目录

Frida高级调试：Stalker

- 最新版本：`1.0.0`
- 更新时间：`20250603`

简介

介绍Frida高级调试中的Stalker的详细用法和技巧，包括各种举例和心得。先是概述Frida的Stalker，然后是介绍其基本用法和API接口；然后给出多个Stalker的详细实例举例，包括完整的代码和输出日志以及相关说明；然后整理出Stalker.follow的相关心得，包括如何利用Staker，events的属性含义，transform的基本逻辑，context有哪些属性，Stalker.follow()的内部实现机制原理，为何函数地址和指令地址不匹配等等；然后介绍用Stalker.exclude提高效率和减少干扰；然后贴出Stalker的工具类函数，尤其是stalkerHookUnnameNative等。最后给出附录和参考资料。

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_advanced_debug_stalker: Frida高级调试：Stalker

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高级调试：Stalker book.crifan.org
- Frida高级调试：Stalker crifan.github.io

离线下载阅读

- Frida高级调试：Stalker PDF
- Frida高级调试：Stalker ePub
- Frida高级调试：Stalker Mobi

版权和用途说明

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

Stalker概览

- `Frida` 的 `Stalker`
  - 名称
    - 别称：`frida-stalker`
    - 笔者译为：`Frida代码跟踪器`
  - 作用：汇编指令级别的hook
  - 用途：追踪真实代码指令的执行过程
  - 官网文档
    - Stalker介绍
    - Stalker的API接口

Stalker单词的含义

- Stalk的英文原意：
  - `n.` （植物的）茎，秆；（动物的）肉柄，肉茎；柄；（车辆）（转向柱上控制指示仪、灯光等的）手柄；悄悄追踪；高视阔步
  - `v.` 偷偷接近，潜近；跟踪，盯梢；怒冲冲地走，趾高气扬地走；令人厌恶地穿过，威胁地通过；<文>（不好的气氛）笼罩，蔓延
- Frida中的Stalker的含义
  - Stalk英文原意：跟踪
    - Stalker英文原意：跟踪者
      - 有人（把Frida的Stalker）翻译成：`潜行者`

何时需要用到Frida的Stalker?

如下场景：

- 你想要搞懂该函数内部的执行的逻辑，想要查看哪个函数，甚至是哪个代码块codeblock，被执行了
- 还比如你想要搞懂，当传入不同参数时，函数内部代码执行的流程路径，是否有何不同

就可以去用：

- `Stalker.follow()`

Stalker介绍

此处详细介绍Frida的Stalker相关内容：

Stalker的用法

- Stalker的用法的概述
  - 核心逻辑
    - 在frida命令上，和普通frida一样，都是调用js

```
frida -U -n akd -l ./fridaStalker_akdSymbol2575.js

frida -U -p 1234 -l yourFridaScript.js

frida -U -f com.apple.Preferences -l yourFridaScript.js
```

    - js内部逻辑
      - 最初要初始化：计算出当前要hook的函数所属的模块和地址
      - 在普通的 `Interceptor.attach` 的 `onEnter` 中，加上 `Stalker.follow`
        - 在 `transform` 中，计算是否是原始函数的代码
          - 如果是，再去：实现特定的调试的目的
            - 打印真实执行的指令的信息 `instruction.toString()`
            - 打印当时的变量的值 `context`
            - 等等

Stalker的API接口

- Stalker的API接口
  - 接口形式
    - 普通用户常直接调用：`JS的API`
      - Stalker - JavaScript API | Frida
    - 内部接口：Gum接口
      - TypeScript type definitions
        - DefinitelyTyped/index.d.ts
  - 接口内容
    - 最核心接口
      - `Stalker.follow([threadId, options])`
    - 常用接口
      - `Stalker.exclude(range)`
      - `Stalker.parse(events[, options])`
      - `Stalker.unfollow([threadId])`
    - 其他
      - `Stalker.flush()`
      - `Stalker.garbageCollect()`
      - `Stalker.invalidate(address)`
      - `Stalker.invalidate(threadId, address)`
      - `Stalker.addCallProbe(address, callback[, data])`
      - `Stalker.removeCallProbe`
      - `Stalker.trustThreshold`
      - `Stalker.queueCapacity`
      - `Stalker.queueDrainInterval`

Stalker举例

下面去给出众多Frida的Stalker的真实使用案例以及调试结果以及相关说明。

___lldb_unnamed_symbol2575$$akd

用 Frida 的 Stalker 调试 arm64 的 akd 函数 ___lldb_unnamed_symbol2575$$akd：

Frida命令

```
frida -U -n akd -l ./fridaStalker_akdSymbol2575.js
```

js文件 fridaStalker_akdSymbol2575.js 的内容

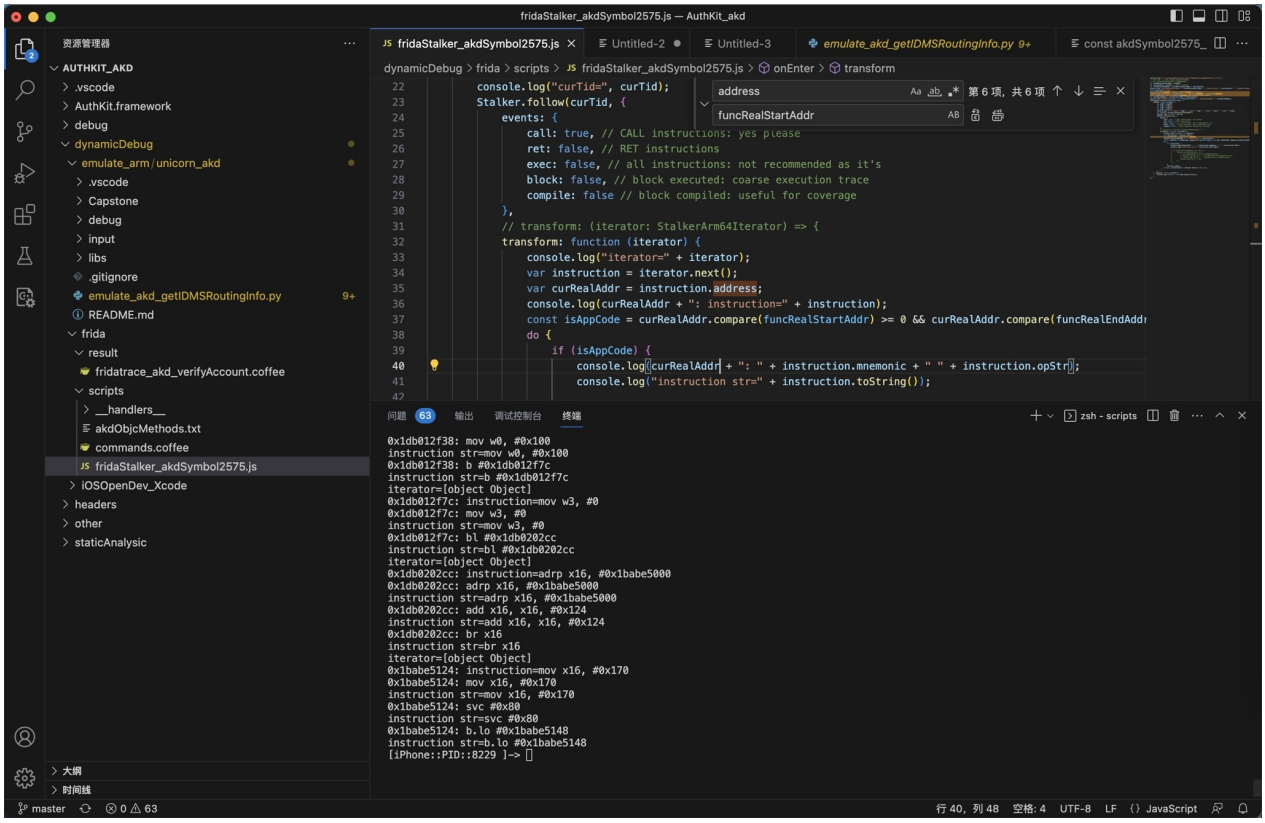
```
console.log("===== Frida Stalker hook for arm64 ___lldb_unnamed_symbol2575$$akd =====");

// arm64 akd: ___lldb_unnamed_symbol2575$$akd
// var funcRelativeStartAddr = 0x1000a0460;
var funcRelativeStartAddr = 0xa0460;
var functionSize = 0x24C8; // 9416 == 0x24C8
var funcRelativeEndAddr = funcRelativeStartAddr + functionSize;
console.log("funcRelativeStartAddr=" + funcRelativeStartAddr + ", functionSize=" + func
tionSize + ", funcRelativeEndAddr=" + funcRelativeEndAddr);
const moduleName = "akd";
const moduleBaseAddress = Module.findBaseAddress(moduleName);
console.log("moduleName=" + moduleName + ", moduleBaseAddress=" + moduleBaseAddress);
// console.log("moduleName=%s, moduleBaseAddress=%p", moduleName, moduleBaseAddress);
const funcRealStartAddr = moduleBaseAddress.add(funcRelativeStartAddr);
// var funcRealEndAddr = funcRealStartAddr + functionSize;
const funcRealEndAddr = funcRealStartAddr.add(functionSize);
console.log("funcRealStartAddr=" + funcRealStartAddr + ", funcRealEndAddr=" + funcRealE
ndAddr);
var curTid = null;
Interceptor.attach(funcRealStartAddr, {
    onEnter: function(args) {
        var arg0 = args[0]
        var arg1 = args[1]
        var arg2 = args[2]
        var arg3 = args[3]
        console.log("----- arg0=" + arg0 + ", arg1=" + arg1 + ", arg2=" + arg2 + ", arg
3=" + arg3);
        var curTid = Process.getCurrentThreadId();
        console.log("curTid=", curTid);
        Stalker.follow(curTid, {
            events: {
                call: false, // CALL instructions: yes please
                ret: true, // RET instructions
                exec: false, // all instructions: not recommended as it's
                block: false, // block executed: coarse execution trace
                compile: false // block compiled: useful for coverage
            },
            // onReceive: Called with `events` containing a binary blob comprised of on
e or more GumEvent structs. See `gumevent.h` for details about the format. Use `Stalker
.parse()` to examine the data.
```

```
            onReceive(events) {
                var parsedEvents = Stalker.parse(events)
                // var parsedEventsStr = JSON.stringify(parsedEventsStr)
                // console.log(">>> into onReceive: parsedEvents=" + parsedEvents + ",
parsedEventsStr=" + parsedEventsStr);
                console.log(">>> into onReceive: parsedEvents=" + parsedEvents);
            },

            // transform: (iterator: StalkerArm64Iterator) => {
            transform: function (iterator) {
                // console.log("iterator=" + iterator);
                var instruction = iterator.next();
                const startAddress = instruction.address;
                // console.log("+++ into iterator: startAddress=" + startAddress);
                // const isAppCode = startAddress.compare(funcRealStartAddr) >= 0 && st
artAddress.compare(funcRealEndAddr) === -1;
                // const isAppCode = (startAddress.compare(funcRealStartAddr) >= 0) &&
(startAddress.compare(funcRealEndAddr) < 0);
                const gt_realStartAddr = startAddress.compare(funcRealStartAddr) >= 0
                const lt_realEndAddr = startAddress.compare(funcRealEndAddr) < 0
                var isAppCode = gt_realStartAddr && lt_realEndAddr
                console.log("+++ into iterator: startAddress=" + startAddress + ", isAp
pCode=" + isAppCode);

                // // for debug
                // isAppCode = true

                // console.log("isAppCode=" + isAppCode + ", gt_realStartAddr=" + gt_re
alStartAddr + ", lt_realEndAddr=" + lt_realEndAddr);
                do {
                    if (isAppCode) {
                        // is origal function code = which we focus on

                        // console.log("instruction: address=" + instruction.address
                        //     + ",next=" + instruction.next()
                        //     + ",size=" + instruction.size
                        //     + ",mnemonic=" + instruction.mnemonic
                        //     + ",opStr=" + instruction.opStr
                        //     + ",operands=" + JSON.stringify(instruction.operands)
                        //     + ",regsAccessed=" + JSON.stringify(instruction.regsAcce
ssed)
                        //     + ",regsRead=" + JSON.stringify(instruction.regsRead)
                        //     + ",regsWritten=" + JSON.stringify(instruction.regsWritt
en)
                        //     + ",groups=" + JSON.stringify(instruction.groups)
                        //     + ",toString()=" + instruction.toString()
                        //     + ",toJSON()=" + instruction.toJSON()
                        // );

                        var curRealAddr = instruction.address;
                        // const isAppCode = curRealAddr.compare(funcRealStartAddr) >=
0 && curRealAddr.compare(funcRealEndAddr) === -1;
                        // console.log(curRealAddr + ": isAppCode=" + isAppCode);
                        var curOffsetHexPtr = curRealAddr.sub(funcRealStartAddr)
                        var curOffsetInt = curOffsetHexPtr.toInt32()
```

```
                        // var instructionStr = instruction.mnemonic + " " + instructio
n.opStr
                        var instructionStr = instruction.toString()
                        // console.log("\t" + curRealAddr + ": " + instructionStr);
                        // console.log("\t" + curRealAddr + " <+" + curOffsetHexPtr + "
>: " + instructionStr);
                        console.log("\t" + curRealAddr + " <+" + curOffsetInt + ">: " +
 instructionStr);
                        if (curOffsetInt == 8516) {
                            console.log("curOffsetInt=" + curOffsetInt);
                            // iterator.putCallout(needDebug);
                            iterator.putCallout((context) => {
                                var contextStr = JSON.stringify(context)
                                console.log("contextStr=" + contextStr);
                                var x9Value1 = context.x9
                                var x9Value2 = context["x9"]
                                console.log("x9Value1=" + x9Value1 + ", x9Value2=" + x9
Value2)
                            });
                        }
                    }
                    iterator.keep();
                } while ((instruction = iterator.next()) !== null);
            }
        });

        // function needDebug(context) {
        //     console.log("into needDebug");
        //     // console.log("into needDebug: context=" + context);
        //     // var contextStr = JSON.stringify(context, null, 2)
        //     // console.log("context=" + contextStr);
        //     // var x9Value1 = context.x9
        //     // var x9Value2 = context["x9"]
        //     // console.log("x9Value1=" + x9Value1 + ", x9Value2=" + x9Value2)
        // }
    },
    onLeave: function(retval) {
        console.log("retval=", new ObjC.Object(retval))
        if (curTid != null) {
            Stalker.unfollow(curTid);
            console.log("Stalker.unfollow curTid=", curTid)
        }
    }
});
```

调试期间

期间某次调试，以及优化代码逻辑时的调试效果：




```
...
instruction str ldp x22, x21, [sp, #0xb0]
0x18134bb70: ldp x24, x23, [sp, #0xa0]
instruction str ldp x24, x23, [sp, #0xa0]
0x18134bb70: ldp x26, x25, [sp, #0x90]
instruction str ldp x26, x25, [sp, #0x90]
0x18134bb70: ldp x28, x27, [sp, #0x80]
instruction str ldp x28, x27, [sp, #0x80]
0x18134bb70: add sp, sp, #0xe0
instruction str add sp, sp, #0xe0
0x18134bb70: ret
instruction str ret
iterator [object Object]
0x1db012f38: instruction ldr x1, [x22]
0x1db012f38: ldr x1, [x22]
instruction str ldr x1, [x22]
0x1db012f38: ldr w2, [x21]
instruction str ldr w2, [x21]
0x1db012f38: mov w0, #0x100
instruction str mov w0, #0x100
0x1db012f38: b #0x1db012f7c
instruction str b #0x1db012f7c
iterator [object Object]
0x1db012f7c: instruction mov w3, #0
0x1db012f7c: mov w3, #0
instruction str mov w3, #0
0x1db012f7c: bl #0x1db0202cc
instruction str bl #0x1db0202cc
iterator [object Object]
0x1db0202cc: instruction adrp x16, #0x1babe5000
0x1db0202cc: adrp x16, #0x1babe5000
```

```
instruction str adrp x16, #0x1babe5000
0x1db0202cc: add x16, x16, #0x124
instruction str add x16, x16, #0x124
0x1db0202cc: br x16
instruction str br x16
iterator=[object Object]
0x1babe5124: instruction mov x16, #0x170
0x1babe5124: mov x16, #0x170
instruction str mov x16, #0x170
0x1babe5124: svc #0x80
instruction str svc #0x80
0x1babe5124: b.lo #0x1babe5148
instruction str b.lo #0x1babe5148
[iPhone::PID::8229 ]->
```

最终输出

最开始输出环境初始化信息：

```
====== Frida Stalker hook for arm64 ___lldb_unnamed_symbol2575$$akd ======
funcRelativeStartAddr=656480, functionSize=9416, funcRelativeEndAddr=665896
moduleName akd, moduleBaseAddress=0x1040e4000
funcRealStartAddr=0x104184460, funcRealEndAddr=0x104186928
[iPhone::akd ]-> ----- arg0=0xfffffffffffffffe, arg1=0x16bf46838, arg2=0x16bf46838, arg3
=0xfffffffffffffffe
curTid=9739
```

（如果开启调试：

```
// for debug
isAppCode = true
```

则会输出）

（最初是）一些非原始函数代码：

```
+++ into iterator  startAddress=0x106d0bcd8, isAppCode=false
    0x106d0bcd8 <+45643896>: b #0x106d0bce8
+++ into iterator  startAddress=0x106d0bce8, isAppCode=false
    0x106d0bce8 <+45643912>: str wzr, [x19, #0x90]
    0x106d0bcec <+45643916>: ldr x0, [x19, #0xa0]
    0x106d0bcf0 <+45643920>: str xzr, [x19, #0xa0]
    0x106d0bcf4 <+45643924>: cbz x0, #0x106d0bcfc
+++ into iterator  startAddress=0x106d0bcf8, isAppCode=false
    0x106d0bcf8 <+45643928>: bl #0x106c0fef8
+++ into iterator  startAddress=0x106d0bcfc, isAppCode=false
    0x106d0bcfc <+45643932>: ldr x0, [x19, #0x98]
    0x106d0bd00 <+45643936>: str xzr, [x19, #0x98]
    0x106d0bd04 <+45643940>: cbz x0, #0x106d0bd14
+++ into iterator  startAddress=0x106d0bd08, isAppCode=false
    0x106d0bd08 <+45643944>: ldp x29, x30, [sp, #0x10]
```

```
    0x106d0bd0c <+45643948>: ldp x20, x19, [sp], #0x20
    0x106d0bd10 <+45643952>: b #0x106c0fef8
+++ into iterator  startAddress 0x106c0fef8, isAppCode=false
    0x106c0fef8 <+44612248>: stp x24, x23, [sp, #-0x40]!
    0x106c0fefc <+44612252>: stp x22, x21, [sp, #0x10]
    0x106c0ff00 <+44612256>: stp x20, x19, [sp, #0x20]
    0x106c0ff04 <+44612260>: stp x29, x30, [sp, #0x30]
    0x106c0ff08 <+44612264>: add x29, sp, #0x30
    0x106c0ff0c <+44612268>: mov x19, x0
    0x106c0ff10 <+44612272>: add x23, x0, #8
    0x106c0ff14 <+44612276>: add x20, x0, #0x10
    0x106c0ff18 <+44612280>: adrp x22, #0x107d28000
    0x106c0ff1c <+44612284>: adrp x21, #0x107d28000
    0x106c0ff20 <+44612288>: add x21, x21, #0x790
    0x106c0ff24 <+44612292>: ldar w24, [x23]
    0x106c0ff28 <+44612296>: cmp w24, #2
    0x106c0ff2c <+44612300>: b.lt #0x106c0ff50
+++ into iterator  startAddress 0x106c0ff30, isAppCode=false
    0x106c0ff30 <+44612304>: bl #0x106c12110
+++ into iterator  startAddress 0x106c0ff34, isAppCode=false
    0x106c0ff34 <+44612308>: sub w8, w24, #1
    0x106c0ff38 <+44612312>: ldaxr w9, [x23]
    0x106c0ff3c <+44612316>: cmp w9, w24
    0x106c0ff40 <+44612320>: b.ne #0x106c0ff7c
    0x106c0ff44 <+44612324>: stlxr w9, w8, [x23]
    0x106c0ff48 <+44612328>: cbnz w9, #0x106c0ff38
+++ into iterator  startAddress 0x106c0ff4c, isAppCode=false
    0x106c0ff4c <+44612332>: b #0x106c0ff84
+++ into iterator  startAddress 0x106c0ff84, isAppCode=false
    0x106c0ff84 <+44612388>: cmp w24, #2
    0x106c0ff88 <+44612392>: b.ne #0x106c10028
+++ into iterator  startAddress 0x106c0ff8c, isAppCode=false
    0x106c0ff8c <+44612396>: tbz w0, #0, #0x106c10028
+++ into iterator  startAddress 0x106c10028, isAppCode=false
    0x106c10028 <+44612552>: ldp x29, x30, [sp, #0x30]
    0x106c1002c <+44612556>: ldp x20, x19, [sp, #0x20]
    0x106c10030 <+44612560>: ldp x22, x21, [sp, #0x10]
    0x106c10034 <+44612564>: ldp x24, x23, [sp], #0x40
    0x106c10038 <+44612568>: ret
+++ into iterator  startAddress 0x106d0a104, isAppCode=false
    0x106d0a104 <+45636772>: ldur x8, [x29, #-0x48]
    0x106d0a108 <+45636776>: adrp x9, #0x107c28000
    0x106d0a10c <+45636780>: ldr x9, [x9, #0x28]
    0x106d0a110 <+45636784>: ldr x9, [x9]
    0x106d0a114 <+45636788>: cmp x9, x8
    0x106d0a118 <+45636792>: b.ne #0x106d0a138
+++ into iterator  startAddress 0x106d0a11c, isAppCode=false
    0x106d0a11c <+45636796>: ldp x29, x30, [sp, #0x100]
    0x106d0a120 <+45636800>: ldp x20, x19, [sp, #0xf0]
    0x106d0a124 <+45636804>: ldp x22, x21, [sp, #0xe0]
    0x106d0a128 <+45636808>: ldp x24, x23, [sp, #0xd0]
    0x106d0a12c <+45636812>: ldp x28, x27, [sp, #0xc0]
    0x106d0a130 <+45636816>: add sp, sp, #0x110
    0x106d0a134 <+45636820>: ret
+++ into iterator  startAddress 0x106cb825c, isAppCode=false
    0x106cb825c <+45301244>: add w22, w22, #1
```

```
    0x106cb8260 <+45301248>: b #0x106cb820c
+++ into iterator  startAddress 0x106cb820c, isAppCode=false
    0x106cb820c <+45301164>: ldr w8, [x25, #8]
    0x106cb8210 <+45301168>: cmp w22, w8
    0x106cb8214 <+45301172>: b.eq #0x106cb8264
+++ into iterator  startAddress 0x106cb8264, isAppCode=false
    0x106cb8264 <+45301252>: ldr w25, [x26, #0x10]
    0x106cb8268 <+45301256>: tbnz w19, #0, #0x106cb8274
+++ into iterator  startAddress 0x106cb8274, isAppCode=false
    0x106cb8274 <+45301268>: ldr x21, [sp, #8]
    0x106cb8278 <+45301272>: mov x0, x25
    0x106cb827c <+45301276>: bl #0x106cc899c
+++ into iterator  startAddress 0x106cb8280, isAppCode=false
    0x106cb8280 <+45301280>: adrp x8, #0x107d2a000
    0x106cb8284 <+45301284>: ldr x0, [x8, #0x780]
    0x106cb8288 <+45301288>: bl #0x106cb8b94
+++ into iterator  startAddress 0x106cb828c, isAppCode=false
    0x106cb828c <+45301292>: cbz w19, #0x106cb829c
+++ into iterator  startAddress 0x106cb8290, isAppCode=false
    0x106cb8290 <+45301296>: ldr x8, [x20, #0x70]
    0x106cb8294 <+45301300>: ldr x9, [sp, #0x10]
    0x106cb8298 <+45301304>: str x8, [x9]
    0x106cb829c <+45301308>: ldr x8, [x27]
    0x106cb82a0 <+45301312>: cbz x8, #0x106cb82dc
+++ into iterator  startAddress 0x106cb82dc, isAppCode=false
    0x106cb82dc <+45301372>: add x27, x20, #0x68
    0x106cb82e0 <+45301376>: ldr x8, [x27]
    0x106cb82e4 <+45301380>: str x8, [x21]
    0x106cb82e8 <+45301384>: tbnz w19, #0, #0x106cb82fc
+++ into iterator  startAddress 0x106cb82fc, isAppCode=false
    0x106cb82fc <+45301404>: ldp x29, x30, [sp, #0x90]
    0x106cb8300 <+45301408>: ldp x20, x19, [sp, #0x80]
    0x106cb8304 <+45301412>: ldp x22, x21, [sp, #0x70]
    0x106cb8308 <+45301416>: ldp x24, x23, [sp, #0x60]
    0x106cb830c <+45301420>: ldp x26, x25, [sp, #0x50]
    0x106cb8310 <+45301424>: ldp x28, x27, [sp, #0x40]
    0x106cb8314 <+45301428>: add sp, sp, #0xa0
    0x106cb8318 <+45301432>: ret
+++ into iterator  startAddress 0x1044fc0b0, isAppCode=false
    0x1044fc0b0 <+3636304>: add sp, sp, #0x10
    0x1044fc0b4 <+3636308>: ldp x1, x0, [sp], #0x10
    0x1044fc0b8 <+3636312>: msr nzcv, x1
    0x1044fc0bc <+3636316>: ldp x1, x2, [sp], #0x10
    0x1044fc0c0 <+3636320>: ldp x3, x4, [sp], #0x10
    0x1044fc0c4 <+3636324>: ldp x5, x6, [sp], #0x10
    0x1044fc0c8 <+3636328>: ldp x7, x8, [sp], #0x10
    0x1044fc0cc <+3636332>: ldp x9, x10, [sp], #0x10
    0x1044fc0d0 <+3636336>: ldp x11, x12, [sp], #0x10
    0x1044fc0d4 <+3636340>: ldp x13, x14, [sp], #0x10
    0x1044fc0d8 <+3636344>: ldp x15, x16, [sp], #0x10
    0x1044fc0dc <+3636348>: ldp x17, x18, [sp], #0x10
    0x1044fc0e0 <+3636352>: ldp x19, x20, [sp], #0x10
    0x1044fc0e4 <+3636356>: ldp x21, x22, [sp], #0x10
    0x1044fc0e8 <+3636360>: ldp x23, x24, [sp], #0x10
    0x1044fc0ec <+3636364>: ldp x25, x26, [sp], #0x10
    0x1044fc0f0 <+3636368>: ldp x27, x28, [sp], #0x10
```

```
    0x1044fc0f4 <+3636372>: ldp x29, x30, [sp], #0x10
    0x1044fc0f8 <+3636376>: ldp q0, q1, [sp], #0x20
    0x1044fc0fc <+3636380>: ldp q2, q3, [sp], #0x20
    0x1044fc100 <+3636384>: ldp q4, q5, [sp], #0x20
    0x1044fc104 <+3636388>: ldp q6, q7, [sp], #0x20
    0x1044fc108 <+3636392>: ldp q8, q9, [sp], #0x20
    0x1044fc10c <+3636396>: ldp q10, q11, [sp], #0x20
    0x1044fc110 <+3636400>: ldp q12, q13, [sp], #0x20
    0x1044fc114 <+3636404>: ldp q14, q15, [sp], #0x20
    0x1044fc118 <+3636408>: ldp q16, q17, [sp], #0x20
    0x1044fc11c <+3636412>: ldp q18, q19, [sp], #0x20
    0x1044fc120 <+3636416>: ldp q20, q21, [sp], #0x20
    0x1044fc124 <+3636420>: ldp q22, q23, [sp], #0x20
    0x1044fc128 <+3636424>: ldp q24, q25, [sp], #0x20
    0x1044fc12c <+3636428>: ldp q26, q27, [sp], #0x20
    0x1044fc130 <+3636432>: ldp q28, q29, [sp], #0x20
    0x1044fc134 <+3636436>: ldp q30, q31, [sp], #0x20
    0x1044fc138 <+3636440>: ldp x16, x17, [sp], #0x10
    0x1044fc13c <+3636444>: ret x16
+++ into iterator  startAddress 0x108040030, isAppCode=false
    0x108040030 <+65780688>: sub sp, sp, #0xf0
    0x108040034 <+65780692>: stp x28, x27, [sp, #0x90]
    0x108040038 <+65780696>: stp x26, x25, [sp, #0xa0]
    0x10804003c <+65780700>: stp x24, x23, [sp, #0xb0]
    0x108040040 <+65780704>: ldr x16, #0x108040048
    0x108040044 <+65780708>: br x16
```

