl.CronetURLRequest
- 【记录】安卓逆向GMS：Frida去hook但找不到类org.chromium.net.impl.CronetURLRequest
- 【记录】安卓逆向GMS：去hook混淆后的类org.chromium.net.impl.CronetURLRequest
- 【记录】安卓逆向GSF：如何找到gsf的进程用于Frida去hook调试
-
- [JsFridaUtil/frida/FridaHookAndroidJava.js at main · crifan/JsFridaUtil](#)
- [JsFridaUtil/frida/FridaAndroidUtil.js at main · crifan/JsFridaUtil](#)
-

crifan.org，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新：
2025-12-13 22:36:18