之后才是：原始函数代码

```
+++ into iterator  startAddress 0x104184470, isAppCode=true
    0x104184470 <+16>: stp x22, x21, [sp, #0xc0]
    0x104184474 <+20>: stp x20, x19, [sp, #0xd0]
    0x104184478 <+24>: stp x29, x30, [sp, #0xe0]
    0x10418447c <+28>: add x29, sp, #0xe0
    0x104184480 <+32>: nop
    0x104184484 <+36>: ldr x8, #0x1041d87d8
    0x104184488 <+40>: ldr x8, [x8]
    0x10418448c <+44>: stur x8, [x29, #-0x58]
    0x104184490 <+48>: mov w26, #0x5a87
    0x104184494 <+52>: movk w26, #0x6c24, lsl #16
    0x104184498 <+56>: add x8, x0, #6
    0x10418449c <+60>: cmp x8, #5
    0x1041844a0 <+64>: ccmn x0, #8, #4, hs
    0x1041844a4 <+68>: mov w8, #1
    0x1041844a8 <+72>: csel w8, wzr, w8, eq
    0x1041844ac <+76>: mov w11, #0xa5a2
    0x1041844b0 <+80>: movk w11, #0x93db, lsl #16
    0x1041844b4 <+84>: cmp x1, #0
    0x1041844b8 <+88>: cset w9, eq
    0x1041844bc <+92>: orr w8, w9, w8
    0x1041844c0 <+96>: add w9, w26, w8
    0x1041844c4 <+100>: add w9, w9, w11
    0x1041844c8 <+104>: sub w9, w9, #0x21
    0x1041844cc <+108>: adr x25, #0x1041a6cb0
    0x1041844d0 <+112>: nop
```

```
    0x1041844d4 <+116>: ldrsw x9, [x25, w9, sxtw #2]
    0x1041844d8 <+120>: nop
    0x1041844dc <+124>: ldr x10, #0x1041dd158
    0x1041844e0 <+128>: add x9, x9, x10
    0x1041844e4 <+132>: mov w22, #-0xafc9
    0x1041844e8 <+136>: br x9
+++ into iterator  startAddress 0x1041844ec, isAppCode=true
    0x1041844ec <+140>: mov x23, x1
    0x1041844f0 <+144>: mov x28, x0
    0x1041844f4 <+148>: eor w8, w8, #1
    0x1041844f8 <+152>: sub w9, w11, #0x29
    0x1041844fc <+156>: madd w22, w8, w9, w26
    0x104184500 <+160>: mov x20, #0xa930
    0x104184504 <+164>: movk x20, #0xea59, lsl #16
    0x104184508 <+168>: movk x20, #0x9bdd, lsl #32
    0x10418450c <+172>: movk x20, #0xd570, lsl #48
    0x104184510 <+176>: add w8, w22, #0xb
    0x104184514 <+180>: adr x24, #0x1041ddfe0
    0x104184518 <+184>: nop
    0x10418451c <+188>: ldr x8, [x24, w8, sxtw #3]
    0x104184520 <+192>: sub x8, x8, #2
    0x104184524 <+196>: mov w0, #0x18
    0x104184528 <+200>: str x8, [sp, #0x60]
    0x10418452c <+204>: blr x8
```

其中触发了 `onReceive`，对应输出的log是：

```
+++ into iterator: startAddress=0x194d31f44, isAppCode=false
+++ into iterator: startAddress=0x102e327ec, isAppCode=true
    0x102e327ec  +9100 : mov w8, #0x6beb
    0x102e327f0  +9104 : movk w8, #0x3920, lsl #16
    0x102e327f4  +9108 : add w24, w21, w23
    0x102e327f8  +9112 : str xzr, [x20]
    0x102e327fc  +9116 : str w8, [x20, #8]
    0x102e32800  +9120 : mov w8, #0xf28c
    0x102e32804  +9124 : movk w8, #0x3820, lsl #16
    0x102e32808  +9128 : str w8, [x20, #0xc]
    0x102e3280c  +9132 : mov x8, #0x77e8
    0x102e32810  +9136 : movk x8, #0xcfa6, lsl #16
    0x102e32814  +9140 : movk x8, #0xde0b, lsl #32
    0x102e32818  +9144 : movk x8, #0x20cd, lsl #48
    0x102e3281c  +9148 : add x0, x19, x8
    0x102e32820  +9152 : blr x27
+++ into iterator: startAddress=0x194d30564, isAppCode=false
>>> into onReceive: parsedEvents=ret,0x1084e0038,0x1085da104,0,ret,0x1085da134,0x108588
25c,-1,ret,0x108588318,0x1036540b0,-2,ret,0x10365413c,0x10984000c,-3,ret,0x1ddc2abc0,0x
194d2e05c,0,ret,0x194d2fbdc,0x194d2dc74,1,ret,0x194d311c4,0x194d2d830,1,ret,0x194d2d898
,0x194d2e0a0,0,ret,0x1ddc2ac00,0x194d2e374,0,ret,0x194d2e3a8,0x194d31214,-1,ret,0x194d3
12e8,0x194d30478,-2,ret,0x194d304a4,0x102e30530,-3,ret,0x1ddc2abc0,0x194d2cd44,0,ret,0x
194d31894,0x194d2f0a0,1,ret,0x194d31b3c,0x194d2f0c0,1,ret,0x194d2f1ac,0x194d2cd88,0,ret
,0x1ddc2ac00,0x194d2d280,0,ret,0x194d2d2b4,0x194d3123c,-1,ret,0x194d312e8,0x194d30478,-
2,ret,0x194d304a4,0x102e305ac,-3,ret,0x1bd80959c,0x102e244e0,-2,ret,0x1ddc36aec,0x183f7
94d8,0,ret,0x183f77ab0,0x183f7958c,0,ret,0x1bd808ca8,0x1bd809c04,1,ret,0x1bd809c40,0x18
3f795a0,0,ret,0x183f7970c,0x183f795e4,0,ret,0x183f7963c,0x183f7aa34,-1,ret,0x183f7aa98,
0x102e24508,-2,ret,0x1bd808b70,0x1bd809194,-1,ret,0x1bd809240,0x102e24528,-2,ret,0x102e
```

```
24630,0x102e315c4,-3,ret,0x1ddc2abc0,0x194d2e05c,0,ret,0x194d311c4,0x194d2d830,1,ret,0x
194d2d898,0x194d2e0a0,0,ret,0x1ddc2ac00,0x194d2e374,0,ret,0x194d2e3a8,0x194d31214,-1,re
t,0x194d312e8,0x194d30478,-2,ret,0x194d304a4,0x102e316b8,-3,ret,0x1ddc2abc0,0x194d2cd44
,0,ret,0x194d31894,0x194d2efe4,1,ret,0x194d2f1ac,0x194d2cd88,0,ret,0x1ddc2ac00,0x194d2d
280,0,ret,0x194d2d2b4,0x194d3123c,-1,ret,0x194d312e8,0x194d30478,-2,ret,0x194d304a4,0x1
02e31744,-3,ret,0x1bd808aa4,0x1bd809b78,-2,ret,0x1bd809bb0,0x102e32168,-3,ret,0x194d305
c4,0x194d30a34,-1,ret,0x194d33380,0x194d30a50,-1,ret,0x194d30a1c,0x194d2c188,-2,ret,0x1
ddc2abc0,0x194d31c80,-2,ret,0x194d2c06c,0x194d31dec,-2,ret,0x194d31894,0x194d31e00,-2,r
et,0x194d31b3c,0x194d31eec,-2,ret,0x194d30e90,0x194d31f34,-2,ret,0x1ddc2ac00,0x102e327e
c,-3
+++ into iterator: startAddress=0x194d30584, isAppCode=false
+++ into iterator: startAddress=0x194d30588, isAppCode=false
```

再往后，就是：**非原始函数代码** 和 **原始函数代码** 交替输出

而我们此处只关心 **原始函数代码**，去调试输出即可。

比如最后此刻所调试的 `+8516` 行的代码

```
+++ into iterator: startAddress=0x1041865a0, isAppCode=true
    0x1041865a0 <+8512>: mov w8, #0
    0x1041865a4 <+8516>: str x9, [x10]
curOffsetInt=8516
    0x1041865a8 <+8520>: b #0x104186740
contextStr={"pc":"0x1041865a4","sp":"0x16bf46740","nzcv":1610612736,"x0":"0x0","x1":"0x
104ba0000","x2":"0xc","x3":"0x29","x4":"0x470b","x5":"0x0","x6":"0x0","x7":"0xec0","x8":
"0x0","x9":"0x3050500","x10":"0x16bf46838","x11":"0x6c245a87","x12":"0x6c245a3d","x13":
"0x1041864dc","x14":"0x45","x15":"0x0","x16":"0xd5709bdf0d7a48f0","x17":"0x123209fc0","
x18":"0x0","x19":"0xdf3221f55389ecc8","x20":"0x1233064b0","x21":"0x93dba5a2","x22":"0x0"
,"x23":"0x39207beb","x24":"0x6c245a87","x25":"0x1041a6cb0","x26":"0x6c245a87","x27":"0x
194d2c100","x28":"0xfffffffffffffffe","fp":"0x16bf46820","lr":"0x104186168","q0":{},"q1"
:{},"q2":{},"q3":{},"q4":{},"q5":{},"q6":{},"q7":{},"q8":{},"q9":{},"q10":{},"q11":{},"
q12":{},"q13":{},"q14":{},"q15":{},"q16":{},"q17":{},"q18":{},"q19":{},"q20":{},"q21":{}
,"q22":{},"q23":{},"q24":{},"q25":{},"q26":{},"q27":{},"q28":{},"q29":{},"q30":{},"q31":
{},"d0":0.5585262685374610e-308,"d1":0.0000169759663317e-308,"d2":7.9499288951273454e-2
75,"d3":0,"d4":0,"d5":0,"d6":0,"d7":0,"d8":0,"d9":0,"d10":0,"d11":0,"d12":0,"d13":0,"d1
4":0,"d15":0,"d16":1.0855813867524649e+251,"d17":0.000002084665455e-308,"d18":3.055469
8911305689e-152,"d19":0.000001322084152e-308,"d20":3.7696966457559071e-175,"d21":1.0004
894955531618e+128,"d22":9.5944260775693913e+225,"d23":-6.8154492514793903e-236,"d24":-2
.1930923744387854e+50,"d25":-3.1153191556287330e+141,"d26":-1.0031492505499153e+302,"d2
7":-3.1584786060327890e-284,"d28":-1.7036104465458726e+239,"d29":-1.1538744009003510e-1
97,"d30":2.4944484440662111e-259,"d31":0,"s0":1.1035169354619356e-39,"s1":1.1210387714598533e-44,"s2":3.8204714345426298e-37,"s3":0,"s4":0,"s5":0,"s6":0,"s7":0,"s8":0,"s9":0,"s10":0,"s11":0,"s12":0,"s13":0,"s14":0,"s15":0,"s16":1.5548730664783592e-21,"s17":-1.3084408235578860e+36,"s18":-8.6088756618229034e-23,"s19":-5.408784745233076e-20,"s20":4.2151578028428229e+37,"s21":18362722504671230,"s22":3.372155249992253e+28,"s23":-7.6784630118876258e-30,"s24":-3715189.5,"s25":2.6426119211413095e+23,"s26":-3.5724724140742379e-23,"s27":-1.6239861190373424e-18,"s28":-205533200,"s29":-0.008124308660626411,"s30":-5.64249413550049896-36,"s31":0}
x9Value1=0x3050500, x9Value2=0x3050500
```

输出了我们所希望的值：

- `x9 = 0x3050500`

即可实现特定的调试目的。

libInFieldCollection.dylib的vsPHbdGf

代码 `fridaStalker_vsPHbdGf.js`

贴出之前的某个Frida的Stalker的完整调试代码：

- `/Users/crifan/dev/dev_root/iosReverse/iOSModifyActivation/subBin/libInFieldCollection_dylib/dynamicDebug/frida/frida_stalker/fridaStalker_vsPHbdGf.js`

```
/**
 * Function: frida Stakler to hook libInFieldCollection.dylib funtion vsPHbdGf
 * Author: Crifan Li
 * Update: 20241217
 * Usage:
 *  cd /Users/crifan/dev/dev_root/iosReverse/iOSModifyActivation/subBin/libInFieldCollection_dylib/dynamicDebug/frida/frida_stalker
 *  frida -U -n mobileactivationd -l fridaStalker_vsPHbdGf.js
*/


// https://github.com/crifan/JsFridaUtil/blob/main/JsUtil.js
// pure JavaScript utils
class JsUtil {

  constructor() {
    console.log("JsUtil constructor")
  }

  static {
  }

  /*---------- Number(Int) ----------*/

  static intToHexStr(intValue, prefix="0x", isUpperCase=true){
    // var hexStr = prefix + intValue.toString(16)
    // var hexStr = prefix + String(intValue).padStart(2, "0")
    // var hexStr = prefix + intValue.toString(16).padStart(2, "0")
    var intHexStr = intValue.toString(16)
    // console.log("intHexStr=" + intHexStr)
    var padding0HexStr = intHexStr.padStart(2, "0")
    // console.log("padding0HexStr=" + padding0HexStr)
    if (isUpperCase) {
      padding0HexStr = padding0HexStr.toUpperCase()
      // console.log("padding0HexStr=" + padding0HexStr)
    }
    var fullHexStr = prefix + padding0HexStr
    // console.log("fullHexStr=" + fullHexStr)
    return fullHexStr
  }

  /*---------- Log ----------*/
```

```
  // Generate single line log string
  // input: logStr="Called: -[NSURLRequest initWithURL:]"
  // output: "============================== Called: -[NSURLRequest initWithURL:] ====
==========================="
  static generateLineStr(logStr, isWithSpace=true, delimiterChar="=", lineWidth=80){
    // console.log("logStr=" + logStr, ", isWithSpace=" + isWithSpace + ", delimiterCha
r=" + delimiterChar + ", lineWidth=" + lineWidth)
    var lineStr = ""

    var realLogStr = ""
    if (isWithSpace) {
      realLogStr = " " + logStr + " "
    } else {
      realLogStr = logStr
    }

    var realLogStrLen = realLogStr.length
    if ((realLogStrLen % 2) > 0){
      realLogStr += " "
      realLogStrLen = realLogStr.length
    }

    var leftRightPaddingStr = ""
    var paddingLen = lineWidth - realLogStrLen
    if (paddingLen > 0) {
      var leftRightPaddingLen = paddingLen / 2
      leftRightPaddingStr = JsUtil.times(delimiterChar, leftRightPaddingLen)
    }

    lineStr = leftRightPaddingStr + realLogStr + leftRightPaddingStr

    // console.log("lineStr:\n" + lineStr)
    return lineStr
  }

  static logStr(curStr, isWithSpace=true, delimiterChar="=", lineWidth=80){
    // let delimiterStr = "--------------------"
    // console.log(delimiterStr + " " + curStr + " " + delimiterChar)
    var lineStr = JsUtil.generateLineStr(curStr, isWithSpace, delimiterChar, lineWidth)
    console.log(lineStr)
  }


  /*---------- Object: Dict/List/... ----------*/

  // convert Object(dict/list/...) to JSON string
  // function toJsonStr(curObj, singleLine=false, space=2){
  static 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()
  }

  /*---------- List ----------*/

  // check whether is item inside the list
  // eg: curItem="abc", curList=["abc", "def"] => true
  static isItemInList(curItem, curList){
    // method1:
    return curList.includes(curItem)
    // // method2:
    // return curList.indexOf(curItem) > -1
  }

  static sortByKey(curList, keyName){
    if (null != curList){
      curList.sort(function(objA, objB) {
        var valueA = objA[keyName]
        var valueB = objB[keyName]
        var valudDiff = valueA - valueB
        // console.log("valueA=" + valueA + ", valueB=" + valueB + " -> valudDiff=" + v
aludDiff)
        return valudDiff
      })
    }
  }

  /*---------- String ----------*/

  /** Function that count occurrences of a substring in a string;
   * @param {String} string               The string
   * @param {String} subString            The sub string to search for
   * @param {Boolean} [allowOverlapping]  Optional. (Default:false)
   *
   * @author Vitim.us https://gist.github.com/victornpb/7736865
   * @see Unit Test https://jsfiddle.net/Victornpb/5axuh96u/
   * @see https://stackoverflow.com/a/7924240/938822
   */
  static occurrences(string, subString, allowOverlapping) {
    // console.log("string=" + string + ",subString=" + subString + ", allowOverlapping
=" + allowOverlapping)
    string += "";
    subString += "";
    if (subString.length <= 0) return (string.length + 1);

    var n = 0,
      pos = 0,
      step = allowOverlapping ? 1 : subString.length;

    while (true) {
      pos = string.indexOf(subString, pos);
      // console.log("pos=" + pos)
      if (pos >= 0) {
        ++n;
        pos += step;
      } else break;
```

```
    }

    return n;
  }

  // String multiple
  // eg: str="=", num=5 => "====="
  static times(str, num){
    return new Array(num + 1).join(str)
  }

  // check string is empty or null
  static strIsEmpty(curStr){
    var isNull = null == curStr
    var isEmp = "" === curStr
    return isNull || isEmp
  }

  /*---------- Byte ----------*/

  // byte decimaal to byte hex
  // eg:
  //    8 => 8
  //    -60 => c4
  // function byteDecimalToByteHex(byteDecimal) {
  static byteDecimalToByteHex(byteDecimal) {
    // var digitCount = 6
    var digitCount = 2
    var minusDigitCount = 0 - digitCount
    // return (byteDecimal + Math.pow(16, 6)).toString(16).substr(-6)
    // var hexStr = (byteDecimal + Math.pow(16, 2)).toString(16).substr(-2)
    // return (byteDecimal + Math.pow(16, digitCount)).toString(16).substr(minusDigitCo
unt)
    var hexStr = (byteDecimal + Math.pow(16, digitCount)).toString(16).substr(minusDigi
tCount)
    // console.log("typeof hexStr=" + (typeof hexStr))
    // console.log("hexStr=" + hexStr)
    var hexValue = parseInt(hexStr, 16)
    // console.log("typeof hexValue=" + (typeof hexValue))
    // console.log("hexValue=" + hexValue)
    return hexValue
  }

  /*---------- Object ----------*/

  // check is js string
  static isJsStr(curObj){
    // console.log("curObj=" + curObj)
    var curObjType = (typeof curObj)
    // console.log("curObjType=" + curObjType)
    var isStr = curObjType === "string"
    // console.log("isStr=" + isStr)
    return isStr
  }

  /*---------- Pointer ----------*/
```

```
  // check pointer is valid or not
  // example
  //         0x103e79560 => true
  //         0xc => false
  static isValidPointer(curPtr){
    let MinValidPointer = 0x10000
    var isValid = curPtr > MinValidPointer
    // console.log("curPtr=" + curPtr, " -> isValid=" + isValid)
    return isValid
  }

}

// https://github.com/crifan/JsFridaUtil/blob/main/frida/FridaUtil.js
// Frida Common Util
class FridaUtil {
  // for Stalker onEnter transform, is show opcode string or not
  static isShowOpcode = true

  constructor() {
    console.log("FridaUtil constructor")
    console.log("FridaUtil Process.platform=" + Process.platform)
  }

  static isiOS(){
    var platform = Process.platform
    // console.log("platform=" + platform)
    var isJavaAvailable = Java.available
    // console.log("isJavaAvailable=" + isJavaAvailable)
    var isDarwin = platform === "darwin"
    // console.log("isDarwin=" + isDarwin)
    var isiOSOS = (!isJavaAvailable) && isDarwin
    // console.log("isiOSOS=" + isiOSOS)
    return isiOSOS
  }

  static isAndroid(){
    var platform = Process.platform
    // console.log("platform=" + platform)
    var isJavaAvailable = Java.available
    // console.log("isJavaAvailable=" + isJavaAvailable)
    var isLinux = platform === "linux"
    // console.log("isLinux=" + isLinux)
    var isAndroidOS = isJavaAvailable && isLinux
    // console.log("isAndroidOS=" + isAndroidOS)
    return isAndroidOS
  }

  // Frida pointer to UTF-8 string
  static ptrToUtf8Str(curPtr){
    var curUtf8Str = curPtr.readUtf8String()
    // console.log("curUtf8Str=" + curUtf8Str)
    return curUtf8Str
  }
```

```
  // Frida pointer to C string
  static ptrToCStr(curPtr){
    // var curCStr = Memory.readCString(curPtr)
    var curCStr = curPtr.readCString()
    // var curCStr = curPtr.readUtf8String()
    // console.log("curCStr=" + curCStr)
    return curCStr
  }

  static genModuleInfoStr(foundModule){
    // console.log("Module: name=" + foundModule.name + ", base=" + foundModule.base +
", size=" + foundModule.size + ", path=" + foundModule.path)
    var endAddress = foundModule.base.add(foundModule.size)
    var sizeHexStr = JsUtil.intToHexStr(foundModule.size)
    // console.log("Module: name=" + foundModule.name + ", address=[" + foundModule.bas
e + "-" + endAddress + "], size=" + sizeHexStr + "=" + foundModule.size + ", path=" + f
oundModule.path)
    var moduleInfoStr = "Module: address=[" + foundModule.base + "-" + endAddress + "],
 name=" + foundModule.name + ", size=" + sizeHexStr + "=" + foundModule.size + ", path="
 + foundModule.path
    return moduleInfoStr
  }

  // print module basic info: name, base, size, path
  static printModuleBasicInfo(foundModule){
    var moduleInfoStr = FridaUtil.genModuleInfoStr(foundModule)
    console.log(moduleInfoStr)
  }

  // print module symbols
  static printModuleSymbols(foundModule){
    var curSymbolList = foundModule.enumerateSymbols()
    console.log("Symbol: length=" + curSymbolList.length + ", list=" + curSymbolList)
    for(var i = 0; i < curSymbolList.length; i++) {
      console.log("---------- Symbol [" + i + "]----------")
      var curSymbol = curSymbolList[i]
      var sectionStr = JSON.stringify(curSymbol.section)
      console.log("name=" + curSymbol.name + ", address=" + curSymbol.address + "isGlob
al=" + curSymbol.isGlobal + ", type=" + curSymbol.type + ", section=" + sectionStr)
    }
  }

  // print module exports
  static printModuleExports(foundModule){
    var curExportList = foundModule.enumerateExports()
    console.log("Export: length=" + curExportList.length + ", list=" + curExportList)
    for(var i = 0; i < curExportList.length; i++) {
      console.log("---------- Export [" + i + "]----------")
      var curExport = curExportList[i]
      console.log("type=" + curExport.type + ", name=" + curExport.name + ", address=" +
 curExport.address)
    }
  }

  // print module info
  static printModuleInfo(moduleName){
```

```
    const foundModule = Module.load(moduleName)
    // const foundModule = Module.ensureInitialized()
    console.log("foundModule=" + foundModule)

    if (null == foundModule) {
      return
    }

    FridaUtil.printModuleBasicInfo(foundModule)

    FridaUtil.printModuleSymbols(foundModule)
    FridaUtil.printModuleExports(foundModule)
  }

  // print process basic info
  static printProcessBasicInfo(){
    console.log(
      "Process: id=" + Process.id
      + ", currentThreadId=" + Process.getCurrentThreadId()
      + ", currentDir=" + Process.getCurrentDir()
      + ", homeDir=" + Process.getHomeDir()
      + ", tmpDir=" + Process.getTmpDir()
      + ", arch=" + Process.arch
      + ", platform=" + Process.platform
      + ", pageSize=" + Process.pageSize
      + ", pointerSize=" + Process.pointerSize
      + ", codeSigningPolicy=" + Process.codeSigningPolicy
      + ", isDebuggerAttached=" + Process.isDebuggerAttached()
    )
  }

  // print all loaded modules basic info of current process
  //  Note: similar to `image list` in lldb
  static printAllLoadedModules(isSort=true){
    FridaUtil.printProcessBasicInfo()

    var moduleList = []

    Process.enumerateModules({
      onMatch: function(module){
        // console.log('Module name: ' + module.name + " - " + "Base Address: " + modul
e.base.toString());
        // FridaUtil.printModuleBasicInfo(module)
        moduleList.push(module)
      },
      onComplete: function(){}
    })

    if (isSort) {
      // moduleList.sort(function(moduleA, moduleB) {
      //   // var isLarge = moduleA.base > moduleB.base
      //   // console.log("moduleA.base=" + moduleA.base + ", moduleB.base=" + moduleB.
base + " -> isLarge=" + isLarge)
      //   var addrDiff = moduleA.base - moduleB.base
      //   console.log("moduleA.base=" + moduleA.base + ", moduleB.base=" + moduleB.bas
e + " -> addrDiff=" + addrDiff)
```

```
    //   return addrDiff
    // })
    JsUtil.sortByKey(moduleList, "base")
  }

  for(var i = 0; i < moduleList.length; i++) {
    var curModule = moduleList[i]
    // var prefixStr = "\t"
    var prefixStr = "  "
    console.log(prefixStr + FridaUtil.genModuleInfoStr(curModule))
  }

}

static printModuleInfoAndStalkerExclude(moduleName){
  var foundModule = Process.getModuleByName(moduleName)
  console.log("moduleName=" + moduleName + " -> foundModule=" + foundModule)
  if (null != foundModule) {
    Stalker.exclude(foundModule)
    // console.log("Stalker.exclude for module:")
    // FridaUtil.printModuleBasicInfo(foundModule)
    console.log("Stalker.exclude for: " + FridaUtil.genModuleInfoStr(foundModule))
  }
}

// print function call and stack, output content type is: address
static printFunctionCallStack_addr(curContext, prefix=""){
  var backtracerType = Backtracer.ACCURATE
  // var backtracerType = Backtracer.FUZZY
  if (!JsUtil.strIsEmpty(prefix)){
    prefix = prefix + " "
  }
  // const linePrefix = "\n"
  // const linePrefix = "\n\t"
  const linePrefix = "\n  "
  // const linePrefix = "\n "
  // const linePrefix = "\n"
  console.log(prefix + 'Stack:' + linePrefix +
    Thread.backtrace(curContext, backtracerType)
    .map(DebugSymbol.fromAddress).join(linePrefix) + '\n');
}

static dumpMemory(toDumpPtr, byteLen=128){
  var buf = toDumpPtr.readByteArray(byteLen)
  var dumpHexStr = hexdump(
    buf,
    {
      offset: 0,
      length: byteLen,
      header: true,
      ansi: true
    }
  )
  console.log("dumpHexStr=\n" + dumpHexStr)
}
```

```
  // convert ByteArray to Opcode string
  static byteArrayToOpcodeStr(byteArr){
    var byteStrList = []
    for(var i = 0; i < byteArr.length; i++) {
      var curByte = byteArr[i]
      // console.log("curByte=" + curByte)
      var curByteStr = JsUtil.intToHexStr(curByte, "", true)
      // console.log("curByteStr=" + curByteStr)
      byteStrList.push(curByteStr)
    }
    // console.log("byteStrList=" + byteStrList)
    var opcodeStr = byteStrList.join(" ")
    // console.log("byteArr=" + byteArr + " -> opcodeStr=" + opcodeStr)
    return opcodeStr
  }

  // read byte array from address
  // Note: curAddress is NativePointer
  static readAddressByteArray(curAddress, byteSize){
    // console.log("curAddress=" + curAddress + ", byteSize=" + byteSize)
    // var instructionByteArrBuffer = curAddress.readByteArray(byteSize)
    var curByteArray = []
    for(var i = 0; i < byteSize; i++){
      var curAddr = curAddress.add(i)
      // console.log("curAddr=" + curAddr)
      var byteU8 = curAddr.readU8()
      // console.log("byteU8=" + byteU8)
      curByteArray.push(byteU8)
    }
    // console.log("curByteArray=" + curByteArray)
    return curByteArray
  }

  static genInstructionOpcodeStr(instruction){
    var instructionByteArr = FridaUtil.readAddressByteArray(instruction.address, instru
ction.size)
    // console.log("instructionByteArr=" + instructionByteArr)

    // var instructionOpcodeStr = hexdump(
    //   instructionByteArr,
    //   {
    //     offset: 0,
    //     length: curInstructionSize,
    //     header: false,
    //     ansi: false
    //   }
    // )
    var instructionOpcodeStr = FridaUtil.byteArrayToOpcodeStr(instructionByteArr)
    // console.log("instructionOpcodeStr=" + instructionOpcodeStr)
    return instructionOpcodeStr
  }

  static printInstructionInfo(instruction){
    // Instruction: address=0x252c0edf8,toString()=br x10,next=0x4,size=4,mnemonic=br,o
pStr=x10,operands=[{"type":"reg","value":"x10","access":"r"}],regsAccessed={"read":["x1
0"],"written":[]},regsRead=[],regsWritten=[],groups=["jump"],toJSON()={"address":"0x252
```

```
c0edf8","next":"0x4","size":4,"mnemonic":"br","opStr":"x10","operands":[{"type":"reg","
value":"x10","access":"r"}],"regsAccessed":{"read":["x10"],"written":[]},"regsRead":[],
"regsWritten":[],"groups":["jump"]}
    console.log("Instruction: address=" + instruction.address
      + ",toString()=" + instruction.toString()
      + ",toJSON()=" + JSON.stringify(instruction.toJSON())
      // + ",next=" + instruction.next
      // + ",size=" + instruction.size
      // + ",mnemonic=" + instruction.mnemonic
      // + ",opStr=" + instruction.opStr
      // + ",operands=" + JSON.stringify(instruction.operands)
      // + ",regsAccessed=" + JSON.stringify(instruction.regsAccessed)
      // + ",regsRead=" + JSON.stringify(instruction.regsRead)
      // + ",regsWritten=" + JSON.stringify(instruction.regsWritten)
      // + ",groups=" + JSON.stringify(instruction.groups)
    )
  }

  // Frida Stalker hoo unknown name native function
  static stalkerHookUnnameNative(moduleBaseAddress, funcRelativeStartAddr, functionSize,
 argNum, hookFuncMap){
    console.log("Frida Stalker hook: module: baseAddress=" + moduleBaseAddress + ", isS
howOpcode=" + FridaUtil.isShowOpcode)

    var functionSizeHexStr = JsUtil.intToHexStr(functionSize)
    var funcRelativeStartAddrHexStr = JsUtil.intToHexStr(funcRelativeStartAddr)
    var funcRelativeEndAddr = funcRelativeStartAddr + functionSize
    var funcRelativeEndAddrHexStr = JsUtil.intToHexStr(funcRelativeEndAddr)
    console.log("function: relativeStartAddr=" + funcRelativeStartAddrHexStr + ", size="
 + functionSize + "=" + functionSizeHexStr + ", relativeEndAddr=" + funcRelativeEndAddr
HexStr)

    const funcRealStartAddr = moduleBaseAddress.add(funcRelativeStartAddr)
    // var funcRealEndAddr = funcRealStartAddr + functionSize
    const funcRealEndAddr = funcRealStartAddr.add(functionSize)
    console.log("funcRealStartAddr=" + funcRealStartAddr + ", funcRealEndAddr=" + funcR
ealEndAddr)
    var curTid = null
    console.log("curTid=" + curTid)
    Interceptor.attach(funcRealStartAddr, {
      onEnter: function(args) {
        JsUtil.logStr("Trigged addr: relative [" + funcRelativeStartAddrHexStr + "] = r
eal [" + funcRealStartAddr + "]")

        for(var i = 0; i < argNum; i ++) {
          var curArg = args[i]
          console.log("arg[" + i + "]=" + curArg)
        }

        var curTid = Process.getCurrentThreadId()
        console.log("curTid=" + curTid)
        Stalker.follow(curTid, {
            events: {
              call: false, // CALL instructions: yes please
              ret: true, // RET instructions
              exec: false, // all instructions: not recommended as it's
```

```
            block: false, // block executed: coarse execution trace
            compile: false // block compiled: useful for coverage
          },
          // onReceive: Called with `events` containing a binary blob comprised of on
e or more GumEvent structs. See `gumevent.h` for details about the format. Use `Stalker
.parse()` to examine the data.
          onReceive(events) {
            var parsedEvents = Stalker.parse(events)
            // var parsedEventsStr = JSON.stringify(parsedEventsStr)
            // console.log(">>> into onReceive: parsedEvents=" + parsedEvents + ", pa
rsedEventsStr=" + parsedEventsStr);
            console.log(">>> into onReceive: parsedEvents=" + parsedEvents);
          },

          // transform: (iterator: StalkerArm64Iterator) => {
          transform: function (iterator) {
            // https://www.radare.org/doc/frida/interfaces/StalkerArmIterator.html

            // console.log("iterator=" + iterator)
            var instruction = iterator.next()
            const startAddress = instruction.address
            // console.log("+++ into iterator: startAddress=" + startAddress)
            // const isAppCode = startAddress.compare(funcRealStartAddr) >= 0 && star
tAddress.compare(funcRealEndAddr) === -1
            // const isAppCode = (startAddress.compare(funcRealStartAddr) >= 0) && (s
tartAddress.compare(funcRealEndAddr) < 0)
            const gt_realStartAddr = startAddress.compare(funcRealStartAddr) >= 0
            const lt_realEndAddr = startAddress.compare(funcRealEndAddr) < 0
            var isAppCode = gt_realStartAddr && lt_realEndAddr
            console.log("+++ into iterator: startAddress=" + startAddress + ", isAppC
ode=" + isAppCode)

            // // for debug
            // isAppCode = true

            // console.log("isAppCode=" + isAppCode + ", gt_realStartAddr=" + gt_real
StartAddr + ", lt_realEndAddr=" + lt_realEndAddr)
            do {
              if (isAppCode) {
                // is origal function code = which we focus on
                // FridaUtil.printInstructionInfo(instruction)

                var curRealAddr = instruction.address
                // console.log("curRealAddr=" + curRealAddr)
                // const isAppCode = curRealAddr.compare(funcRealStartAddr) >= 0 && c
urRealAddr.compare(funcRealEndAddr) === -1
                // console.log(curRealAddr + ": isAppCode=" + isAppCode)
                var curOffsetHexPtr = curRealAddr.sub(funcRealStartAddr)
                var curOffsetInt = curOffsetHexPtr.toInt32()
                console.log("current: realAddr=" + curRealAddr + " -> offset: hex=" +
 curOffsetHexPtr + "=" + curOffsetInt)

                // var instructionStr = instruction.mnemonic + " " + instruction.opStr

                var instructionStr = instruction.toString()
                // console.log("\t" + curRealAddr + ": " + instructionStr);
```

```
                // console.log("\t" + curRealAddr + " <+" + curOffsetHexPtr + ">: " +
 instructionStr)
                // console.log("\t" + curRealAddr + " <+" + curOffsetInt + ">: " + in
structionStr)

                var opcodeStr = ""
                if (FridaUtil.isShowOpcode) {
                  opcodeStr = " " + FridaUtil.genInstructionOpcodeStr(instruction)
                }
                var instructionFullLogStr = "\t" + curRealAddr + " <+" + curOffsetInt
 + ">" + opcodeStr + ": " + instructionStr
                console.log(instructionFullLogStr)
                // 0x252c0edf8 <+356>: br x10
                // 0x252c0edf8 <+356> 40 01 1F D6: br x10

                if (curOffsetInt in hookFuncMap){
                  console.log("offset: " + curOffsetHexPtr + "=" + curOffsetInt)
                  // let curHookFunc = hookFuncMap.get(curOffsetInt)
                  var curHookFunc = hookFuncMap[curOffsetInt]
                  // console.log("curOffsetInt=" + curOffsetInt + " -> curHookFunc="
 + curHookFunc)

                  // putCallout -> https://www.radare.org/doc/frida/interfaces/Stalke
rArmIterator.html#putCallout
                  // StalkerScriptCallout -> https://www.radare.org/doc/frida/types/S
talkerScriptCallout.html
                  // CpuContext -> https://www.radare.org/doc/frida/types/CpuContext.
html
                  // Arm64CpuContext -> https://www.radare.org/doc/frida/interfaces/A
rm64CpuContext.html

                  // work: normal
                  iterator.putCallout(curHookFunc)

                  // var extraDataDict = {
                  //   "curOffsetInt": curOffsetInt
                  // }
                  // Not work: abnormal
                  // iterator.putCallout((context) => {
                  // // iterator.putCallout((context, extraDataDict) => {
                  //   // console.log("match offset: " + curOffsetHexPtr + ", curReal
Addr=" + curRealAddr)
                  //   // curHookFunc(context, curOffsetInt, moduleBaseAddress)
                  //   // context.curOffsetInt = curOffsetInt
                  //   // context.curOffsetHexPtr = curOffsetHexPtr
                  //   // context.moduleBaseAddress = moduleBaseAddress
                  //   // context[curOffsetInt] = curOffsetInt
                  //   // context[curOffsetHexPtr] = curOffsetHexPtr
                  //   // context[moduleBaseAddress] = moduleBaseAddress
                  //   // curHookFunc(context, extraDataDict)
                  //   curHookFunc(context)
                  // })
                }

              }
              iterator.keep()
```

```
            } while ((instruction = iterator.next()) !== null)
          }
        });

        // function needDebug(context) {
        //     console.log("into needDebug")
        //     // console.log("into needDebug: context=" + context)
        //     // var contextStr = JSON.stringify(context, null, 2)
        //     // console.log("context=" + contextStr)
        //     // var x9Value1 = context.x9
        //     // var x9Value2 = context["x9"]
        //     // console.log("x9Value1=" + x9Value1 + ", x9Value2=" + x9Value2)
        // }
      },
      onLeave: function(retval) {
        console.log("addr: relative [" + funcRelativeStartAddrHexStr + "] real [" + fun
cRealStartAddr + "] -> retval=" + retval)
        if (curTid != null) {
          Stalker.unfollow(curTid)
          console.log("Stalker.unfollow curTid=", curTid)
        }
      }
    })
  }


}

/**********************************************************************************
 * Other Util
**********************************************************************************/

// function logMain(mainStr){
//   const mainDelimiter = "======================================="
//   console.log(mainDelimiter + " " + mainStr + " " + mainDelimiter)
// }

// function logSub(subStr){
//   const subDelimiter = "-----------------------------"
//   console.log(subDelimiter + " " + subStr + " " + subDelimiter)
// }

/**********************************************************************************
 * Main Hook
**********************************************************************************/

function stalkerHookNative_vsPHbdGf(){
  // console.log("into stalkerHookNative_vsPHbdGf:")
  // logSub("into stalkerHookNative_vsPHbdGf")
  JsUtil.logStr("into stalkerHookNative_vsPHbdGf", true, "-")

  var libraryName = "libInFieldCollection.dylib"

  // for debug
  // libraryName = "mobileactivationd"
```

```
  const moduleBaseAddress = Module.findBaseAddress(libraryName)
  console.log("libraryName=" + libraryName + " -> moduleBaseAddress=" + moduleBaseAddre
ss)
  if (null == moduleBaseAddress) {
    console.error("Failed to find library " + libraryName)
    return
  }

  var funcName = "vsPHbdGf"

  // for debug
  // -[MACollectionInterface copyIngestData:]    __text    0000000100067B4    00000D84
    00000270        R    .    .    .    .    .    B    T    .
  // funcName = "-[MACollectionInterface copyIngestData:]"
  // funcName = "DERDecodeItem"
  // funcName = "mach_msg"

  var origFuncPtr = Module.findExportByName(libraryName, funcName)
  console.log("origFuncPtr=" + origFuncPtr)

  if (null == origFuncPtr) {
    console.error("Failed to find function " + funcName + "in lib " + libraryName)
    return
  }

  var funcRelativeStartAddr = origFuncPtr - moduleBaseAddress
  console.log("funcRelativeStartAddr=" + funcRelativeStartAddr + "=" + JsUtil.intToHexS
tr(funcRelativeStartAddr))

  // for debug
  /*
    [iPhone::mobileactivationd ]-> ============= Trigged addr: relative [0x13c94] = real
 [0x1c280ec94]  =============
    ...
    +++ into iterator: startAddress=0x1c2813d78, isAppCode=true
    current: realAddr=0x1c2813d78 -> offset: hex=0x50e4=20708
        0x1c2813d78 <+20708>: str x0, [sp, #0xc8]
    current: realAddr=0x1c2813d7c -> offset: hex=0x50e8=20712
        0x1c2813d7c <+20712>: bl #0x1c2842d88
  */
  // // use later address as start address, try to avoid only hook part code == not hoo
k all code
  // funcRelativeStartAddr = funcRelativeStartAddr + 0x50e4

  var functionSize = 0x61FC
  // for debug: make larger, tyr to support hook more code
  // var functionSize = 0x10000

  // CollectPCRT == void __fastcall vsPHbdGf(__int64 a1, __int64 a2)
  var argNum = 2
  console.log("functionSize=" + functionSize + "=" + JsUtil.intToHexStr(functionSize) +
", argNum=" + argNum)

  // try exclude not-concern functions inside common libs

  // // var moduleName_libObjc = "/usr/lib/libobjc.A.dylib"
```

```
  // var moduleName_libObjc = "libobjc.A.dylib"
  // // FridaUtil.printModuleInfo(moduleName_libObjc)
  // var module_libObjc = Process.getModuleByName(moduleName_libObjc)
  // console.log("module_libObjc=" + module_libObjc)
  // if (null != module_libObjc) {
  //   FridaUtil.printModuleBasicInfo(module_libObjc)

  //   Stalker.exclude(module_libObjc)
  // }

  // // var moduleName_system =
  // var moduleName_system = "libSystem.B.dylib"
  // // FridaUtil.printModuleInfo(moduleName_system)
  // var module_system = Process.getModuleByName(moduleName_system)
  // console.log("module_system=" + module_system)
  // if (null != module_system) {
  //   FridaUtil.printModuleBasicInfo(module_system)

  //   Stalker.exclude(module_system)
  // }

  // var moduleName_MobileGestalt = "libMobileGestalt.dylib"
  // var module_MobileGestalt = Process.getModuleByName(moduleName_MobileGestalt)
  // console.log("module_MobileGestalt=" + module_MobileGestalt)
  // if (null != module_MobileGestalt) {
  //   FridaUtil.printModuleBasicInfo(module_MobileGestalt)
  // }

  // var moduleName_IOKit = "IOKit"
  // var module_IOKit = Process.getModuleByName(moduleName_IOKit)
  // console.log("module_IOKit=" + module_IOKit)
  // if (null != module_IOKit) {
  //   FridaUtil.printModuleBasicInfo(module_IOKit)
  // }

  // var moduleName_Foundation = "Foundation"
  // var module_Foundation = Process.getModuleByName(moduleName_Foundation)
  // console.log("module_Foundation=" + module_Foundation)
  // if (null != module_Foundation) {
  //   FridaUtil.printModuleBasicInfo(module_Foundation)
  // }

  // var moduleName_CoreFoundation = "CoreFoundation"
  // var module_CoreFoundation = Process.getModuleByName(moduleName_CoreFoundation)
  // console.log("module_CoreFoundation=" + module_CoreFoundation)
  // if (null != module_CoreFoundation) {
  //   FridaUtil.printModuleBasicInfo(module_CoreFoundation)
  // }

  // FridaUtil.printModuleInfoAndStalkerExclude("libobjc.A.dylib") // /usr/lib/libobjc.
A.dylib
  // FridaUtil.printModuleInfoAndStalkerExclude("libSystem.B.dylib") // /usr/lib/libSys
tem.B.dylib
  // FridaUtil.printModuleInfoAndStalkerExclude("libsystem_malloc.dylib") // /usr/lib/s
ystem/libsystem_malloc.dylib
```

```
  FridaUtil.printAllLoadedModules()

  var toExcludeModuleList = [
    "libobjc.A.dylib", // /usr/lib/libobjc.A.dylib
    "libSystem.B.dylib", // /usr/lib/libSystem.B.dylib
    "libsystem_malloc.dylib", // /usr/lib/system/libsystem_malloc.dylib
    "libsystem_m.dylib", // /usr/lib/system/libsystem_m.dylib
    "libsystem_networkextension.dylib", // /usr/lib/system/libsystem_networkextension.d
ylib
    "libsystem_notify.dylib", // /usr/lib/system/libsystem_notify.dylib
    "libsystem_sandbox.dylib", // /usr/lib/system/libsystem_sandbox.dylib
    "libsystem_kernel.dylib", // /usr/lib/system/libsystem_kernel.dylib
    "libsystem_platform.dylib", // /usr/lib/system/libsystem_platform.dylib
    "libsystem_pthread.dylib", // /usr/lib/system/libsystem_pthread.dylib
    "libsystem_symptoms.dylib", // /usr/lib/system/libsystem_symptoms.dylib
    "libsystem_trace.dylib", // /usr/lib/system/libsystem_trace.dylib
    "libunwind.dylib", // /usr/lib/system/libunwind.dylib
    "libxpc.dylib", // /usr/lib/system/libxpc.dylib

    // Module: address=[0x19a60a000-0x19a60f000], name=libcache.dylib, size=0x5000=2048
0, path=/usr/lib/system/libcache.dylib
    "libcache.dylib",
    // Module: address=[0x19a60f000-0x19a61b000], name=libcommonCrypto.dylib, size=0xc0
00=49152, path=/usr/lib/system/libcommonCrypto.dylib
    "libcommonCrypto.dylib",
    // Module: address=[0x19a61b000-0x19a620000], name=libcompiler_rt.dylib, size=0x500
0=20480, path=/usr/lib/system/libcompiler_rt.dylib
    "libcompiler_rt.dylib",
    // Module: address=[0x19a620000-0x19a629000], name=libcopyfile.dylib, size=0x9000=3
6864, path=/usr/lib/system/libcopyfile.dylib
    "libcopyfile.dylib",
    // Module: address=[0x19a629000-0x19a68d000], name=libcorecrypto.dylib, size=0x6400
0=409600, path=/usr/lib/system/libcorecrypto.dylib
    "libcorecrypto.dylib",
    // Module: address=[0x19a68d000-0x19a6fd000], name=libdispatch.dylib, size=0x70000=
458752, path=/usr/lib/system/libdispatch.dylib
    "libdispatch.dylib",
    // Module: address=[0x19a6fd000-0x19a727000], name=libdyld.dylib, size=0x2a000=1720
32, path=/usr/lib/system/libdyld.dylib
    "libdyld.dylib",

    // Module: address=[0x22ab49000-0x22abcb000], name=libsystem_c.dylib, size=0x82000=
532480, path=/usr/lib/system/libsystem_c.dylib
    "libsystem_c.dylib",
    // Module: address=[0x22af96000-0x22b2f5000], name=CoreFoundation, size=0x35f000=35
34848, path=/System/Library/Frameworks/CoreFoundation.framework/CoreFoundation
    "CoreFoundation",
    // Module: address=[0x101720000-0x10172c000], name=MAMock.dylib, size=0xc000=49152,
 path=/Library/MobileSubstrate/DynamicLibraries/MAMock.dylib
    "MAMock.dylib",

    // // Module: address=[0x22b61f000-0x22b654000], name=libMobileGestalt.dylib, size=
0x35000=217088, path=/usr/lib/libMobileGestalt.dylib
    // "libMobileGestalt.dylib",
  ]
```

```
  for(var i = 0; i < toExcludeModuleList.length; i++) {
    // console.log("---------- Module [" + i + "]----------")
    var curModule = toExcludeModuleList[i]

    FridaUtil.printModuleInfoAndStalkerExclude(curModule)
  }

  let hookFuncMap = {
    0x164: // +356
      function (context) {
        var contextStr = JSON.stringify(context)
        console.log("[0x164] contextStr=" + contextStr)
        var x10 = context.x10
        console.log("[0x164] x10=" + x10)
      },
    0x1DC: // +476
      function (context) {
        var contextStr = JSON.stringify(context)
        console.log("[0x1DC] contextStr=" + contextStr)
        var x10 = context.x10
        console.log("[0x1DC] x10=" + x10)
      },
  }

  FridaUtil.isShowOpcode = false
  FridaUtil.stalkerHookUnnameNative(moduleBaseAddress, funcRelativeStartAddr, functionS
ize, argNum, hookFuncMap)
}

function hookNative(){
  // console.log("into hookNative:")
  // logMain("into hookNative")
  JsUtil.logStr("into hookNative")
  stalkerHookNative_vsPHbdGf()
}

hookNative()
```

输出日志

部分调试输出日志：

```
➜  frida_stalker frida -U -n mobileactivationd -l fridaStalker_vsPHbdGf.js
     ____
    / _  |   Frida 16.5.9 - A world class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
```

```
     . . . .   Connected to iPhone (id=9bad8d3c9530bd2fc62072c30ee3accf3cdc45d6)
Attaching...
=============================== into hookNative  ===============================
----------------------- into stalkerHookNative_vsPHbdGf  -----------------------
libraryName=libInFieldCollection.dylib -> moduleBaseAddress=null
Failed to find library libInFieldCollection.dylib
[iPhone::mobileactivationd ]-> %reload
=============================== into hookNative  ===============================
----------------------- into stalkerHookNative_vsPHbdGf  -----------------------
libraryName=libInFieldCollection.dylib -> moduleBaseAddress=null
Failed to find library libInFieldCollection.dylib
[iPhone::mobileactivationd ]-> %reload
=============================== into hookNative  ===============================
----------------------- into stalkerHookNative_vsPHbdGf  -----------------------
libraryName=libInFieldCollection.dylib -> moduleBaseAddress=null
Failed to find library libInFieldCollection.dylib
[iPhone::mobileactivationd ]-> %reload
=============================== into hookNative  ===============================
----------------------- into stalkerHookNative_vsPHbdGf  -----------------------
libraryName=libInFieldCollection.dylib -> moduleBaseAddress=null
Failed to find library libInFieldCollection.dylib
[iPhone::mobileactivationd ]-> %reload
=============================== into hookNative  ===============================
----------------------- into stalkerHookNative_vsPHbdGf  -----------------------
libraryName=libInFieldCollection.dylib -> moduleBaseAddress=0x252bfb000
origFuncPtr=0x252c0ec94
funcRelativeStartAddr=81044
functionSize=25084=0x61fc, argNum=2
Process: id=9871, currentThreadId=11303, currentDir=/, homeDir=/var/mobile, tmpDir=/var/
tmp, arch=arm64, platform=darwin, pageSize=16384, pointerSize=8, codeSigningPolicy=opti
onal, isDebuggerAttached=false
  Module: address=[0x100620000-0x10082c000], name=mobileactivationd, size=0x20c000=2146
304, path=/usr/libexec/mobileactivationd
  Module: address=[0x100884000-0x100888000], name=MobileSubstrate.dylib, size=0x4000=16
384, path=/Library/MobileSubstrate/MobileSubstrate.dylib
  Module: address=[0x1008cc000-0x1008d0000], name=SubstrateInserter.dylib, size=0x4000=
16384, path=/usr/lib/substrate/SubstrateInserter.dylib
  Module: address=[0x100a00000-0x100a18000], name=SubstrateLoader.dylib, size=0x18000=9
8304, path=/usr/lib/substrate/SubstrateLoader.dylib
  Module: address=[0x100a44000-0x100a9c000], name=dyld, size=0x58000=360448, path=/usr/
lib/dyld
  Module: address=[0x100b44000-0x100b50000], name=MAMock.dylib, size=0xc000=49152, path=
/Library/MobileSubstrate/DynamicLibraries/MAMock.dylib
  Module: address=[0x100d00000-0x100e14000], name=libsubstrate.dylib, size=0x114000=113
0496, path=/usr/lib/libsubstrate.dylib
  Module: address=[0x22a217000-0x22a219000], name=libSystem.B.dylib, size=0x2000=8192,
path=/usr/lib/libSystem.B.dylib
  Module: address=[0x22a219000-0x22a26f000], name=libc++.1.dylib, size=0x56000=352256,
path=/usr/lib/libc++.1.dylib
  Module: address=[0x22a26f000-0x22a282000], name=libc++abi.dylib, size=0x13000=77824,
path=/usr/lib/libc++abi.dylib
  Module: address=[0x22a282000-0x22aa0a000], name=libobjc.A.dylib, size=0x788000=7897088
, path=/usr/lib/libobjc.A.dylib
  Module: address=[0x22aa0a000-0x22aa0f000], name=libcache.dylib, size=0x5000=20480, pa
th=/usr/lib/system/libcache.dylib
  Module: address=[0x22aa0f000-0x22aa1b000], name=libcommonCrypto.dylib, size=0xc000=49
```

```
152, path: /usr/lib/system/libcommonCrypto.dylib
  Module: address: [0x22aa1b000-0x22aa20000], name: libcompiler_rt.dylib, size: 0x5000 (204
80), path: /usr/lib/system/libcompiler_rt.dylib
...
  Module: address: [0x22cbc2000-0x22cbc3000], name: vecLib, size: 0x1000 (4096), path: /System
/Library/Frameworks/Accelerate.framework/Frameworks/vecLib.framework/vecLib
...
  Module: address: [0x252bfb000-0x252c51000], name: libInFieldCollection.dylib, size: 0x56
000 (352256), path: /usr/lib/libInFieldCollection.dylib
  Module: address: [0x252c62000-0x252c70000], name: libMobileGestaltExtensions.dylib, size
: 0xe000 (57344), path: /usr/lib/libMobileGestaltExtensions.dylib
...
  Module: address: [0x256365000-0x2563fa000], name: SampleAnalysis, size: 0x95000 (610304),
path: /System/Library/PrivateFrameworks/SampleAnalysis.framework/SampleAnalysis
---------- Module [0]----------
moduleName:libobjc.A.dylib -> foundModule:[object Object]
Stalker.exclude for: Module: address: [0x22a282000-0x22aa0a000], name: libobjc.A.dylib, s
ize: 0x788000 (7897088), path: /usr/lib/libobjc.A.dylib
---------- Module [1]----------
moduleName:libSystem.B.dylib -> foundModule:[object Object]
Stalker.exclude for: Module: address: [0x22a217000-0x22a219000], name: libSystem.B.dylib,
 size: 0x2000 (8192), path: /usr/lib/libSystem.B.dylib
---------- Module [2]----------
moduleName:libsystem_malloc.dylib -> foundModule:[object Object]
Stalker.exclude for: Module: address: [0x22ac81000-0x22aca4000], name: libsystem_malloc.d
ylib, size: 0x23000 (143360), path: /usr/lib/system/libsystem_malloc.dylib
---------- Module [3]----------
moduleName:libsystem_m.dylib -> foundModule:[object Object]
Stalker.exclude for: Module: address: [0x22ac53000-0x22ac81000], name: libsystem_m.dylib,
 size: 0x2e000 (188416), path: /usr/lib/system/libsystem_m.dylib
---------- Module [4]----------
moduleName:libsystem_networkextension.dylib -> foundModule:[object Object]
Stalker.exclude for: Module: address: [0x22aca4000-0x22acb0000], name: libsystem_networke
xtension.dylib, size: 0xc000 (49152), path: /usr/lib/system/libsystem_networkextension.dyli
b
---------- Module [5]----------
moduleName:libsystem_notify.dylib -> foundModule:[object Object]
Stalker.exclude for: Module: address: [0x22acb0000-0x22acb7000], name: libsystem_notify.d
ylib, size: 0x7000 (28672), path: /usr/lib/system/libsystem_notify.dylib
---------- Module [6]----------
moduleName:libsystem_sandbox.dylib -> foundModule:[object Object]
Stalker.exclude for: Module: address: [0x22acd3000-0x22acd6000], name: libsystem_sandbox.
dylib, size: 0x3000 (12288), path: /usr/lib/system/libsystem_sandbox.dylib
---------- Module [7]----------
moduleName:libsystem_kernel.dylib -> foundModule:[object Object]
Stalker.exclude for: Module: address: [0x22ac28000-0x22ac53000], name: libsystem_kernel.d
ylib, size: 0x2b000 (176128), path: /usr/lib/system/libsystem_kernel.dylib
---------- Module [8]----------
moduleName:libsystem_platform.dylib -> foundModule:[object Object]
Stalker.exclude for: Module: address: [0x22acb7000-0x22acc2000], name: libsystem_platform.
dylib, size: 0xb000 (45056), path: /usr/lib/system/libsystem_platform.dylib
---------- Module [9]----------
moduleName:libsystem_pthread.dylib -> foundModule:[object Object]
Stalker.exclude for: Module: address: [0x22acc2000-0x22acd3000], name: libsystem_pthread.
dylib, size: 0x11000 (69632), path: /usr/lib/system/libsystem_pthread.dylib
---------- Module [10]----------
```

```
moduleName libsystem_symptoms.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22acd6000 0x22acde000], name libsystem_symptoms.
dylib, size 0x8000 32768, path /usr/lib/system/libsystem_symptoms.dylib
---------- Module [11]----------
moduleName libsystem_trace.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22acde000 0x22acf4000], name libsystem_trace.dy
lib, size 0x16000 90112, path /usr/lib/system/libsystem_trace.dylib
---------- Module [12]----------
moduleName libunwind.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22acf4000 0x22acfa000], name libunwind.dylib, s
ize 0x6000 24576, path /usr/lib/system/libunwind.dylib
---------- Module [13]----------
moduleName libxpc.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22acfa000 0x22ad28000], name libxpc.dylib, size
0x2e000 188416, path /usr/lib/system/libxpc.dylib
---------- Module [14]----------
moduleName libcache.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22aa0a000 0x22aa0f000], name libcache.dylib, si
ze 0x5000 20480, path /usr/lib/system/libcache.dylib
---------- Module [15]----------
moduleName libcommonCrypto.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22aa0f000 0x22aa1b000], name libcommonCrypto.dy
lib, size 0xc000 49152, path /usr/lib/system/libcommonCrypto.dylib
---------- Module [16]----------
moduleName libcompiler_rt.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22aa1b000 0x22aa20000], name libcompiler_rt.dyl
ib, size 0x5000 20480, path /usr/lib/system/libcompiler_rt.dylib
---------- Module [17]----------
moduleName libcopyfile.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22aa20000 0x22aa29000], name libcopyfile.dylib,
 size 0x9000 36864, path /usr/lib/system/libcopyfile.dylib
---------- Module [18]----------
moduleName libcorecrypto.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22aa29000 0x22aa8d000], name libcorecrypto.dylib
, size 0x64000 409600, path /usr/lib/system/libcorecrypto.dylib
---------- Module [19]----------
moduleName libdispatch.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22aa8d000 0x22aafd000], name libdispatch.dylib,
 size 0x70000 458752, path /usr/lib/system/libdispatch.dylib
---------- Module [20]----------
moduleName libdyld.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22aafd000 0x22ab27000], name libdyld.dylib, size
 0x2a000 172032, path /usr/lib/system/libdyld.dylib
---------- Module [21]----------
moduleName libsystem_c.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22ab49000 0x22abcb000], name libsystem_c.dylib,
 size 0x82000 532480, path /usr/lib/system/libsystem_c.dylib
---------- Module [22]----------
moduleName CoreFoundation -> foundModule [object Object]
Stalker.exclude for: Module: address [0x22af96000 0x22b2f5000], name CoreFoundation, si
ze 0x35f000 3534848, path /System/Library/Frameworks/CoreFoundation.framework/CoreFound
ation
---------- Module [23]----------
moduleName MAMock.dylib -> foundModule [object Object]
Stalker.exclude for: Module: address [0x100b44000 0x100b50000], name MAMock.dylib, size
0xc000 49152, path /Library/MobileSubstrate/DynamicLibraries/MAMock.dylib
```

```
Frida Stalker hook: module: baseAddress 0x252bfb000
function: relativeStartAddr 0x13c94, size 25084 0x61fc, relativeEndAddr 0x19e90
funcRealStartAddr 0x252c0ec94, funcRealEndAddr 0x252c14e90
curTid null
[iPhone  mobileactivationd ]-> ============ Trigged addr  relative [0x13c94] = real [0x
252c0ec94]  ============
arg[0] 0x16f866988
arg[1] 0x16f866984
curTid 21011
+++ into iterator  startAddress 0x10f3c1768, isAppCode false
+++ into iterator  startAddress 0x10f3c1778, isAppCode false
...
+++ into iterator  startAddress 0x11074c030, isAppCode false
+++ into iterator  startAddress 0x252c0eca4, isAppCode true
Instruction: address 0x252c0eca4,toString() stp x20, x19, [sp, #0x40],next 0x4,size 4,m
nemonic stp,opStr x20, x19, [sp, #0x40],operands [{"type":"reg","value":"x20","access":
"r"},{"type":"reg","value":"x19","access":"r"},{"type":"mem","value":{"base":"sp","disp"
:64},"access":"rw"}],regsAccessed {"read":["x20","x19","sp"],"written":[]},regsRead [],
regsWritten [],groups [],toJSON ={"address":"0x252c0eca4","next":"0x4","size":4,"mnemo
nic":"stp","opStr":"x20, x19, [sp, #0x40]","operands":[{"type":"reg","value":"x20","acc
ess":"r"},{"type":"reg","value":"x19","access":"r"},{"type":"mem","value":{"base":"sp",
"disp":64},"access":"rw"}],"regsAccessed":{"read":["x20","x19","sp"],"written":[]},"reg
sRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c0eca4 -> offset: hex=0x10=16
    0x252c0eca4 <+16>: stp x20, x19, [sp, #0x40]
Instruction: address 0x252c0eca8,toString() stp x29, x30, [sp, #0x50],next 0x4,size 4,m
nemonic stp,opStr x29, x30, [sp, #0x50],operands [{"type":"reg","value":"fp","access":"
r"},{"type":"reg","value":"lr","access":"r"},{"type":"mem","value":{"base":"sp","disp":
80},"access":"rw"}],regsAccessed {"read":["fp","lr","sp"],"written":[]},regsRead [],reg
sWritten [],groups [],toJSON ={"address":"0x252c0eca8","next":"0x4","size":4,"mnemonic"
:"stp","opStr":"x29, x30, [sp, #0x50]","operands":[{"type":"reg","value":"fp","access":
"r"},{"type":"reg","value":"lr","access":"r"},{"type":"mem","value":{"base":"sp","disp":
80},"access":"rw"}],"regsAccessed":{"read":["fp","lr","sp"],"written":[]},"regsRead":[],
"regsWritten":[],"groups":[]}
current: realAddr 0x252c0eca8 -> offset: hex=0x14=20
    0x252c0eca8 <+20>: stp x29, x30, [sp, #0x50]
Instruction: address 0x252c0ecac,toString() add x29, sp, #0x50,next 0x4,size 4,mnemonic
add,opStr x29, sp, #0x50,operands [{"type":"reg","value":"fp","access":"w"},{"type":"re
g","value":"sp","access":"r"},{"type":"imm","value":"80","access":"r"}],regsAccessed {"
read":["sp"],"written":["fp"]},regsRead [],regsWritten [],groups [],toJSON ={"address":
"0x252c0ecac","next":"0x4","size":4,"mnemonic":"add","opStr":"x29, sp, #0x50","operands"
:[{"type":"reg","value":"fp","access":"w"},{"type":"reg","value":"sp","access":"r"},{"t
ype":"imm","value":"80","access":"r"}],"regsAccessed":{"read":["sp"],"written":["fp"]},
"regsRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c0ecac -> offset: hex=0x18=24
    0x252c0ecac <+24>: add x29, sp, #0x50
Instruction: address 0x252c0ecb0,toString() sub sp, sp, #1, lsl #12,next 0x4,size 4,mne
monic sub,opStr sp, sp, #1, lsl #12,operands [{"type":"reg","value":"sp","access":"w"},{
"type":"reg","value":"sp","access":"r"},{"type":"imm","value":"1","shift":{"type":"lsl",
"value":12},"access":"r"}],regsAccessed {"read":["sp"],"written":["sp"]},regsRead [],re
gsWritten [],groups [],toJSON ={"address":"0x252c0ecb0","next":"0x4","size":4,"mnemoni
c":"sub","opStr":"sp, sp, #1, lsl #12","operands":[{"type":"reg","value":"sp","access":
"w"},{"type":"reg","value":"sp","access":"r"},{"type":"imm","value":"1","shift":{"type":
"lsl","value":12},"access":"r"}],"regsAccessed":{"read":["sp"],"written":["sp"]},"regsR
ead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c0ecb0 -> offset: hex=0x1c=28
```

```
    0x252c0ecb0 <+28>: sub sp, sp, #1, lsl #12
Instruction: address 0x252c0ecb4,toString()=sub sp, sp, #0xf00,next 0x4,size 4,mnemonic
sub,opStr sp, sp, #0xf00,operands [{"type":"reg","value":"sp","access":"w"},{"type":"re
g","value":"sp","access":"r"},{"type":"imm","value":"3840","access":"r"}],regsAccessed={
"read":["sp"],"written":["sp"]},regsRead=[],regsWritten=[],groups=[],toJSON()={"address"
:"0x252c0ecb4","next":"0x4","size":4,"mnemonic":"sub","opStr":"sp, sp, #0xf00","operand
s":[{"type":"reg","value":"sp","access":"w"},{"type":"reg","value":"sp","access":"r"},{
"type":"imm","value":"3840","access":"r"}],"regsAccessed":{"read":["sp"],"written":["sp"
]},"regsRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c0ecb4 -> offset: hex=0x20=32
    0x252c0ecb4 <+32>: sub sp, sp, #0xf00
Instruction: address 0x252c0ecb8,toString()=str x1, [sp, #0x88],next 0x4,size 4,mnemonic
=str,opStr x1, [sp, #0x88],operands [{"type":"reg","value":"x1","access":"r"},{"type":"
mem","value":{"base":"sp","disp":136},"access":"rw"}],regsAccessed={"read":["x1","sp"],
"written":[]},regsRead=[],regsWritten=[],groups=[],toJSON()={"address":"0x252c0ecb8","n
ext":"0x4","size":4,"mnemonic":"str","opStr":"x1, [sp, #0x88]","operands":[{"type":"reg"
,"value":"x1","access":"r"},{"type":"mem","value":{"base":"sp","disp":136},"access":"rw"
}],"regsAccessed":{"read":["x1","sp"],"written":[]},"regsRead":[],"regsWritten":[],"gro
ups":[]}
current: realAddr 0x252c0ecb8 -> offset: hex=0x24=36
    0x252c0ecb8 <+36>: str x1, [sp, #0x88]
Instruction: address 0x252c0ecbc,toString()=str x0, [sp, #0xa8],next 0x4,size 4,mnemonic
=str,opStr x0, [sp, #0xa8],operands [{"type":"reg","value":"x0","access":"r"},{"type":"
mem","value":{"base":"sp","disp":168},"access":"rw"}],regsAccessed={"read":["x0","sp"],
"written":[]},regsRead=[],regsWritten=[],groups=[],toJSON()={"address":"0x252c0ecbc","n
ext":"0x4","size":4,"mnemonic":"str","opStr":"x0, [sp, #0xa8]","operands":[{"type":"reg"
,"value":"x0","access":"r"},{"type":"mem","value":{"base":"sp","disp":168},"access":"rw"
}],"regsAccessed":{"read":["x0","sp"],"written":[]},"regsRead":[],"regsWritten":[],"gro
ups":[]}
current: realAddr 0x252c0ecbc -> offset: hex=0x28=40
    0x252c0ecbc <+40>: str x0, [sp, #0xa8]
Instruction: address 0x252c0ecc0,toString()=add x28, sp, #0x428,next 0x4,size 4,mnemonic
=add,opStr x28, sp, #0x428,operands [{"type":"reg","value":"x28","access":"w"},{"type":
"reg","value":"sp","access":"r"},{"type":"imm","value":"1064","access":"r"}],regsAccess
ed={"read":["sp"],"written":["x28"]},regsRead=[],regsWritten=[],groups=[],toJSON()={"ad
dress":"0x252c0ecc0","next":"0x4","size":4,"mnemonic":"add","opStr":"x28, sp, #0x428","
operands":[{"type":"reg","value":"x28","access":"w"},{"type":"reg","value":"sp","access"
:"r"},{"type":"imm","value":"1064","access":"r"}],"regsAccessed":{"read":["sp"],"writte
n":["x28"]},"regsRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c0ecc0 -> offset: hex=0x2c=44
    0x252c0ecc0 <+44>: add x28, sp, #0x428
Instruction: address 0x252c0ecc4,toString()=add x23, sp, #0x315,next 0x4,size 4,mnemonic
=add,opStr x23, sp, #0x315,operands [{"type":"reg","value":"x23","access":"w"},{"type":
"reg","value":"sp","access":"r"},{"type":"imm","value":"789","access":"r"}],regsAccessed
={"read":["sp"],"written":["x23"]},regsRead=[],regsWritten=[],groups=[],toJSON()={"addr
ess":"0x252c0ecc4","next":"0x4","size":4,"mnemonic":"add","opStr":"x23, sp, #0x315","op
erands":[{"type":"reg","value":"x23","access":"w"},{"type":"reg","value":"sp","access":
"r"},{"type":"imm","value":"789","access":"r"}],"regsAccessed":{"read":["sp"],"written":
["x23"]},"regsRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c0ecc4 -> offset: hex=0x30=48
    0x252c0ecc4 <+48>: add x23, sp, #0x315
Instruction: address 0x252c0ecc8,toString()=add x26, sp, #0x158,next 0x4,size 4,mnemonic
=add,opStr x26, sp, #0x158,operands [{"type":"reg","value":"x26","access":"w"},{"type":
"reg","value":"sp","access":"r"},{"type":"imm","value":"344","access":"r"}],regsAccessed
={"read":["sp"],"written":["x26"]},regsRead=[],regsWritten=[],groups=[],toJSON()={"addr
ess":"0x252c0ecc8","next":"0x4","size":4,"mnemonic":"add","opStr":"x26, sp, #0x158","op
```

erands":[{"type":"reg","value":"x26","access":"w"},{"type":"reg","value":"sp","access":"r"},{"type":"imm","value":"344","access":"r"}],"regsAccessed":{"read":["sp"],"written":["x26"]},"regsRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c0ecc8 -> offset: hex 0x34 52

...

    0x252c0edf0 <+348>: add x10, x8, x10
Instruction: address 0x252c0edf4,toString()=add w8, w9, #1,next 0x4,size 4,mnemonic add,opStr w8, w9, #1,operands [{"type":"reg","value":"w8","access":"w"},{"type":"reg","value":"w9","access":"r"},{"type":"imm","value":"1","access":"r"}],regsAccessed={"read":["w9"],"written":["w8"]},regsRead [],regsWritten=[],groups=[],toJSON()={"address":"0x252c0edf4","next":"0x4","size":4,"mnemonic":"add","opStr":"w8, w9, #1","operands":[{"type":"reg","value":"w8","access":"w"},{"type":"reg","value":"w9","access":"r"},{"type":"imm","value":"1","access":"r"}],"regsAccessed":{"read":["w9"],"written":["w8"]},"regsRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c0edf4 -> offset: hex 0x160 352
    0x252c0edf4 <+352>: add w8, w9, #1
Instruction: address 0x252c0edf8,toString()=br x10,next 0x4,size 4,mnemonic br,opStr x10,operands=[{"type":"reg","value":"x10","access":"r"}],regsAccessed={"read":["x10"],"written":[]},regsRead=[],regsWritten=[],groups=["jump"],toJSON()={"address":"0x252c0edf8","next":"0x4","size":4,"mnemonic":"br","opStr":"x10","operands":[{"type":"reg","value":"x10","access":"r"}],"regsAccessed":{"read":["x10"],"written":[]},"regsRead":[],"regsWritten":[],"groups":["jump"]}
current: realAddr 0x252c0edf8 -> offset: hex 0x164 356
    0x252c0edf8 <+356>: br x10
offset: 0x164 356
[0x164] contextStr={"pc":"0x252c0edf8","sp":"0x16f864a10","nzcv":536870912,"x0":"0x58eef7c8","x1":"0x16f866984","x2":"0x16f8669c8","x3":"0x1","x4":"0x100c71750","x5":"0x8","x6":"0x64","x7":"0x0","x8":"0x7","x9":"0x6","x10":"0x252c0ee00","x11":"0x6","x12":"0x2668d857","x13":"0x0","x14":"0x10082738d","x15":"0x2648f095a","x16":"0x252c4ea00","x17":"0xa711087f","x18":"0x0","x19":"0x16f8669c8","x20":"0x100827377","x21":"0x100834018","x22":"0x0","x23":"0x16f864d25","x24":"0x399258cb","x25":"0x0","x26":"0x16f864b68","x27":"0x176dc56e","x28":"0x16f864e38","fp":"0x16f866960","lr":"0x11074c00c","q0":{},"q1":{},"q2":{},"q3":{},"q4":{},"q5":{},"q6":{},"q7":{},"q8":{},"q9":{},"q10":{},"q11":{},"q12":{},"q13":{},"q14":{},"q15":{},"q16":{},"q17":{},"q18":{},"q19":{},"q20":{},"q21":{},"q22":{},"q23":{},"q24":{},"q25":{},"q26":{},"q27":{},"q28":{},"q29":{},"q30":{},"q31":{},"d0":6.0132879286022618e-154,"d1":1.9999992847442624,"d2":7.9499288951273454e-275,"d3":0,"d4":0,"d5":0,"d6":0,"d7":0,"d8":0,"d9":0,"d10":0,"d11":0,"d12":0,"d13":0,"d14":0,"d15":0,"d16":1.0855813867524649e+251,"d17":0.000002084565455e-308,"d18":3.0554698911305689e-152,"d19":0.000001322084152e-308,"d20":3.7696966457559071e-175,"d21":1.0004894955531618e+128,"d22":9.59442607756939l3e+225,"d23":-6.8154492514793903e-236,"d24":2.19309237443878554e+50,"d25":0,"d26":0,"d27":0,"d28":0,"d29":0,"d30":0,"d31":0,"s0":1.3563115067909484e-19,"s1":1.9999998807907104,"s2":3.8204714345426298e-37,"s3":0,"s4":0,"s5":0,"s6":0,"s7":0,"s8":0,"s9":0,"s10":0,"s11":0,"s12":0,"s13":0,"s14":0,"s15":0,"s16":1.5548730664783592e-21,"s17":1.3084408235578860e+36,"s18":-8.6088756618229034e-23,"s19":5.4087847452330076e-20,"s20":4.2151578028428229e+37,"s21":18362722504671230,"s22":3.372155249992253e+28,"s23":7.6784630118876258e-30,"s24":-3715189.5,"s25":0,"s26":0,"s27":0,"s28":0,"s29":0,"s30":0,"s31":0}
[0x164] x10=0x252c0ee00
+++ into iterator: startAddress=0x252c0ee00, isAppCode=true
Instruction: address 0x252c0ee00,toString()=mov w25, #0xddc9,next 0x4,size 4,mnemonic mov,opStr w25, #0xddc9,operands=[{"type":"reg","value":"w25","access":"w"},{"type":"imm","value":"56777","access":""}],regsAccessed={"read":[],"written":["w25"]},regsRead=[],regsWritten=[],groups=[],toJSON()={"address":"0x252c0ee00","next":"0x4","size":4,"mnemonic":"mov","opStr":"w25, #0xddc9","operands":[{"type":"reg","value":"w25","access":"w"},{

"type":"imm","value":"56777","access":""}],"regsAccessed":{"read":[],"written":["w25"]},
"regsRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c0ee00 -> offset: hex=0x16c=364
    0x252c0ee00 <+364>: mov w25, #0xddc9
Instruction: address 0x252c0ee04,toString()=movk w25, #0x6c8d, lsl #16,next 0x4,size 4,
mnemonic movk,opStr w25, #0x6c8d, lsl #16,operands=[{"type":"reg","value":"w25","access"
:"rw"},{"type":"imm","value":"27789","shift":{"type":"lsl","value":16},"access":"r"}],r
egsAccessed={"read":["w25"],"written":["w25"]},regsRead=[],regsWritten=[],groups=[],toJ
SON()={"address":"0x252c0ee04","next":"0x4","size":4,"mnemonic":"movk","opStr":"w25, #0
x6c8d, lsl #16","operands":[{"type":"reg","value":"w25","access":"rw"},{"type":"imm","v
alue":"27789","shift":{"type":"lsl","value":16},"access":"r"}],"regsAccessed":{"read":[
"w25"],"written":["w25"]},"regsRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c0ee04 -> offset: hex=0x170=368

...

    0x252c13cd8 <+20548>: add x8, x8, x9
Instruction: address 0x252c13cdc,toString()=br x8,next 0x4,size 4,mnemonic br,opStr x8,
operands=[{"type":"reg","value":"x8","access":"r"}],regsAccessed={"read":["x8"],"writte
n":[]},regsRead=[],regsWritten=[],groups=["jump"],toJSON()={"address":"0x252c13cdc","ne
xt":"0x4","size":4,"mnemonic":"br","opStr":"x8","operands":[{"type":"reg","value":"x8",
"access":"r"}],"regsAccessed":{"read":["x8"],"written":[]},"regsRead":[],"regsWritten":[
],"groups":["jump"]}
current: realAddr 0x252c13cdc -> offset: hex=0x5048=20552
    0x252c13cdc <+20552>: br x8
+++ into iterator: startAddress 0x252c13ce4, isAppCode=true
Instruction: address 0x252c13ce4,toString()=mov w22, #0xf6e,next 0x4,size 4,mnemonic mov
,opStr w22, #0xf6e,operands=[{"type":"reg","value":"w22","access":"w"},{"type":"imm","v
alue":"3950","access":""}],regsAccessed={"read":[],"written":["w22"]},regsRead=[],regsW
ritten=[],groups=[],toJSON()={"address":"0x252c13ce4","next":"0x4","size":4,"mnemonic":
"mov","opStr":"w22, #0xf6e","operands":[{"type":"reg","value":"w22","access":"w"},{"typ
e":"imm","value":"3950","access":""}],"regsAccessed":{"read":[],"written":["w22"]},"reg
sRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c13ce4 -> offset: hex=0x5050=20560
    0x252c13ce4 <+20560>: mov w22, #0xf6e
Instruction: address 0x252c13ce8,toString()=movk w22, #0x327e, lsl #16,next 0x4,size 4,
mnemonic movk,opStr w22, #0x327e, lsl #16,operands=[{"type":"reg","value":"w22","access"
:"rw"},{"type":"imm","value":"12926","shift":{"type":"lsl","value":16},"access":"r"}],r
egsAccessed={"read":["w22"],"written":["w22"]},regsRead=[],regsWritten=[],groups=[],toJ
SON()={"address":"0x252c13ce8","next":"0x4","size":4,"mnemonic":"movk","opStr":"w22, #0
x327e, lsl #16","operands":[{"type":"reg","value":"w22","access":"rw"},{"type":"imm","v
alue":"12926","shift":{"type":"lsl","value":16},"access":"r"}],"regsAccessed":{"read":[
"w22"],"written":["w22"]},"regsRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c13ce8 -> offset: hex=0x5054=20564

...

    0x252c13d70 <+20700>: sub w25, w8, #0x2a
Instruction: address 0x252c13d74,toString()=bl #0x252c42e9c,next 0x4,size 4,mnemonic bl,
opStr=0x252c42e9c,operands=[{"type":"imm","value":"9978523292","access":"r"}],regsAcce
ssed={"read":[],"written":["lr"]},regsRead=[],regsWritten=["lr"],groups=["call","jump",
"branch_relative"],toJSON()={"address":"0x252c13d74","next":"0x4","size":4,"mnemonic":"
bl","opStr":"#0x252c42e9c","operands":[{"type":"imm","value":"9978523292","access":"r"}]
,"regsAccessed":{"read":[],"written":["lr"]},"regsRead":[],"regsWritten":["lr"],"groups"
:["call","jump","branch_relative"]}
current: realAddr 0x252c13d74 -> offset: hex=0x50e0=20704

```
    0x252c13d74 <+20704>:  bl   #0x252c42e9c
+++ into iterator  startAddress 0x252c42e9c, isAppCode false
+++ into iterator  startAddress 0x22ac914c0, isAppCode false
+++ into iterator  startAddress 0x22ac914dc, isAppCode false
+++ into iterator  startAddress 0x22ac914f0, isAppCode false
+++ into iterator  startAddress 0x252c13d78, isAppCode true
Instruction: address 0x252c13d78,toString()=str x0, [sp, #0xc8],next 0x4,size 4,mnemonic
=str,opStr x0, [sp, #0xc8],operands [{"type":"reg","value":"x0","access":"r"},{"type":"
mem","value":{"base":"sp","disp":200},"access":"rw"}],regsAccessed={"read":["x0","sp"],
"written":[]},regsRead [],regsWritten [],groups=[],toJSON()={"address":"0x252c13d78","n
ext":"0x4","size":4,"mnemonic":"str","opStr":"x0, [sp, #0xc8]","operands":[{"type":"reg"
,"value":"x0","access":"r"},{"type":"mem","value":{"base":"sp","disp":200},"access":"rw"
}],"regsAccessed":{"read":["x0","sp"],"written":[]},"regsRead":[],"regsWritten":[],"gro
ups":[]}
current: realAddr 0x252c13d78 -> offset: hex=0x50e4=20708
    0x252c13d78 <+20708>:  str x0, [sp, #0xc8]
Instruction: address 0x252c13d7c,toString()=bl #0x252c42d88,next 0x4,size 4,mnemonic bl,
opStr=#0x252c42d88,operands [{"type":"imm","value":"9978523016","access":"r"}],regsAcce
ssed={"read":[],"written":["lr"]},regsRead [],regsWritten=["lr"],groups=["call","jump",
"branch_relative"],toJSON()={"address":"0x252c13d7c","next":"0x4","size":4,"mnemonic":"
bl","opStr":"#0x252c42d88","operands":[{"type":"imm","value":"9978523016","access":"r"}]
,"regsAccessed":{"read":[],"written":["lr"]},"regsRead":[],"regsWritten":["lr"],"groups"
:["call","jump","branch_relative"]}
current: realAddr 0x252c13d7c -> offset: hex=0x50e8=20712
    0x252c13d7c <+20712>:  bl   #0x252c42d88
+++ into iterator  startAddress 0x252c42d88, isAppCode false
+++ into iterator  startAddress 0x22ab7145c, isAppCode false
+++ into iterator  startAddress 0x22ab71478, isAppCode false
+++ into iterator  startAddress 0x22ab7147c, isAppCode false
+++ into iterator  startAddress 0x22ab71490, isAppCode false
+++ into iterator  startAddress 0x22ab714a0, isAppCode false
+++ into iterator  startAddress 0x22ab714cc, isAppCode false
+++ into iterator  startAddress 0x22ab714d0, isAppCode false
+++ into iterator  startAddress 0x22ab714f8, isAppCode false
+++ into iterator  startAddress 0x22ab7151c, isAppCode false
+++ into iterator  startAddress 0x252c13d80, isAppCode true
Instruction: address 0x252c13d80,toString()=mov w5, #0x87f,next 0x4,size 4,mnemonic mov,
opStr w5, #0x87f,operands [{"type":"reg","value":"w5","access":"w"},{"type":"imm","valu
e":"2175","access":""}],regsAccessed={"read":[],"written":["w5"]},regsRead [],regsWritt
en=[],groups=[],toJSON()={"address":"0x252c13d80","next":"0x4","size":4,"mnemonic":"mov"
,"opStr":"w5, #0x87f","operands":[{"type":"reg","value":"w5","access":"w"},{"type":"imm"
,"value":"2175","access":""}],"regsAccessed":{"read":[],"written":["w5"]},"regsRead":[],
"regsWritten":[],"groups":[]}
current: realAddr 0x252c13d80 -> offset: hex=0x50ec=20716
    0x252c13d80 <+20716>:  mov w5, #0x87f
Instruction: address 0x252c13d84,toString()=movk w5, #0xa711, lsl #16,next 0x4,size 4,m
nemonic movk,opStr w5, #0xa711, lsl #16,operands=[{"type":"reg","value":"w5","access":"
rw"},{"type":"imm","value":"42769","shift":{"type":"lsl","value":16},"access":"r"}],reg
sAccessed={"read":["w5"],"written":["w5"]},regsRead [],regsWritten=[],groups=[],toJSON(
)={"address":"0x252c13d84","next":"0x4","size":4,"mnemonic":"movk","opStr":"w5, #0xa711,
 lsl #16","operands":[{"type":"reg","value":"w5","access":"rw"},{"type":"imm","value":"
42769","shift":{"type":"lsl","value":16},"access":"r"}],"regsAccessed":{"read":["w5"],"
written":["w5"]},"regsRead":[],"regsWritten":[],"groups":[]}
current: realAddr 0x252c13d84 -> offset: hex=0x50f0=20720

...
```

libmetasec_ml.so的ms_bd_c_l_a

此处示例代码特点是：

已整理出Frida的Stalker的hook的通用的工具类函数

- FridaUtil.js
  - `stalkerHookUnnameNative`
    - 支持通过 `hookFuncMap` 传入，要hook的，多个对应的代码的点=位置

以及调用到的其他相关工具类函数：

- JsUtil.js
  - `isValidPointer`
  - `logStr`
  - `intToHexStr`
  - `isItemInList`
- FridaUtil.js
  - `printFunctionCallStack_addr`
  - `dumpMemory`
  - `ptrToCStr`
  - `ptrToUtf8Str`
- FridaAndroidUtil.js
  - `hookAfterLibLoaded`
  - `getJavaClassName`
  - `getJclassName`

`stalkerHookNative_ms_bd_c_l_a` 及相关代码

贴出此处示例的核心hook代码：

```
function hookNative_ms_bd_c_l_a(moduleBaseAddress){
  /*
    ==================== Trigged RegisterNatives [0x738cd109f8] ====================
    jniEnv=0x74504ec130
    clazz=0x85 -> jclassName=ms.bd.c.l
    methodsPtr=0x72d02cce60
    nMethods=0x1 -> methodNum=1
    -------------------------------- method [0] --------------------------------
    name: pos=0x72d02cce60 -> ptr=0x73a04e22d0 -> str=a
    signature: pos=0x72d02cce68 -> ptr=0x73d04fcb90 -> str=(IIJLjava/lang/String;Ljava/
lang/Object;)Ljava/lang/Object;
    fnPtr: pos=0x72d02cce70 -> ptr=0x727a656100 -> offset=0x104100
    Module: name=libmetasec_ml.so, base=0x727a552000, size=undefined, path=/data/app/~~
cJq-AFnKyZwZZqpRJY9hWA==/com.ss.android.ugc.aweme-1LCghzY2HKvNbI0WF3vccQ==/lib/arm64/li
bmetasec_ml.so

    sources/ms/bd/c/l.java
```

```
      public class l {
        public static native Object a(int i, int i2, long j, String str, Object obj);
  */
  var funcRelativeStartAddr = 0x104100
  console.log("funcRelativeStartAddr=" + funcRelativeStartAddr)

  const funcRealStartAddr = moduleBaseAddress.add(funcRelativeStartAddr)
  console.log("funcRealStartAddr=" + funcRealStartAddr)

  // var funcName = "ms.bd.c.l.a"
  var funcName = "ms_bd_c_l_a"
  Interceptor.attach(funcRealStartAddr, {
    onEnter: function(args) {
      JsUtil.logStr("Trigged " + funcName + " [" + funcRealStartAddr + "]")
      // FridaUtil.printFunctionCallStack_addr(this.context, funcName)

      var arg0 = args[0]
      var arg1 = args[1]
      var arg2 = args[2]
      var arg3 = args[3]
      var arg4 = args[4]
      var arg5 = args[5]
      var arg6 = args[6]

      var x0 = this.context.x0
      var x1 = this.context.x1
      var x2 = this.context.x2
      var x3 = this.context.x3
      var x4 = this.context.x4
      var x5 = this.context.x5
      var x6 = this.context.x6
      var sp = this.context.sp

      var realArg_i = arg3 // W3
      var realArg_i2 = arg2 // W2
      var realArg_j = arg4 // X4
      var realArg_str = arg5 // X5
      var realArg_obj = arg6 // X6

      var realArg_i_type = typeof realArg_i
      var realArg_i_int = parseInt(realArg_i)
      console.log("realArg_i_type=" + realArg_i_type + ", realArg_i_int=" + realArg_i_i
nt)

      let RI_REPORT_I_LIST = [
        196609, // 0x30001
        196610, // 0x30002
        196611, // 0x30003
      ]

      // if (JsUtil.isItemInList(realArg_i_int, RI_REPORT_I_LIST)){
        console.log("arg0=" + arg0 + ", arg1=" + arg1 + ", arg2=" + arg2 + ", arg3=" +
arg3 + ", arg4=" + arg4 + ", arg5=" + arg5 + ", arg6=" + arg6)

        console.log("realArg_i=" + realArg_i + ", realArg_i2=" + realArg_i2 + ", realAr
g_j=" + realArg_j + ", realArg_str=" + realArg_str + ", realArg_obj=" + realArg_obj)
```

```
        console.log("x0=" + x0 + ", x1=" + x1 + ", x2=" + x2 + ", x3=" + x3 + ", x4=" +
 x4 + ", x5=" + x5 + ", x6=" + x6 + ", sp=" + sp)

        // // var newBrX1Ptr_1 = ptr(sp - 0x58)
        // var newBrX1Ptr = sp.sub(0x58)
        // // console.log("newBrX1Ptr_1=" + newBrX1Ptr_1 + ", newBrX1Ptr=" + newBrX1Ptr)

        // console.log("newBrX1Ptr=" + newBrX1Ptr)
        // var newBrX1_nativePointer = newBrX1Ptr.readPointer()
        // // console.log("newBrX1_nativePointer=" + newBrX1_nativePointer)
        // var newBrX1_U64 = newBrX1Ptr.readU64()
        // // console.log("newBrX1_U64=" + newBrX1_U64)
        // console.log("newBrX1: nativePointer=" + newBrX1_nativePointer + ", U64=" + n
ewBrX1_U64)

        // var toDumpPtr = newBrX1Ptr
        // // var toDumpPtr = newBrX1Ptr - 0x10
        // // var toDumpPtr = ptr(newBrX1Ptr - 0x10)
        // console.log("toDumpPtr=" + toDumpPtr)
        // var byteLen = 64
        // var byteLen = 128
        // FridaUtil.dumpMemory(toDumpPtr, byteLen)

        if (JsUtil.isValidPointer(realArg_str)){
          var cStr = FridaUtil.ptrToCStr(realArg_str)
          console.log("cStr=" + cStr)
          // var utf8Str = FridaUtil.ptrToUtf8Str(realArg_str)
          // console.log("utf8Str=" + utf8Str)
          // console.log("cStr=" + cStr + ", utf8Str=" + utf8Str)
        }

        if (JsUtil.isValidPointer(realArg_obj)){
          var javaClassName = FridaAndroidUtil.getJavaClassName(realArg_obj)
          console.log("javaClassName=" + javaClassName)
          // var jclassName = FridaAndroidUtil.getJclassName(realArg_obj)
          // console.log("jclassName=" + jclassName)
          // console.log("jclassName=" + jclassName + ", javaClassName=" + javaClassNam
e)
        }
      // }

    },
    onLeave: function(retVal){
      console.log(funcName + " [0x104100]: retVal=" + retVal)
    }
  })
}


function stalkerHookNative_sub_10413C(moduleBaseAddress){
  var funcRelativeStartAddr = 0x10413C
  var functionSize = 0x14 // 0x104150 - 0x10413C = 0x14
  var argNum = 0
  let hookFuncMap = {
    0x08:
```

```
      function (context) {
        var contextStr = JSON.stringify(context)
        console.log("[0x08] contextStr=" + contextStr)
        var sp = context.sp
        console.log("sp=" + sp)
      },
    0x0C:
      function (context) {
        var contextStr = JSON.stringify(context)
        console.log("[0x0C] contextStr=" + contextStr)
        var x0 = context.x0
        console.log("x0=" + x0)
      },
  }

  FridaUtil.stalkerHookUnnameNative(moduleBaseAddress, funcRelativeStartAddr, functionS
ize, argNum, hookFuncMap)
}

function stalkerHookNative_loc_104190(moduleBaseAddress){
  var funcRelativeStartAddr = 0x104190
  var functionSize = 0x40 // @@@ Tracing basic block at 0x114190, block size = 0x40
  var argNum = 0
  let hookFuncMap = {
    0x18:
      function (context) {
        var contextStr = JSON.stringify(context)
        console.log("[0x18] contextStr=" + contextStr)
        var x23 = context.x23
        console.log("[0x18] x23=" + x23)
      },
    0x1C:
      function (context) {
        var contextStr = JSON.stringify(context)
        console.log("[0x1C] contextStr=" + contextStr)
        var x23 = context.x23
        console.log("[0x1C] x23=" + x23)
      },
    0x20:
      function (context) {
        var contextStr = JSON.stringify(context)
        console.log("[0x20] contextStr=" + contextStr)
        var x8 = context.x8
        console.log("[0x20] x8=" + x8)
      },
  }

  FridaUtil.stalkerHookUnnameNative(moduleBaseAddress, funcRelativeStartAddr, functionS
ize, argNum, hookFuncMap)
}

function stalkerHookNative_ms_bd_c_l_a(moduleBaseAddress){
  var funcRelativeStartAddr = 0x104100
  var functionSize = 0x38 // 0x104138 - 0x104100 = 0x38
  // var functionSize = 0x50 // including sub_10413C: 0x104150 - 0x104100 = 0x50
  var argNum = 5
```

```
  // let hookFuncMap = new Map()
  // hookFuncMap.set(
  //   0x30,
  //   function (context) {
  //     var contextStr = JSON.stringify(context)
  //     console.log("contextStr=" + contextStr)
  //     var x0 = context.x0
  //     var x0ToOffset = x0.sub(moduleBaseAddress)
  //     console.log("x0=" + x0 + "-> offset=" + x0ToOffset)
  //     var x1 = context.x1
  //     console.log("x1=" + x1)
  //     var x1ToOffset = x1.sub(moduleBaseAddress)
  //     console.log("x1=" + x1 + "-> offset=" + x1ToOffset)
  //   }
  // )
  let hookFuncMap = {
    // 0x2C:
    //   // function (context, curOffsetInt, moduleBaseAddr) {
    //   // function (context, extraDataDict) {
    //   function (context) {
    //     // var curOffsetInt = context.curOffsetInt
    //     // var curOffsetHexPtr = context.curOffsetHexPtr
    //     // var moduleBaseAddr = context.moduleBaseAddress
    //     // var curOffsetInt = context[curOffsetInt]
    //     // var curOffsetHexPtr = context[curOffsetHexPtr]
    //     // var moduleBaseAddr = context[moduleBaseAddress]
    //     // console.log("in 0x2C: curOffsetInt=" + curOffsetInt + "=" + curOffsetHexP
tr)

    //     // var contextStr = JSON.stringify(context)
    //     // console.log("[0x2C] contextStr=" + contextStr)
    //   },
    0x30:
      function (context) {
      // function (context, extraDataDict) {
      // function (context, curOffsetInt, moduleBaseAddr) {
        // console.log("extraDataDict=" + extraDataDict)
        // var curOffsetInt = context.curOffsetInt
        // var curOffsetHexPtr = context.curOffsetHexPtr
        // var moduleBaseAddr = context.moduleBaseAddress
        // var curOffsetInt = context[curOffsetInt]
        // var curOffsetHexPtr = context[curOffsetHexPtr]
        // var moduleBaseAddr = context[moduleBaseAddress]
        // console.log("called 0x30: curOffsetInt=" + curOffsetInt + ", moduleBaseAddr=
" + moduleBaseAddr)
        // // var curOffsetHexPtr = JsUtil.intToHexStr(curOffsetInt)
        // console.log("----- current offset: [" + curOffsetHexPtr + "] -----")

        // var contextStr = JSON.stringify(context)
        // console.log("[0x30] contextStr=" + contextStr)
        var x0 = context.x0
        var x0ToOffset = x0.sub(moduleBaseAddress)
        console.log("[0x30] x0=" + x0 + "-> offset=" + x0ToOffset)
      },
    0x38:
      // function (context, extraDataDict) {
```

```
      // function (context, curOffsetInt, moduleBaseAddress) {
      function (context) {
        // var curOffsetInt = context.curOffsetInt
        // var curOffsetHexPtr = context.curOffsetHexPtr
        // var moduleBaseAddr = context.moduleBaseAddress
        // var curOffsetInt = context[curOffsetInt]
        // var curOffsetHexPtr = context[curOffsetHexPtr]
        // var moduleBaseAddr = context[moduleBaseAddress]
        // console.log("in 0x38: curOffsetInt=" + curOffsetInt)
        // var curOffsetHexPtr = JsUtil.intToHexStr(curOffsetInt)
        // console.log("----- current offset: [" + curOffsetHexPtr + "] -----")

        // var contextStr = JSON.stringify(context)
        // console.log("[0x38] contextStr=" + contextStr)
        var x1 = context.x1
        var x1ToOffset = x1.sub(moduleBaseAddress)
        console.log("[0x38] x1=" + x1 + "-> offset=" + x1ToOffset)
      }
  }

  FridaUtil.stalkerHookUnnameNative(moduleBaseAddress, funcRelativeStartAddr, functionS
ize, argNum, hookFuncMap)

  // console.log("===== Frida Stalker hook for libmetasec_ml.so ms.bd.c.l a function ==
===");
  // /*
  //   ==================== Trigged RegisterNatives [0x738cd109f8] ====================
  //   jniEnv=0x74504ec130
  //   clazz=0x85 -> jclassName=ms.bd.c.l
  //   methodsPtr=0x72d02cce60
  //   nMethods=0x1 -> methodNum=1
  //   -------------------------------- method [0] --------------------------------
  //   name: pos=0x72d02cce60 -> ptr=0x73a04e22d0 -> str=a
  //   signature: pos=0x72d02cce68 -> ptr=0x73d04fcb90 -> str=(IIJLjava/lang/String;Lja
va/lang/Object;)Ljava/lang/Object;
  //   fnPtr: pos=0x72d02cce70 -> ptr=0x727a656100 -> offset=0x104100
  //   Module: name=libmetasec_ml.so, base=0x727a552000, size=undefined, path=/data/app
/~~cJq-AFnKyZwZZqpRJY9hWA==/com.ss.android.ugc.aweme-1LCghzY2HKvNbI0WF3vccQ==/lib/arm64
/libmetasec_ml.so

  //   sources/ms/bd/c/l.java
  //     public class l {
  //       public static native Object a(int i, int i2, long j, String str, Object obj);

  // */

  // var funcRelativeStartAddr = 0x104100
  // // console.log("funcRelativeStartAddr=" + funcRelativeStartAddr)
  // var funcRelativeStartAddrHexStr = JsUtil.intToHexStr(funcRelativeStartAddr)
  // // var functionSize = 0x38 // 0x104138 - 0x104100 = 0x38
  // var functionSize = 0x50 // including sub_10413C: 0x104150 - 0x104100 = 0x50
  // var functionSizeHexStr = JsUtil.intToHexStr(functionSize)
  // var funcRelativeEndAddr = funcRelativeStartAddr + functionSize
  // var funcRelativeEndAddrHexStr = JsUtil.intToHexStr(funcRelativeEndAddr)
  // console.log("function: relativeStartAddr=" + funcRelativeStartAddrHexStr + ", size
=" + functionSize + "=" + functionSizeHexStr + ", relativeEndAddr=" + funcRelativeEndAd
```

```
drHexStr)
  // // const moduleName = "libmetasec_ml.so"
  // // const moduleBaseAddress = Module.findBaseAddress(moduleName)
  // // console.log("moduleName=" + moduleName + ", moduleBaseAddress=" + moduleBaseAdd
ress)
  // console.log("moduleBaseAddress=" + moduleBaseAddress)
  // const funcRealStartAddr = moduleBaseAddress.add(funcRelativeStartAddr)
  // // var funcRealEndAddr = funcRealStartAddr + functionSize
  // const funcRealEndAddr = funcRealStartAddr.add(functionSize)
  // console.log("funcRealStartAddr=" + funcRealStartAddr + ", funcRealEndAddr=" + func
RealEndAddr)
  // var curTid = null
  // console.log("curTid=" + curTid)
  // Interceptor.attach(funcRealStartAddr, {
  //     onEnter: function(args) {
  //         var arg0 = args[0]
  //         var arg1 = args[1]
  //         var arg2 = args[2]
  //         var arg3 = args[3]
  //         var arg4 = args[4]
  //         console.log("----- arg0=" + arg0 + ", arg1=" + arg1 + ", arg2=" + arg2 + "
, arg3=" + arg3 + ", arg4=" + arg4)

  //         var curTid = Process.getCurrentThreadId()
  //         console.log("curTid=", curTid)
  //         Stalker.follow(curTid, {
  //             events: {
  //                 call: false, // CALL instructions: yes please
  //                 ret: true, // RET instructions
  //                 exec: false, // all instructions: not recommended as it's
  //                 block: false, // block executed: coarse execution trace
  //                 compile: false // block compiled: useful for coverage
  //             },
  //             // onReceive: Called with `events` containing a binary blob comprised
of one or more GumEvent structs. See `gumevent.h` for details about the format. Use `St
alker.parse()` to examine the data.
  //             onReceive(events) {
  //                 var parsedEvents = Stalker.parse(events)
  //                 // var parsedEventsStr = JSON.stringify(parsedEventsStr)
  //                 // console.log(">>> into onReceive: parsedEvents=" + parsedEvents
+ ", parsedEventsStr=" + parsedEventsStr);
  //                 console.log(">>> into onReceive: parsedEvents=" + parsedEvents);
  //             },

  //             // transform: (iterator: StalkerArm64Iterator) => {
  //             transform: function (iterator) {
  //                 // console.log("iterator=" + iterator)
  //                 var instruction = iterator.next()
  //                 const startAddress = instruction.address
  //                 // console.log("+++ into iterator: startAddress=" + startAddress)
  //                 // const isAppCode = startAddress.compare(funcRealStartAddr) >= 0
&& startAddress.compare(funcRealEndAddr) === -1
  //                 // const isAppCode = (startAddress.compare(funcRealStartAddr) >= 0
) && (startAddress.compare(funcRealEndAddr) < 0)
  //                 const gt_realStartAddr = startAddress.compare(funcRealStartAddr) >
= 0
```

```
  //                 const lt_realEndAddr = startAddress.compare(funcRealEndAddr) < 0
  //                 var isAppCode = gt_realStartAddr && lt_realEndAddr
  //                 console.log("+++ into iterator: startAddress=" + startAddress + ",
 isAppCode=" + isAppCode)

  //                 // // for debug
  //                 // isAppCode = true

  //                 // console.log("isAppCode=" + isAppCode + ", gt_realStartAddr=" +
gt_realStartAddr + ", lt_realEndAddr=" + lt_realEndAddr)
  //                 do {
  //                     if (isAppCode) {
  //                         // is origal function code = which we focus on

  //                         // console.log("instruction: address=" + instruction.addre
ss
  //                         //     + ",next=" + instruction.next()
  //                         //     + ",size=" + instruction.size
  //                         //     + ",mnemonic=" + instruction.mnemonic
  //                         //     + ",opStr=" + instruction.opStr
  //                         //     + ",operands=" + JSON.stringify(instruction.operand
s)
  //                         //     + ",regsAccessed=" + JSON.stringify(instruction.reg
sAccessed)
  //                         //     + ",regsRead=" + JSON.stringify(instruction.regsRea
d)
  //                         //     + ",regsWritten=" + JSON.stringify(instruction.regs
Written)
  //                         //     + ",groups=" + JSON.stringify(instruction.groups)
  //                         //     + ",toString()=" + instruction.toString()
  //                         //     + ",toJSON()=" + instruction.toJSON()
  //                         // );

  //                         var curRealAddr = instruction.address
  //                         // console.log("curRealAddr=" + curRealAddr)
  //                         // const isAppCode = curRealAddr.compare(funcRealStartAddr
) >= 0 && curRealAddr.compare(funcRealEndAddr) === -1
  //                         // console.log(curRealAddr + ": isAppCode=" + isAppCode)
  //                         var curOffsetHexPtr = curRealAddr.sub(funcRealStartAddr)
  //                         var curOffsetInt = curOffsetHexPtr.toInt32()
  //                         console.log("current: realAddr=" + curRealAddr + "-> offse
t: hex=" + curOffsetHexPtr + "=" + curOffsetInt)

  //                         // var instructionStr = instruction.mnemonic + " " + instr
uction.opStr
  //                         var instructionStr = instruction.toString()
  //                         // console.log("\t" + curRealAddr + ": " + instructionStr);

  //                         // console.log("\t" + curRealAddr + " <+" + curOffsetHexPt
r + ">: " + instructionStr)
  //                         console.log("\t" + curRealAddr + " <+" + curOffsetInt + ">
: " + instructionStr)
  //                         if (curOffsetInt == 0x30) {
  //                           console.log("offset: " + curOffsetHexPtr)
  //                           // iterator.putCallout(needDebug);
  //                           iterator.putCallout((context) => {
```

```
  //                                var contextStr = JSON.stringify(context)
  //                                console.log("contextStr=" + contextStr)
  //                                var x0 = context.x0
  //                                var x0ToOffset = x0.sub(moduleBaseAddress)
  //                                console.log("x0=" + x0 + "-> offset=" + x0ToOffset)
  //                                var x1 = context.x1
  //                                console.log("x1=" + x1)
  //                                var x1ToOffset = x1.sub(moduleBaseAddress)
  //                                console.log("x1=" + x1 + "-> offset=" + x1ToOffset)
  //                            })
  //                        } else if  (curOffsetInt == 0x38) {
  //                            console.log("offset: " + curOffsetHexPtr)
  //                            // iterator.putCallout(needDebug)
  //                            iterator.putCallout((context) => {
  //                                var contextStr = JSON.stringify(context)
  //                                console.log("contextStr=" + contextStr)
  //                                var x1 = context.x1
  //                                console.log("x1=" + x1)
  //                                var x1ToOffset = x1.sub(moduleBaseAddress)
  //                                console.log("x1ToOffset=" + x1ToOffset)
  //                            })
  //                        }
  //                    }
  //                    iterator.keep();
  //                } while ((instruction = iterator.next()) !== null)
  //            }
  //        });

  //        // function needDebug(context) {
  //        //     console.log("into needDebug")
  //        //     // console.log("into needDebug: context=" + context)
  //        //     // var contextStr = JSON.stringify(context, null, 2)
  //        //     // console.log("context=" + contextStr)
  //        //     // var x9Value1 = context.x9
  //        //     // var x9Value2 = context["x9"]
  //        //     // console.log("x9Value1=" + x9Value1 + ", x9Value2=" + x9Value2)
  //        // }
  //    },
  //    onLeave: function(retval) {
  //        console.log("retval=" + retval)
  //        if (curTid != null) {
  //            Stalker.unfollow(curTid);
  //            console.log("Stalker.unfollow curTid=", curTid)
  //        }
  //    }
  // })
}


function afterLibLoaded_libmetasec_ml(libraryName) {
  const moduleBaseAddress = Module.findBaseAddress(libraryName)
  console.log("libraryName=" + libraryName + " -> moduleBaseAddress=" + moduleBaseAddre
ss)

  // hookNative_ms_bd_c_l_a(moduleBaseAddress)
  // stalkerHookNative_ms_bd_c_l_a(moduleBaseAddress)
```

```
  // stalkerHookNative_sub_10413C(moduleBaseAddress)
  stalkerHookNative_loc_104190(moduleBaseAddress)
}

function hookNative_libmetasec_ml(){
  FridaAndroidUtil.hookAfterLibLoaded("libmetasec_ml.so", afterLibLoaded_libmetasec_ml)
}

function hookDouyin_Native(){
  hookNative_libmetasec_ml()
}

function hookDouyin() {
  // hookDouyin_Java()
  hookDouyin_Native()
}

function hookAndroid() {
  if( Java.available){
    console.error("Java is not available")
    return
  }

  console.log("Java is available")
  console.log("Java.androidVersion=" + Java.androidVersion)

  Java.perform(function () {
    hookInit()

    hookDouyin()

    console.log("-------------------- Begin Hook --------------------")
  })

}

setImmediate(hookAndroid)
```

调试日志

某次的调试输出的日志：

```
➜  frida frida -U -f com.ss.android.ugc.aweme -l hook_douyin.js
     ____
    / _  |   Frida 16.4.8 - A world class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
```

```
    . . . .   Connected to M2101K9C (id=9C181A8D3C3F3B)
Spawning `com.ss.android.ugc.aweme`...
FridaAndroidUtil.cacheDictEnvClazz [object Object]
FridaAndroidUtil.curThrowableCls <class: java.lang.Throwable>
FridaAndroidUtil.JavaArray <class: java.lang.reflect.Array>
FridaAndroidUtil.JavaArrays <class: java.util.Arrays>
FridaAndroidUtil.JavaArrayList <class: java.util.ArrayList>
FridaAndroidUtil.JavaByteArr <class: [B>
FridaAndroidUtil.JavaObjArr <class: [Ljava.lang.Object;>
JNIEnvIdx.CallObjectMethod 34
Java is available
Java.androidVersion 13
Spawned `com.ss.android.ugc.aweme`. Resuming main thread
[M2101K9C::com.ss.android.ugc.aweme::] -> libName=libmetasec_ml.so
libraryName libmetasec_ml.so, callback_afterLibLoaded function afterLibLoaded_libmetase
c_ml libraryName) {
  const moduleBaseAddress = Module.findBaseAddress(libraryName)
  console.log("libraryName=" + libraryName + " -> moduleBaseAddress=" + moduleBaseAddre
ss)

  // hook_JNI_OnLoad libraryName

  // FridaAndroidUtil.printModuleInfo("libmetasec_ml.so")

  // hookNative__JNIEnv_CallStaticObjectMethod(moduleBaseAddress)
  // hookNative__JNIEnv_NewObject(moduleBaseAddress)
  // hookNative_getJNIEnv(moduleBaseAddress)
  // hookNative__JNIEnv_CallObjectMethod(moduleBaseAddress)

  // hookNative_getSecDeviceToken(moduleBaseAddress)
  // hookNative_doHttpReqSignByUrl(moduleBaseAddress)

  // hookNative_ms_bd_c_l_a(moduleBaseAddress)
  // stalkerHookNative_ms_bd_c_l_a(moduleBaseAddress)
  // stalkerHookNative_sub_10413C(moduleBaseAddress)
  stalkerHookNative_loc_104190(moduleBaseAddress)
}
android_dlopen_ext 0x7a9cb230b8
-------------------- Begin Hook --------------------
+++ Loaded lib libmetasec_ml.so
libraryName libmetasec_ml.so -> moduleBaseAddress=0x7708060000
Frida Stalker hook: module: baseAddress=0x7708060000
function: relativeStartAddr=0x104190, size=64=0x40, relativeEndAddr=0x1041d0
funcRealStartAddr=0x7708164190, funcRealEndAddr=0x77081641d0
curTid=null
=========== Trigged addr: relative:[0x104190] - real:[0x7708164190] ===========
curTid=20389
+++ into iterator: startAddress=0x77d7300360, isAppCode=false
+++ into iterator: startAddress=0x77d7300370, isAppCode=false
+++ into iterator: startAddress=0x77d7300380, isAppCode=false
...
+++ into iterator: startAddress=0x77d72c484c, isAppCode=false
+++ into iterator: startAddress=0x7a9cdab0b4, isAppCode=false
+++ into iterator: startAddress=0x7a9cd03530, isAppCode=false
+++ into iterator: startAddress=0x77081641a0, isAppCode=true
current: realAddr=0x77081641a0 -> offset: hex=0x10=16
```

```
    0x77081641a0 <+16>:  stp x29, x30, [sp, #0x40]
current: realAddr 0x77081641a4 -> offset: hex 0x14 20
    0x77081641a4 <+20>:  add x29, sp, #0x40
current: realAddr 0x77081641a8 -> offset: hex 0x18 24
    0x77081641a8 <+24>:  mrs x23, tpidr_el0
offset: 0x18 24
current: realAddr 0x77081641ac -> offset: hex 0x1c 28
    0x77081641ac <+28>:  ldr x8, [x23, #0x28]
offset: 0x1c 28
current: realAddr 0x77081641b0 -> offset: hex 0x20 32
    0x77081641b0 <+32>:  mov x19, x4
offset: 0x20 32
current: realAddr 0x77081641b4 -> offset: hex 0x24 36
    0x77081641b4 <+36>:  mov x20, x3
current: realAddr 0x77081641b8 -> offset: hex 0x28 40
    0x77081641b8 <+40>:  mov x21, x2
current: realAddr 0x77081641bc -> offset: hex 0x2c 44
    0x77081641bc <+44>:  mov w22, w1
current: realAddr 0x77081641c0 -> offset: hex 0x30 48
    0x77081641c0 <+48>:  str x8, [sp, #8]
current: realAddr 0x77081641c4 -> offset: hex 0x34 52
    0x77081641c4 <+52>:  str w0, [sp, #4]
current: realAddr 0x77081641c8 -> offset: hex 0x38 56
    0x77081641c8 <+56>:  adr x9, #0x77081641c8
current: realAddr 0x77081641cc -> offset: hex 0x3c 60
    0x77081641cc <+60>:  b #0x77081641d4
[0x18] contextStr {"pc":"0x77081641a8","sp":"0x773cc9abf0","nzcv":2684354560,"x0":"0x10
00001","x1":"0x0","x2":"0x0","x3":"0x773cc9ad08","x4":"0x773cc9ad0c","x5":"0x773cc9ad08"
,"x6":"0x773cc9ad0c","x7":"0x0","x8":"0x8d4e175111831754","x9":"0x8d4e175111831754","x1
0":"0x30","x11":"0x7","x12":"0x7c33","x13":"0xffffffffffffffff","x14":"0x775dc83dcc","x
15":"0xf339","x16":"0x77081641a0","x17":"0x775e01f23c","x18":"0x773c532000","x19":"0x79
60801bb0","x20":"0x0","x21":"0x0","x22":"0x78b07cadd0","x23":"0x1","x24":"0x1","x25":"0
x3c","x26":"0x6e","x27":"0x52","x28":"0x135c5d40","fp":"0x773cc9ac30","lr":"0x7a9cd0350
c","q0":{},"q1":{},"q2":{},"q3":{},"q4":{},"q5":{},"q6":{},"q7":{},"q8":{},"q9":{},"q10"
:{},"q11":{},"q12":{},"q13":{},"q14":{},"q15":{},"q16":{},"q17":{},"q18":{},"q19":{},"q
20":{},"q21":{},"q22":{},"q23":{},"q24":{},"q25":{},"q26":{},"q27":{},"q28":{},"q29":{},
"q30":{},"q31":{},"d0":5.3304815262372 08e+228,"d1":1.6766152451136167e+243,"d2":0,"d3":-
8.348743279794574e-210,"d4":0,"d5":4.003911019303815,"d6":0,"d7":3.1152901395860827e-307
,"d8":1.697596633 17e-313,"d9":0,"d10":0,"d11":0,"d12":0,"d13":0,"d14":0,"d15":0,"d16":-
16.04693223164578,"d17":32.00198376226035,"d18":5.30498948e-315,"d19":1.95354366e-314,"
d20":1.3467897878976211e+116,"d21":-4.197101104091689e-187,"d22":2.3236709860767083e+1
30,"d23":1.8121650802628091e+267,"d24":1.9535436624e-314,"d25":1.953543663e-314,"d26":1
.9535436634e-314,"d27":1.953543664e-314,"d28":1.9535436644e-314,"d29":1.953543665e-314,
"d30":1.9535436653e-314,"d31":8.900981273429906e-307,"s0":2849164481055948800 00,"s1":1.
6926149821424084e+22,"s2":0,"s3":-1.8800355907811577e+23,"s4":0,"s5":2.250244379043579,
"s6":0,"s7":7.255137319835417e-39,"s8":1.1210387714598537e-44,"s9":0,"s10":0,"s11":0,"s
12":0,"s13":0,"s14":0,"s15":0,"s16":-2.7507331371307373,"s17":2.369642786801667e-38,"s1
8":2.000000238418579,"s19":-4.1950209245322364e+26,"s20":56.58197021484375,"s21":9.2733
93360036293e+31,"s22":-4.102617163036699e-37,"s23":-1.6180783810626933e+28,"s24":-4.195
022769206644e+26,"s25":-4.1950231381415 25e+26,"s26":-4.195023507076407e+26,"s27":-4.195
023876011288e+26,"s28":-4.1950242449461696e+26,"s29":-4.195024613881051e+26,"s30":-4.19
50249828159326e+26,"s31":9.183685541750161e-39}
[0x18] x23 0x1
[0x1C] contextStr {"pc":"0x77081641ac","sp":"0x773cc9abf0","nzcv":2684354560,"x0":"0x10
00001","x1":"0x0","x2":"0x0","x3":"0x773cc9ad08","x4":"0x773cc9ad0c","x5":"0x773cc9ad08"
,"x6":"0x773cc9ad0c","x7":"0x0","x8":"0x8d4e175111831754","x9":"0x8d4e175111831754","x1
```

0":"0x30","x11":"0x7","x12":"0x7c33","x13":"0xffffffffffffffff","x14":"0x775dc83dcc","x15":"0xf339","x16":"0x77081641a0","x17":"0x775e01f23c","x18":"0x773c532000","x19":"0x7960801bb0","x20":"0x0","x21":"0x0","x22":"0x78b07cadd0","x23":"0x773cc9c000","x24":"0x1","x25":"0x3c","x26":"0x6e","x27":"0x52","x28":"0x135c5d40","fp":"0x773cc9ac30","lr":"0x7a9cd0350c","q0":{},"q1":{},"q2":{},"q3":{},"q4":{},"q5":{},"q6":{},"q7":{},"q8":{},"q9":{},"q10":{},"q11":{},"q12":{},"q13":{},"q14":{},"q15":{},"q16":{},"q17":{},"q18":{},"q19":{},"q20":{},"q21":{},"q22":{},"q23":{},"q24":{},"q25":{},"q26":{},"q27":{},"q28":{},"q29":{},"q30":{},"q31":{},"d0":5.330481526237208e+228,"d1":1.6766152451136167e+243,"d2":0,"d3":8.348743279794574e-210,"d4":0,"d5":4.003911019303815,"d6":0,"d7":3.1152901395860827e-307,"d8":1.69759663317e-313,"d9":0,"d10":0,"d11":0,"d12":0,"d13":0,"d14":0,"d15":0,"d16":16.04693223164578,"d17":32.00198376226035,"d18":5.30498948e-315,"d19":1.95354366e-314,"d20":1.3467897878976211e+116,"d21":-4.197101104091689e-187,"d22":2.3236709860767083e+130,"d23":1.8121650802628091e+267,"d24":1.9535436624e-314,"d25":1.953543663e-314,"d26":1.95354366634e-314,"d27":1.953543664e-314,"d28":1.95354366644e-314,"d29":1.9535436665e-314,"d30":1.95354366653e-314,"d31":8.900981273429906e-307,"s0":2849164481055948800000,"s1":1.6926149821424084e+22,"s2":0,"s3":-1.8800355907811577e+23,"s4":0,"s5":2.250244379043579,"s6":0,"s7":7.255137319835417e-39,"s8":1.1210387714598537e-44,"s9":0,"s10":0,"s11":0,"s12":0,"s13":0,"s14":0,"s15":0,"s16":-2.7507331371307373,"s17":2.3696427868016 67e-38,"s18":2.000000238418579,"s19":-4.1950209245322364e+26,"s20":56.58197021484375,"s21":9.2733933600362 93e+31,"s22":-4.102617163036699e-37,"s23":-1.6180783810626933e+28,"s24":-4.195022769206644e+26,"s25":-4.195023138141525e+26,"s26":-4.195023507076407e+26,"s27":-4.195023876011288e+26,"s28":-4.1950242449461696e+26,"s29":-4.195024613881051e+26,"s30":-4.1950249828159326e+26,"s31":9.183685541750161e-39}

[0x1C] **x23** 0x773cc9c000

[0x20] contextStr={"pc":"0x77081641b0","sp":"0x773cc9abf0","nzcv":2684354560,"x0":"0x1000001","x1":"0x0","x2":"0x0","x3":"0x773cc9ad08","x4":"0x773cc9ad0c","x5":"0x773cc9ad08","x6":"0x773cc9ad0c","x7":"0x0","x8":"0x8d4e175111831754","x9":"0x8d4e175111831754","x10":"0x30","x11":"0x7","x12":"0x7c33","x13":"0xffffffffffffffff","x14":"0x775dc83dcc","x15":"0xf339","x16":"0x77081641a0","x17":"0x775e01f23c","x18":"0x773c532000","x19":"0x7960801bb0","x20":"0x0","x21":"0x0","x22":"0x78b07cadd0","x23":"0x773cc9c000","x24":"0x1","x25":"0x3c","x26":"0x6e","x27":"0x52","x28":"0x135c5d40","fp":"0x773cc9ac30","lr":"0x7a9cd0350c","q0":{},"q1":{},"q2":{},"q3":{},"q4":{},"q5":{},"q6":{},"q7":{},"q8":{},"q9":{},"q10":{},"q11":{},"q12":{},"q13":{},"q14":{},"q15":{},"q16":{},"q17":{},"q18":{},"q19":{},"q20":{},"q21":{},"q22":{},"q23":{},"q24":{},"q25":{},"q26":{},"q27":{},"q28":{},"q29":{},"q30":{},"q31":{},"d0":5.330481526237208e+228,"d1":1.6766152451136167e+243,"d2":0,"d3":8.348743279794574e-210,"d4":0,"d5":4.003911019303815,"d6":0,"d7":3.1152901395860827e-307,"d8":1.69759663317e-313,"d9":0,"d10":0,"d11":0,"d12":0,"d13":0,"d14":0,"d15":0,"d16":16.04693223164578,"d17":32.00198376226035,"d18":5.30498948e-315,"d19":1.95354366e-314,"d20":1.3467897878976211e+116,"d21":-4.197101104091689e-187,"d22":2.3236709860767083e+130,"d23":1.8121650802628091e+267,"d24":1.9535436624e-314,"d25":1.953543663e-314,"d26":1.95354366634e-314,"d27":1.953543664e-314,"d28":1.95354366644e-314,"d29":1.9535436665e-314,"d30":1.95354366653e-314,"d31":8.900981273429906e-307,"s0":2849164481055948800000,"s1":1.6926149821424084e+22,"s2":0,"s3":-1.8800355907811577e+23,"s4":0,"s5":2.250244379043579,"s6":0,"s7":7.255137319835417e-39,"s8":1.1210387714598537e-44,"s9":0,"s10":0,"s11":0,"s12":0,"s13":0,"s14":0,"s15":0,"s16":-2.7507331371307373,"s17":2.3696427868016 67e-38,"s18":2.000000238418579,"s19":-4.1950209245322364e+26,"s20":56.58197021484375,"s21":9.2733933600362 93e+31,"s22":-4.102617163036699e-37,"s23":-1.6180783810626933e+28,"s24":-4.195022769206644e+26,"s25":-4.195023138141525e+26,"s26":-4.195023507076407e+26,"s27":-4.195023876011288e+26,"s28":-4.1950242449461696e+26,"s29":-4.195024613881051e+26,"s30":-4.1950249828159326e+26,"s31":9.183685541750161e-39}

[0x20] **x8** 0x8d4e175111831754

+++ into iterator startAddress 0x77081641d4, isAppCode=false

...

+++ into iterator startAddress 0x7708164b10, isAppCode=false

>>> into onReceive parsedEvents ret,0x77d73f17ec,0x77d72fe844,0,ret,0x77d72ffd0c,0x77d72c478c, 1,ret,0x77d72c486c,0x7a9cdab0b4, 2,ret,0x7a9cdab140,0x7a9cd0350c, 3,ret,0x7708

```
16421c,0x77081641fc,-3,ret,0x7708164280,0x7708164264,-3,ret,0x7708164280,0x770816429c,-3
,ret,0x770816755c,0x770816857c,-1,ret,0x77080a29cc,0x7708168584,-1,ret,0x770816755c,0x7
70816857c,-1,ret,0x77080a29cc,0x7708168584,-1,ret,0x770816755c,0x770816857c,-1,ret,0x77
080a29cc,0x7708168584,-1,ret,0x770816755c,0x770816857c,-1,ret,0x77080a29cc,0x7708168584,
-1,ret,0x77081685b8,0x770816798c,-2,ret,0x7708167dcc,0x770816799c,-2,ret,0x7708168654,0
x77081679b0,-2,ret,0x77081685e4,0x77081679d4,-2,ret,0x77080a29cc,0x77081679e0,-2,ret,0x
77081685e4,0x77081679f4,-2,ret,0x7708167a18,0x77081643f8,-3,ret,0x7a9084eee4,0x7708164a
4c,-1,ret,0x7a9084dfc0,0x7708164a54,-1,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,
0x7a907da374,3,ret,0x7a90855ff4,0x7a907d8a3c,4,ret,0x7a907e4f5c,0x7a907da45c,3,ret,0x7a
907da514,0x7a907dab70,2,ret,0x7a907dab84,0x7a907d5134,1,ret,0x7a907d514c,0x77081ca484,0,
ret,0x77081ca4a0,0x7708164a6c,-1,ret,0x77ee47dc04,0x77ee64bc90,2,ret,0x7a90855d34,0x7a9
07d89a8,8,ret,0x7a907d89b4,0x7a907da330,7,ret,0x7a90855ff4,0x7a907d8a3c,8,ret,0x7a907d8
a4c,0x7a907da45c,7,ret,0x7a907da514,0x7a907dab70,6,ret,0x7a907dab84,0x7a907d5134,5,ret,
0x7a907d514c,0x7a80a5abc8,4,ret,0x7a80a5abe4,0x77ee64c050,3,ret,0x77ee64c108,0x77ee64bc
dc,2,ret,0x77ee47dcb4,0x77ee64bd34,2,ret,0x77ee653610,0x77ee64be34,2,ret,0x7a90855df4,0
x7a907dec6c,4,ret,0x7a90855d34,0x7a907d89a8,5,ret,0x7a907d89b4,0x7a907decb0,4,ret,0x7a9
0855ff4,0x7a907d8a3c,4,ret,0x7a907e4f5c,0x7a907da9c4,3,ret,0x7a907da9d4,0x77ee64be88,2,
ret,0x77ee64beb8,0x77080acbd0,1,ret,0x77080acbf0,0x7708164b10,0
+++ into iterator startAddress 0x7708164b98, isAppCode=false
...
+++ into iterator startAddress 0x7a9084e288, isAppCode=false
[M2101K9C com.ss.android.ugc.aweme ]->
+++ into iterator startAddress 0x7a9084e2fc, isAppCode=false
[M2101K9C com.ss.android.ugc.aweme ]-> +++ into iterator startAddress 0x7a9084e308, i
sAppCode=false
+++ into iterator startAddress 0x7a9084e4a0, isAppCode=false
...
+++ into iterator startAddress 0x77ee69c190, isAppCode=false
>>> into onReceive parsedEvents ret,0x7708164bb8,0x7708164a74,-1,ret,0x7a9084e020,0x77
08164a80,-1,ret,0x7708164ad4,0x77080ac5a0,-2,ret,0x7a9084dfd8,0x7708164be0,-1,ret,0x77e
e47dc04,0x77ee64bc90,1,ret,0x7a90855d34,0x7a907d89a8,7,ret,0x7a907d89b4,0x7a907da330,6,
ret,0x7a90855ff4,0x7a907d8a3c,7,ret,0x7a907e4f5c,0x7a907da45c,6,ret,0x7a907da514,0x7a90
7dab70,5,ret,0x7a907dab84,0x7a907d5134,4,ret,0x7a907d514c,0x7a80a5abc8,3,ret,0x7a80a5ab
e4,0x77ee64c050,2,ret,0x77ee64c108,0x77ee64bcdc,1,ret,0x77ee47dcb4,0x77ee64bd34,1,ret,0
x77ee653610,0x77ee64be34,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8
,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7
a907da9c4,2,ret,0x7a907da9d4,0x77ee64be88,1,ret,0x77ee64beb8,0x77080acbd0,0,ret,0x77080
acbf0,0x7708164bec,-1,ret,0x7708164bf8,0x77080ac5a8,-2,ret,0x770809fd08,0x77080ac5b0,-2,
ret,0x77080ac5f4,0x77080ac5d4,-2,ret,0x770809fd08,0x77080ac628,-2,ret,0x7708164948,0x77
08164928,-1,ret,0x77081649b0,0x7708164994,-1,ret,0x77081649b0,0x77081649cc,-1,ret,0x770
8164a10,0x77080ac63c,-2,ret,0x770816369c,0x77080ac65c,-2,ret,0x7a90855cb4,0x7a9084e598,3
,ret,0x7a9084e534,0x7a9084e318,2,ret,0x7a9084e284,0x7708160d8c,1,ret,0x7708160db4,0x770
80a0c98,0,ret,0x7a90855d34,0x7a907d89a8,5,ret,0x7a907d89b4,0x7a907da330,4,ret,0x7a90855
ff4,0x7a907d8a3c,5,ret,0x7a907d8a4c,0x7a907da45c,4,ret,0x7a907da514,0x7a907dab70,3,ret,
0x7a907dab84,0x7a907d5134,2,ret,0x7a907d514c,0x77081ca484,1,ret,0x77081ca4a0,0x77080a0c
a4,0,ret,0x7a90855ed4,0x7a9084e854,1,ret,0x7a9084e888,0x77080a0cb8,0,ret,0x77080a0cd8,0
x7708165ba8,-1,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907da330,3,ret,0x7a
90855ff4,0x7a907d8a3c,4,ret,0x7a907d8a4c,0x7a907da45c,3,ret,0x7a907da514,0x7a907dab70,2,
ret,0x7a907dab84,0x7a907d5134,1,ret,0x7a907d514c,0x77081ca484,0,ret,0x77081ca4a0,0x7708
165bbc,-1,ret,0x770809fd08,0x770815fb6c,0,ret,0x770815fbb0,0x770815fb90,0,ret,0x770809f
d08,0x770815fbe4,0,ret,0x7708164948,0x7708164928,1,ret,0x77081649b0,0x7708164994,1,ret,
0x77081649b0,0x77081649cc,1,ret,0x7708164a10,0x770815fbf8,0,ret,0x770815fde4,0x7708165b
d8,-1,ret,0x7a907e2310,0x77ee69b978,0,ret,0x77ee47dc04,0x77ee69bcdc,0,ret,0x77ee6b7d1c,
0x77ee6b6618,1,ret,0x77ee6b6adc,0x77ee69bf40,0,ret,0x77ee5c880c,0x77ee4c3f70,1,ret,0x77
ee4c3fa8,0x77ee69bf50,0
+++ into iterator startAddress 0x77ee853518, isAppCode=false
```

```
...
+++ into iterator  startAddress 0x77ee858198, isAppCode=false
>>> into onReceive  parsedEvents ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77e
e85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d8
9b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,
0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee
90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0
x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775
995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,
ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee
69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855d
f4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0
x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590
,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7
a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa24
9f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret
,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee853
5d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,
0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a
907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,
ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee
8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315
a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0
x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0
,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x7
7ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a908
55d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret
,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47e
db0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,
0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77
5995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,
ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee
69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855d
f4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0
x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590
,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7
a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa24
9f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret
,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee853
5d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,
0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a
907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,
ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee
8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315
a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0
x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0
,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x7
7ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a908
55d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret
,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47e
db0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,
0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7a
a249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,
ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee
8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1
b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0
```

x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4
,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x7
7ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07
315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret
,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f
2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,
0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a
90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,
ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee
47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e2
70,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0
x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248
,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x7
7ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee4
7e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret
,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855d34,0x7a907d89a8,6,ret,0x7a907d89b4,0x7a907dc
50c,5,ret,0x7a907dc518,0x7a907dcee4,4,ret,0x7a90855ff4,0x7a907d8a3c,5,ret,0x7a907d8a4c,
0x7a907dcf00,4,ret,0x7a90855d34,0x7a907d89a8,5,ret,0x7a907d89b4,0x7a907dcf30,4,ret,0x7a
907dcfe8,0x7a907ded74,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,
ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee
47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e2
70,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0
x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4
,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x7
7ee69c198,0,ret,0x7a907fa42c,0x7a88f5e248,6,ret,0x7a88f5e270,0x77f0731568,5,ret,0x77f07
31580,0x7aa249f1d4,4,ret,0x7aa249f1dc,0x77ee47e214,3,ret,0x77ee47e28c,0x77ee47dfc0,2,ret
,0x77ee47e35c,0x77ee47dfcc,2,ret,0x7a907fa42c,0x7a88f5e248,5,ret,0x7a88f5e270,0x77f0731
5c4,4,ret,0x77f07315d0,0x7aa249f2d0,3,ret,0x7aa249f2d4,0x77ee47e130,2,ret,0x77ee47dc04,
0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a
90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,
ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee
47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e2
70,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0
x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4
,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x7
7ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a908
55df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret
,0x7aa3c2c440,0x7a907e4120,5,ret,0x7a907e412c,0x7a907d8a98,4,ret,0x7a907d8ab0,0x7a907de
d64,3,ret,0x7a90855d34,0x7a907d89a8,6,ret,0x7a907d89b4,0x7a907dc50c,5,ret,0x7a907dc518,
0x7a907dcee4,4,ret,0x7a90855ff4,0x7a907d8a3c,5,ret,0x7a907d8a4c,0x7a907dcf00,4,ret,0x7a
90855d34,0x7a907d89a8,5,ret,0x7a907d89b4,0x7a907dcf30,4,ret,0x7a907dcfe8,0x7a907ded74,3,
ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee
853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed
44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0
x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4
,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7
aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee4
7dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret
,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8
a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,
0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a
907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,
ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee
8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315
d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0
x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c

```
,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7
a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee4
7dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret
,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f
254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,
0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77
f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,
ret,0x7a907fa42c,0x7a88f5e248,6,ret,0x7a88f5e270,0x77f0731568,5,ret,0x77f0731580,0x7aa2
49f1d4,4,ret,0x7aa249f1dc,0x77ee47e214,3,ret,0x77ee47e28c,0x77ee47dfc0,2,ret,0x77ee47e3
5c,0x77ee47dfcc,2,ret,0x7a907fa42c,0x7a88f5e248,5,ret,0x7a88f5e270,0x77f07315c4,4,ret,0
x77f07315d0,0x7aa249f2d0,3,ret,0x7aa249f2d4,0x77ee47e130,2,ret,0x77ee47dc04,0x77ee47df90
,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7
a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907
d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret
,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e
248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,
0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a
88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,
ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee
85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89
b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0
x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90
,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x7
7f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee4
1d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret
,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee853
5d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,
0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a
907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,
ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee
47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e2
70,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0
x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248
,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x7
7ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee4
7e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret
,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da
9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,
0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a
88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,
ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88
f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2
d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x7a907fa42c,0x7a88f5e248,6,ret,0
x7a88f5e270,0x77f0731568,5,ret,0x77f0731580,0x7aa249f1d4,4,ret,0x7aa249f1dc,0x77ee47e214
,3,ret,0x77ee47e28c,0x77ee47dfc0,2,ret,0x77ee47e35c,0x77ee47dfcc,2,ret,0x7a907fa42c,0x7
a88f5e248,5,ret,0x7a88f5e270,0x77f07315c4,4,ret,0x77f07315d0,0x7aa249f2d0,3,ret,0x7aa24
9f2d4,0x77ee47e130,2,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret
,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907de
cb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,
0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77
ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,
ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee
4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e2
70,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0
x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c
,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7
```

```
a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907
da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret
,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731
598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,
0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77
f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,
ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a90
7dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855f
f4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0
x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0
,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7
aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907
fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret
,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x7a907fa42c,0x7a88f5e
248,6,ret,0x7a88f5e270,0x77f0731568,5,ret,0x77f0731580,0x7aa249f1d4,4,ret,0x7aa249f1dc,
0x77ee47e214,3,ret,0x77ee47e28c,0x77ee47dfc0,2,ret,0x77ee47e35c,0x77ee47dfcc,2,ret,0x77
ee47dc44,0x77ee47dfdc,2,ret,0x7a907e88a0,0x7a90838c08,3,ret,0x7a90838c1c,0x77ee47e040,2,
ret,0x7a907e88a0,0x77ee47e048,2,ret,0x77ee47dc44,0x77ee47e12c,2,ret,0x7a907fa42c,0x7a88
f5e248,5,ret,0x7a88f5e270,0x77f07315c4,4,ret,0x77f07315d0,0x7aa249f2d0,3,ret,0x7aa249f2
d4,0x77ee47e130,2,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0
x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0
,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x7
7ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee4
7ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret
,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4
bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,
0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77
ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,
ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a90
7decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9
d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0
x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254
,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7
a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa24
9f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret
,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d8
9a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,
0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77
ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,
ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee
4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e2
70,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0
x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c
,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7
a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907
da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret
,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f
254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,
0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7a
a249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,
ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a90
7d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a
4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0
x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598
,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x7
7ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f
```

```
5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret
,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee853
55c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,
0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a
907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,
ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa2
49f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c51
14,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0
x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198
,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7
a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a908
55ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret
,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e
248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,
0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a
907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,
ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee
47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d
34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0
x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0
,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7
a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa24
9f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret
,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee853
5d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,
0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a
907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,
ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee
47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e2
70,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0
x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248
,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x7
7ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee4
7e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret
,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da
9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,
0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a
88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,
ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0
7315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535
f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0
x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0
,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x7
7ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee4
7ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret
,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee853
5d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,
0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77
ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,
ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a90
7d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc
74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0
x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254
,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7
a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa24
9f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret
```

```
,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d8
9a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,
0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a
907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,
ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee
8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa4
2c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0
x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90
,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7
a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907
d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret
,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731
598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,
0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77
f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,
ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a90
7dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855f
f4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0
x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248
,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x7
7ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907
fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret
,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47d
f90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,
0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a
907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,
ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0
731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f4
84,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0
x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198
,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7
a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a908
55ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret
,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e
248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,
0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a
907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,
ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee
47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d
34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0
x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0
,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x7
7f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77599
5f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret
,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c
198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,
0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a
90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,
ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88
f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f2
5c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0
x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8
,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x7
7ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907
d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret
,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee853
```

```
5b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,
0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a
907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,
ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee
47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d
34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0
x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0
,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x7
7f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee4
1d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret
,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee853
5d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,
0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a
907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,
ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee
8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315
a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0
x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0
,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x7
7ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a908
55d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret
,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47e
db0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,
0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77
ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,
ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee
8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1
b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0
x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4
,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x7
7ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07
315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret
,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731
5c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,
0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a
90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,
ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee
853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa4
2c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0
x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4
,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7
aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x7a907
fa42c,0x7a88f5e248,6,ret,0x7a88f5e270,0x77f0731568,5,ret,0x77f0731580,0x7aa249f1d4,4,ret
,0x7aa249f1dc,0x77ee47e214,3,ret,0x77ee47e28c,0x77ee47dfc0,2,ret,0x77ee47e35c,0x77ee47d
fcc,2,ret,0x77ee47dc44,0x77ee47dfdc,2,ret,0x7a907e4f5c,0x77ee47e040,2,ret,0x77ee47dc44,
0x77ee47e12c,2,ret,0x7a907fa42c,0x7a88f5e248,5,ret,0x7a88f5e270,0x77f07315c4,4,ret,0x77
f07315d0,0x7aa249f2d0,3,ret,0x7aa249f2d4,0x77ee47e130,2,ret,0x7a907fa42c,0x7a88f5e248,6,
ret,0x7a88f5e270,0x77f0731568,5,ret,0x77f0731580,0x7aa249f1d4,4,ret,0x7aa249f1dc,0x77ee
47e214,3,ret,0x77ee47e28c,0x77ee47dfc0,2,ret,0x77ee47e35c,0x77ee47dfcc,2,ret,0x7a907fa4
2c,0x7a88f5e248,5,ret,0x7a88f5e270,0x77f07315c4,4,ret,0x77f07315d0,0x7aa249f2d0,3,ret,0
x7aa249f2d4,0x77ee47e130,2,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c
,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7
a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907
da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47dc74,0x77ee47edb0,2,ret
,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e
248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,
```

```
0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a
907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,
ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee
47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d
34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0
x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0
,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x7
7f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee4
1d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret
,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee853
5d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,
0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a
907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,
ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x77ee47ed44,0x77ee
8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315
a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0
x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4
,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x7
7ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a908
55df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret
,0x7a907e46e8,0x7a907ddccc,4,ret,0x7a90855d34,0x7a907d89a8,6,ret,0x7a907d89b4,0x7a907dc
50c,5,ret,0x7a907dc518,0x7a907ddd5c,4,ret,0x7a907ddebc,0x7a907dddb8,4,ret,0x7a90855ff4,
0x7a907d8a3c,4,ret,0x7a907d8a4c,0x7a907dedb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a
907e4f5c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,
ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88
f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f2
5c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0
x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0
,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x7
7ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a908
55d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret
,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47e
db0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,
0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7a
a249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,
ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa2
49f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc
04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0
x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c
,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x7
7ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907
fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret
,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee853
5d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,
0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77
ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,
ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a90
7d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc
74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0
x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254
,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x7
7ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07
315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret
,0x7a907fa42c,0x7a88f5e248,6,ret,0x7a88f5e270,0x77f0731568,5,ret,0x77f0731580,0x7aa249f
1d4,4,ret,0x7aa249f1dc,0x77ee47e214,3,ret,0x77ee47e28c,0x77ee47dfc0,2,ret,0x77ee47e35c,
0x77ee47dfcc,2,ret,0x7a907fa42c,0x7a88f5e248,5,ret,0x7a88f5e270,0x77f07315c4,4,ret,0x77
```

```
f07315d0,0x7aa249f2d0,3,ret,0x7aa249f2d4,0x77ee47e130,2,ret,0x77ee47dc04,0x77ee47df90,2,
ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a90
7d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a
4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0
x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248
,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x7
7ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907
fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret
,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47d
f90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,
0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a
907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,
ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88
f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f2
5c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0
x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8
,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x7
7ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907
d89b4,0x7a907decb0,3,ret,0x7aa3c2c440,0x7a907e4120,5,ret,0x7a907e412c,0x7a907d8a98,4,ret
,0x7a907d8ab0,0x7a907ded64,3,ret,0x7a90855d34,0x7a907d89a8,6,ret,0x7a907d89b4,0x7a907dc
50c,5,ret,0x7a907dc518,0x7a907dcee4,4,ret,0x7a90855ff4,0x7a907d8a3c,5,ret,0x7a907d8a4c,
0x7a907dcf00,4,ret,0x7a90855d34,0x7a907d89a8,5,ret,0x7a907d89b4,0x7a907dcf30,4,ret,0x7a
907dcfe8,0x7a907ded74,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,
ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee
47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e2
70,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0
x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248
,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x7
7ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee4
7e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret
,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da
9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,
0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a
88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,
ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88
f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2
d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0
x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8
,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7
a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907
e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret
,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee853
5bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,
0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7a
a249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,
ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a90
7d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a
4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0
x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248
,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x7
7ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907
fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret
,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47d
f90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,
0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a
907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,
```

```
ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88
f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f2
5c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0
x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8
,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x7
7ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907
d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret
,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47e
e90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,
0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77
ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,
ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee
8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1
b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0
x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4
,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x7
7ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f
5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret
,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e
248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,
0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77
ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,
ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a90
7da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f
5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0
x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc
,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7
a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa24
9f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret
,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d8
9a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7aa3c2c440,0x7a907e4120,5,ret,0x7a907e412c,
0x7a907d8a98,4,ret,0x7a907d8ab0,0x7a907ded64,3,ret,0x7a90855d34,0x7a907d89a8,6,ret,0x7a
907d89b4,0x7a907dc50c,5,ret,0x7a907dc518,0x7a907dcee4,4,ret,0x7a90855ff4,0x7a907d8a3c,5,
ret,0x7a907d8a4c,0x7a907dcf00,4,ret,0x7a90855d34,0x7a907d89a8,5,ret,0x7a907d89b4,0x7a90
7dcf30,4,ret,0x7a907dcfe8,0x7a907ded74,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a
4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0
x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248
,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x7
7ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f
5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret
,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee853
55c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,
0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a
907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,
ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0
731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1
b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0
x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8
,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x7
7ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907
d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret
,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47e
e90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,
0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77
ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,
ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee
```

```
8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1
b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0
x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4
,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x7
7ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f
5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret
,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e
248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,
0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77
ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,
ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a90
7da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f
5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0
x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc
,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7
a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa24
9f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret
,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d8
9a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,
0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a
907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,
ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee
8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa4
2c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0
x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90
,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7
a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907
d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret
,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e
248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,
0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a
907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,
ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee
47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d
34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0
x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0
,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7
a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa24
9f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret
,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f
2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x7a907fa42c,
0x7a88f5e248,6,ret,0x7a88f5e270,0x77f0731568,5,ret,0x77f0731580,0x7aa249f1d4,4,ret,0x7a
a249f1dc,0x77ee47e214,3,ret,0x77ee47e28c,0x77ee47dfc0,2,ret,0x77ee47e35c,0x77ee47dfcc,2,
ret,0x7a907fa42c,0x7a88f5e248,5,ret,0x7a88f5e270,0x77f07315c4,4,ret,0x77f07315d0,0x7aa2
49f2d0,3,ret,0x7aa249f2d4,0x77ee47e130,2,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1
b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0
x7a907d89b4,0x7a907decb0,3,ret,0x7a907e46e8,0x7a907ddccc,4,ret,0x7a90855d34,0x7a907d89a8
,6,ret,0x7a907d89b4,0x7a907dc50c,5,ret,0x7a907dc518,0x7a907ddd5c,4,ret,0x7a907ddebc,0x7
a907dddb8,4,ret,0x7a90855ff4,0x7a907d8a3c,4,ret,0x7a907d8a4c,0x7a907dedb0,3,ret,0x7a908
55ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret
,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee853
5b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,
0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a
907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,
ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee
47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d
```

34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x7a90855df4,0x7a907dec6c,3,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907decb0,3,ret,0x7a90855ff4,0x7a907d8a3c,3,ret,0x7a907d8a4c,0x7a907da9c4,2,ret,0x7a907da9d4,0x77ee853590,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x77ee41d1b4,0x77ee4c4bb0,2,ret,0x77ee4c5114,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x77ee743674,0x77ee8535a4,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc74,0x77ee69c184,0,ret,0x77ee47dc04,0x77ee47df90,2,ret,0x77ee47e1b4,0x77ee85355c,1,ret,0x77ee743674,0x77ee8535a4,1,ret,0x77ee47dc74,0x77ee47edb0,2,ret,0x7a907e4f5c,0x77ee47ee90,2,ret,0x77ee47ed44,0x77ee8535b0,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f0731598,3,ret,0x77f07315a8,0x7aa249f254,2,ret,0x7aa249f25c,0x77ee8535bc,1,ret,0x775995f484,0x77ee8535d4,1,ret,0x7a907fa42c,0x7a88f5e248,4,ret,0x7a88f5e270,0x77f07315c4,3,ret,0x77f07315d0,0x7aa249f2d0,2,ret,0x7aa249f2d4,0x77ee8535d8,1,ret,0x77ee8535f8,0x77ee69c198,0,ret,0x77ee47dc74,0x77ee69c184,0,ret,0x77ee69c6c0,0x7708165bf8,1,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907da330,3,ret,0x7a90855ff4,0x7a907d8a3c,4,ret,0x7a907d8a4c,0x7a907da45c,3,ret,0x7a907da514,0x7a907dab70,2,ret,0x7a907dab84,0x7a907d5134,1,ret,0x7a907d514c,0x77081ca484,0,ret,0x77081ca4a0,0x7708165c14,1,ret,0x770809fd08,0x770815fe1c,0,ret,0x770815fe60,0x770815fe40,0,ret,0x770809fd08,0x770815fe94,0,ret,0x7708164948,0x7708164928,1,ret,0x77081649b0,0x7708164994,1,ret,0x77081649b0,0x77081649cc,1,ret,0x7708164a10,0x770815ff0c,0,ret,0x770815ff84,0x7708165c30,1,ret,0x7a90855d34,0x7a907d89a8,4,ret,0x7a907d89b4,0x7a907da374,3,ret,0x7a90855ff4,0x7a907d8a3c,4,ret,0x7a907d8a4c,0x7a907da45c,3,ret,0x7a907da514,0x7a907dab70,2,ret,0x7a907dab84,0x7a907d5134,1,ret,0x7a907d514c,0x77081ca484,0,ret,0x77081ca4a0,0x7708165c5c,1,ret,0x770809fd08,0x770815ffbc,0,ret,0x7708160000,0x770815ffe0,0,ret,0x770809fd08,0x7708160034,0,ret,0x7708164948,0x7708164928,1,ret,0x77081649b0,0x7708164994,1,ret,0x77081649b0,0x77081649cc,1,ret,0x7708164a10,0x7708160048,0,ret,0x7708160120,0x7708160100,0,ret,0x7708160274,0x7708165c8c,1,ret,0x7708180fb4,0x77081653e8,1,ret,0x77ee47dc04,0x77ee47df90,3,ret,0x7a907ee238,0x7aa264e240,4,ret,0x7aa264e244,0x77ee47e178,3,ret,0x77ee47e1b4,0x77ee660d54,2,ret,0x77ee47dc44,0x77ee480b4c,3,ret,0x77ee47dc74,0x77ee47edb0,4,ret,0x7a907e4f5c,0x77ee47ee90,4,ret,0x77ee47ed44,0x77ee480b68,3,ret,0x7a907e88a0,0x7a90838c08,4,ret,0x7a90838c1c,0x77ee480b88,3,ret,0x7a907e88a0,0x77ee480b90,3,ret,0x77ee47dc04,0x77ee47df90,4,ret,0x7a907ee238,0x7aa264e240,5,ret,0x7aa264e244,0x77ee47e178,4,ret,0x77ee47e1b4,0x77ee480bd8,3,ret,0x77ee47dc44,0x77ee480c08,3,ret,0x77ee480c3c,0x77ee660d6c,2,ret,0x77ee47dc74,0x77ee47edb0,3,ret,0x7a907e4f5c,0x77ee47ee90,3,ret,0x77ee47ed44,0x77ee660d84,2,ret,0x77ee47dc04,0x77ee47df90,3,ret,0x7a907ee238,0x7aa264e240,4,ret,0x7aa264e244,0x77ee47e178

```
,3,ret,0x77ee47e1b4,0x77ee660d54,2,ret,0x77ee47dc44,0x77ee480b4c,3,ret,0x77ee47dc74,0x7
7ee47edb0,4,ret,0x7a907e4f5c,0x77ee47ee90,4,ret,0x77ee47ed44,0x77ee480b68,3,ret,0x7a907
e4f5c,0x77ee480b88,3,ret,0x77ee47dc04,0x77ee47df90,4,ret,0x7a907ee238,0x7aa264e240,5,ret
,0x7aa264e244,0x77ee47e178,4,ret,0x77ee47e1b4,0x77ee480bd8,3,ret,0x77ee47dc44,0x77ee480
c08,3,ret,0x77ee480c3c,0x77ee660d6c,2,ret,0x77ee47dc74,0x77ee47edb0,3,ret,0x7a907e4f5c,
0x77ee47ee90,3,ret,0x77ee47ed44,0x77ee660d84,2,ret,0x77ee47dc04,0x77ee660c44,2,ret,0x77
ee5c99dc,0x77ee6b638c,3,ret,0x77ee6b645c,0x77ee660e0c,2,ret,0x77ee47dc04,0x77ee47df90,4,
ret,0x77ee47e1b4,0x77ee85355c,3,ret,0x7a90855df4,0x7a907dec6c,5,ret,0x7a90855d34,0x7a90
7d89a8,6,ret,0x7a907d89b4,0x7a907decb0,5,ret,0x7a90855ff4,0x7a907d8a3c,5,ret,0x7a907d8a
4c,0x7a907da9c4,4,ret,0x7a907da9d4,0x77ee853590,3,ret,0x77ee47dc74,0x77ee47edb0,4,ret,0
x77ee47ed44,0x77ee8535b0,3,ret,0x7a907fa42c,0x7a88f5e248,6,ret,0x7a88f5e270,0x77f0731598
,5,ret,0x77f07315a8,0x7aa249f254,4,ret,0x7aa249f25c,0x77ee8535bc,3,ret,0x775995f484,0x7
7ee8535d4,3,ret,0x7a907fa42c,0x7a88f5e248,6,ret,0x7a88f5e270,0x77f07315c4,5,ret,0x77f07
315d0,0x7aa249f2d0,4,ret,0x7aa249f2d4,0x77ee8535d8,3,ret,0x77ee8535f8,0x77ee661068,2,ret
,0x77ee47dc04,0x77ee47df90,4,ret,0x77ee47e1b4,0x77ee85355c,3,ret,0x7a90855df4,0x7a907de
c6c,5,ret,0x7a90855d34,0x7a907d89a8,6,ret,0x7a907d89b4,0x7a907decb0,5,ret,0x7a90855ff4,
0x7a907d8a3c,5,ret,0x7a907d8a4c,0x7a907da9c4,4,ret,0x7a907da9d4,0x77ee853590,3,ret,0x77
ee47dc74,0x77ee47edb0,4,ret,0x77ee47ed44,0x77ee8535b0,3,ret,0x7a907fa42c,0x7a88f5e248,6,
ret,0x7a88f5e270,0x77f0731598,5,ret,0x77f07315a8,0x7aa249f254,4,ret,0x7aa249f25c,0x77ee
8535bc,3,ret,0x775995f484,0x77ee8535d4,3,ret,0x7a907fa42c,0x7a88f5e248,6,ret,0x7a88f5e2
70,0x77f07315c4,5,ret,0x77f07315d0,0x7aa249f2d0,4,ret,0x7aa249f2d4,0x77ee8535d8,3,ret,0
x77ee8535f8,0x77ee661068,2,ret,0x77ee47dc04,0x77ee47df90,4,ret,0x77ee47e1b4,0x77ee85355c
,3,ret,0x77ee743674,0x77ee8535a4,3,ret,0x77ee47dc74,0x77ee47edb0,4,ret,0x77ee47ed44,0x7
7ee8535b0,3,ret,0x7a907fa42c,0x7a88f5e248,6,ret,0x7a88f5e270,0x77f0731598,5,ret,0x77f07
315a8,0x7aa249f254,4,ret,0x7aa249f25c,0x77ee8535bc,3,ret,0x775995f484,0x77ee8535d4,3,ret
,0x7a907fa42c,0x7a88f5e248,6,ret,0x7a88f5e270,0x77f07315c4,5,ret,0x77f07315d0,0x7aa249f
2d0,4,ret,0x7aa249f2d4,0x77ee8535d8,3,ret,0x77ee8535f8,0x77ee661068,2,ret,0x77ee47dc74,
0x77ee661054,2,ret,0x77ee6610b8,0x7708165400,1,ret,0x77ee6b5f00,0x7708165414,1,ret,0x77
ee47dc04,0x77ee662a50,2
+++ into iterator  startAddress 0x77ee662c0c, isAppCode false
...
+++ into iterator  startAddress 0x77ee64e7ec, isAppCode false

Thank you for using Frida
```

调试loginWithPhoneNbr函数

参考：

【iOS逆向】某营业厅算法分析_小陈_InfoQ写作社区

的 `ts` = `TypeScript` 代码的 `Frida` 的 `Stalker` 调试 `loginWithPhoneNbr` 函数的例子：

```
var addr = 0x0000000029FD08 // loginWithPhoneNbr函数的起始地址
var mainModule = Process.enumerateModules()[0];
console.log(JSON.stringify(mainModule));
var mainName: string = mainModule.name;
var baseAddr = Module.findBaseAddress(mainName);
Interceptor.attach(baseAddr.add(addr), {
    onEnter: function(args) {
        console.log(addr.toString(16), "= loginWithPhoneNbr onEnter =");
        var tid = Process.getCurrentThreadId();
        Stalker.follow(tid, {
            events: {
                call: true, // CALL instructions: yes please
                ret: false, // RET instructions
                exec: false, // all instructions: not recommended as it's
                block: false, // block executed: coarse execution trace
                compile: false // block compiled: useful for coverage
            },
            transform: (iterator: StalkerArm64Iterator) => {
                let instruction = iterator.next();
                const startAddress = instruction!.address;
                var isAppCode = startAddress.compare(baseAddr.add(addr)) >= 0 && startA
ddress.compare(baseAddr.add(addr).add(10000)) === -1;
                do {
                    if (isAppCode) {
                        if (instruction!.mnemonic === "bl") {
                            iterator.putCallout((ctx) => {
                                var arm64Context = ctx as Arm64CpuContext;
                                console.log("bl x0 = " + new ObjC.Object(arm64Context.x0
))
                                console.log("bl x1 = " + arm64Context.x1.readCString())
                            });
                        }
                    }
                    iterator.keep();
                } while ((instruction = iterator.next()) !== null);
            }
        })
    }, onLeave: function(retval) {
        console.log("retval:", new ObjC.Object(retval))
        console.log(addr.toString(16), "= loginWithPhoneNbr onLeave =");
    }
});
```

2025-06-04 09:29:55

某小说App的Stalker用法

参考：

Frida Stalker 是什么？ - 奋飞安全

的Stalker示例代码：

```
function trace_entry(baseAddr, tatgetAddr) {
  Interceptor.attach(tatgetAddr, {
    onEnter: function (args) {
      console.log("enter tatgetAddr=========================================================
================");

      this.pid = Process.getCurrentThreadId();
      Stalker.follow(this.pid, {
        events: {
          // 暂时不需要这些 events
          call: false,
          ret: false,
          exec: false,

          block: false,
          compile: false
        },
        onReceive: function (events) {
        },

        transform: function (iterator) {
          var instruction = iterator.next();
          const startAddress = instruction.address;
          // 从ida里面 找到 Java_com_baidu_searchbox_NativeBds_dae1 函数的 代码 在 0xE84 和
 0x126C 之间
          var isModule = startAddress.compare(baseAddr.add(0xE84)) >= 0 && startAddress.
compare(baseAddr.add(0x126C)) < 0;
          do {
            if (isModule) {
              console.log(instruction.address.sub(baseAddr) + "\t:\t" + instruction);

            if(instruction.address.sub(baseAddr) == 0xfb8){
              iterator.putCallout((context) => {
                var string = Memory.readCString(context["x21"]);
                console.log("####  key = " + string)
              })
            }
            }
            iterator.keep();
          } while ((instruction = iterator.next()) !== null);
        },

        onCallSummary: function (summary) {

        }
      });
```

```
    }, onLeave: function (retval) {
      Stalker.unfollow(this.pid);
      console.log("retval:" + retval);
      console.log("leave tatgetAddr==========================================================
================");
    }
  });
}
```

部分输出日志：

```
0xfb4           :         add x21, x21, x0, lsr #1
0xfb8           :         mov x0, x21
0xfbc           :         ldr x22, [x8, #0x580]
0xfc0           :         bl #0x7a751c55f0
####  key = D0CD8B760CE07BC3
0xfc4           :         mov x1, x0
0xfc8           :         mov x0, x20
0xfcc           :         blr x22
0xfd0           :         ldr x8, [x20]
```

misc/frida-stalker-example.py

别处：

misc/frida-stalker-example.py at master · poxyran/misc · GitHub

的某个Python代码内包含Frida的Stalker的js的例子：

```
import sys
import frida

def on_message(message, data):
    print "[%s] -> %s" % (message, data)

def main(target_process):
    session = frida.attach(target_process)
    script = session.create_script("""
function StalkerExeample()
{
    var threadIds = [];

    Process.enumerateThreads({
        onMatch: function (thread)
        {
            threadIds.push(thread.id);
            console.log("Thread ID: " + thread.id.toString());
        },

        onComplete: function ()
        {
            threadIds.forEach(function (threadId)
                {
                    Stalker.follow(threadId,
                    {
                        events: {call: true},

                    onReceive: function (events)
                    {
                        console.log("onReceive called.");
                    },
                    onCallSummary: function (summary)
                    {
                        console.log("onCallSummary called.");
                    }
                });
            });
        }
    });
}

StalkerExeample();
""")
    script.on('message', on_message)
    script.load()
```

```
    raw_input('[!] Press <Enter> at any time to detach from instrumented program.\n\n')
    session.detach()

if __name__ == '__main__':
    if len(sys.argv) < 2:
        print 'Usage: %s <process name or PID>' % __file__
        sys.exit(1)

    try:
        target_process = int(sys.argv[1])
    except ValueError:
        target_process = sys.argv[1]

    main(target_process)
```

Stalker心得

Stalker.follow

此处介绍Frida的Stalker中，相对最核心的Stalker.follow的相关心得。

Stalker的利用方式

Frida的Stalker的利用方式，目前算明白了，至少有2种：

- （1）给特定位置加上额外的代码
- （2）针对特定位置，去调试输出打印相关值
  - 查看寄存器的值
  - （读取）查看内存的值
  - 等等

举例说明：

（1）给特定位置加上额外的代码

官网的例子：

JavaScript API | Frida • A world-class dynamic instrumentation toolkit

```
//
// Advanced users: This is how you can plug in your own
//                 StalkerTransformer, where the provided
//                 function is called synchronously
//                 whenever Stalker wants to recompile
//                 a basic block of the code that's about
//                 to be executed by the stalked thread.
//
//transform(iterator) {
//  let instruction = iterator.next();
//
//  const startAddress = instruction.address;
//  const isAppCode = startAddress.compare(appStart) >= 0 &&
//      startAddress.compare(appEnd) === -1;
//
//  do {
//    if (isAppCode && instruction.mnemonic === 'ret') {
//      iterator.putCmpRegI32('eax', 60);
//      iterator.putJccShortLabel('jb', 'nope', 'no-hint');
//
//      iterator.putCmpRegI32('eax', 90);
//      iterator.putJccShortLabel('ja', 'nope', 'no-hint');
//
//      iterator.putCallout(onMatch);
//
//      iterator.putLabel('nope');
//    }
//
//    iterator.keep();
//  } while ((instruction = iterator.next()) !== null);
//},
//
// The default implementation is just:
//
```

```
//   while (iterator.next() !== null)
//     iterator.keep();
//
// The example above shows how you can insert your own code
// just before every `ret` instruction across any code
// executed by the stalked thread inside the app's own
// memory range. It inserts code that checks if the `eax`
// register contains a value between 60 and 90, and inserts
// a synchronous callout back into JavaScript whenever that
// is the case. The callback receives a single argument
// that gives it access to the CPU registers, and it is
// also able to modify them.
//
// function onMatch (context) {
//   console.log('Match! pc=' + context.pc +
//       ' rax=' + context.rax.toInt32());
// }
//
// Note that not calling keep() will result in the
// instruction getting dropped, which makes it possible
// for your transform to fully replace certain instructions
// when this is desirable.
//
```

意思是：

此处希望实现的特定效果

当发现是ret指令，即在ret真正运行，函数返回值，再去：

插入对应的代码：

```
//      iterator.putCmpRegI32('eax', 60);
//      iterator.putJccShortLabel('jb', 'nope', 'no-hint');
//
//      iterator.putCmpRegI32('eax', 90);
//      iterator.putJccShortLabel('ja', 'nope', 'no-hint');
//      iterator.putLabel('nope');
```

实现特定的逻辑：

检查eax寄存器是否包含 60 到 90 之间的值

-》如果你有类似需求，也可以去实现：

当满足条件（比如某个特定的指令，具体到某行代码），去插入特定代码，实现你的特定的逻辑

（2）针对特定位置，去调试输出打印相关值

比如我前面的：

___lldb_unnamed_symbol2575$$akd

-》核心代码是：

```
var funcRelativeStartAddr = 0xa0460;
var functionSize = 0x24C8; // 9416 == 0x24C8
var funcRelativeEndAddr = funcRelativeStartAddr + functionSize;
console.log("funcRelativeStartAddr=" + funcRelativeStartAddr + ", functionSize=" + func
tionSize + ", funcRelativeEndAddr=" + funcRelativeEndAddr);
const moduleName = "akd";
const moduleBaseAddress = Module.findBaseAddress(moduleName);
console.log("moduleName=" + moduleName + ", moduleBaseAddress=" + moduleBaseAddress);
// console.log("moduleName=%s, moduleBaseAddress=%p", moduleName, moduleBaseAddress);
const funcRealStartAddr = moduleBaseAddress.add(funcRelativeStartAddr);
// var funcRealEndAddr = funcRealStartAddr + functionSize;
const funcRealEndAddr = funcRealStartAddr.add(functionSize);
console.log("funcRealStartAddr=" + funcRealStartAddr + ", funcRealEndAddr=" + funcRealE
ndAddr);

Interceptor.attach(funcRealStartAddr, {
    onEnter: function(args) {
        var arg0 = args[0]
        var arg1 = args[1]
        var arg2 = args[2]
        var arg3 = args[3]
        console.log("----- arg0=" + arg0 + ", arg1=" + arg1 + ", arg2=" + arg2 + ", arg
3=" + arg3);
        var curTid = Process.getCurrentThreadId();
        console.log("curTid=", curTid);
        Stalker.follow(curTid, {
            events: {
                call: true, // CALL instructions: yes please
                ret: false, // RET instructions
                exec: false, // all instructions: not recommended as it's
                block: false, // block executed: coarse execution trace
                compile: false // block compiled: useful for coverage
            },
            // transform: (iterator: StalkerArm64Iterator) => {
            transform: function (iterator) {
                var instruction = iterator.next();
                const startAddress = instruction.address;
                const gt_realStartAddr = startAddress.compare(funcRealStartAddr) >= 0
                const lt_realEndAddr = startAddress.compare(funcRealEndAddr) < 0
                var isAppCode = gt_realStartAddr && lt_realEndAddr
                console.log("+++ into iterator: startAddress=" + startAddress + ", isAp
pCode=" + isAppCode);
                do {
                    if (isAppCode) {
                        // is origal function code = which we focus on


                        var curRealAddr = instruction.address;
                        // const isAppCode = curRealAddr.compare(funcRealStartAddr) >=
0 && curRealAddr.compare(funcRealEndAddr) === -1;
                        // console.log(curRealAddr + ": isAppCode=" + isAppCode);
                        var curOffsetHexPtr = curRealAddr.sub(funcRealStartAddr)
                        var curOffsetInt = curOffsetHexPtr.toInt32()

                        // var instructionStr = instruction.mnemonic + " " + instructio
n.opStr
```

```
                        var instructionStr = instruction.toString()
                        console.log("\t" + curRealAddr + " <+" + curOffsetInt + ">: " +
 instructionStr);
                        if (curOffsetInt == 8516) {
                            console.log("curOffsetInt=" + curOffsetInt);
                            iterator.putCallout((context) => {
                                var contextStr = JSON.stringify(context)
                                console.log("contextStr=" + contextStr);
                                var x9Value1 = context.x9
                                var x9Value2 = context["x9"]
                                console.log("x9Value1=" + x9Value1 + ", x9Value2=" + x9
Value2)
                            });
                        }
                    }
                    iterator.keep();
                } while ((instruction = iterator.next()) !== null);
            }
        });
```

其中就是：

当判断到是 +8516 行代码时，去查看x9的值（是否是我们希望的，最后所要返回的值）

所以就去判断：

- 是否是改行代码
  - `if (curOffsetInt == 8516) {`
- 如果是，则去：
  - 通过 `putCallout` ，传入匿名函数，其中去调试
    - 去打印 `x9` 寄存器的值

`Stalker.follow` 中的 `events` 的属性含义

对于：

```
Interceptor.attach(funcRealStartAddr, {
    onEnter: function(args) {
...
        Stalker.follow(curTid, {
            events: {
              call: false, // CALL instructions: yes please
              ret: true, // RET instructions
              exec: false, // all instructions: not recommended as it's
              block: false, // block executed: coarse execution trace
              compile: false // block compiled: useful for coverage
            },
            // onReceive: Called with `events` containing a binary blob comprised of on
e or more GumEvent structs. See `gumevent.h` for details about the format. Use `Stalker
.parse()` to examine the data.
            onReceive(events) {
              var parsedEvents = Stalker.parse(events)
              // var parsedEventsStr = JSON.stringify(parsedEventsStr)
              // console.log(">>> into onReceive: parsedEvents=" + parsedEvents + ", pa
rsedEventsStr=" + parsedEventsStr);
              console.log(">>> into onReceive: parsedEvents=" + parsedEvents);
            },

            // transform: (iterator: StalkerArm64Iterator) => {
            transform: function (iterator) {
...
```

中：

- `Stalker.follow` 中的 `events` 的属性含义
  - 概述
    - `Stalker.follow` 中的 `events` 中某个属性是 `true` ，含义是：当出现对应指令，则触发对应**event事件**
  - 详解
    - 属性
      - 对应的属性的含义是
        - `call` ：call指令
          - `Intel` = `X86` 的：`call` 指令
          - `ARM` 的：`BL` 类的指令
            - 普通的= `arm64` 的：`BL` 、`BLR` 等
            - `arm64e` 的，带 `PAC` 的：`BLRAA` 、`BLRAAZ` 、`BLRAB` 、`BLRABZ` 等
        - `ret` ：ret指令
        - `exec` ：所有指令
        - `block` ：（单个）block的（所有）指令
        - `compile` ：特殊，（单个）block被编译时，仅用于测试代码覆盖率?

        - 除去特殊的 `compile` 参数，其他几个参数，按照范围大小去划分，更容易理解：
            - `exec` ： 所有代码的级别
                - `block` ： 单个代码块的级别
                    - 某些特殊指令的级别
                        - `call` ： 单独的call指令
                        - `ret` ： 单独的ret指令
    - event事件
        - 会触发 `onReceive(events)` 函数
            - 其中可以 `events` 是二进制（的 `blob` ），需要去用 `Stalker.parse()` 解析后才能看懂
- -> `events` 和 `onReceive` 的作用
    - 暂时不完全懂，只是知道，可以设置参数，决定 `call` 、 `ret` 等指令的触发时去打印，其他用途暂时不清楚

Frida的Stalker中transform的逻辑

除了官网文档介绍了内部具体实现机制和过程之外：

对于，想要搞懂如何利用transform去调试代码来说：

需要明白的逻辑是：

此处代码：

```
Interceptor.attach(funcRealStartAddr, {
    onEnter: function(args) {
...
        var curTid = Process.getCurrentThreadId();
        console.log("curTid=", curTid);
        Stalker.follow(curTid, {
            events: {
                call: true, // CALL instructions: yes please
                ret: false, // RET instructions
                exec: false, // all instructions: not recommended as it's
                block: false, // block executed: coarse execution trace
                compile: false // block compiled: useful for coverage
            },
            // transform: (iterator: StalkerArm64Iterator) => {
            transform: function (iterator) {
                ...
```

触发到的 `transform的iterator` 来说：

- 每次触发=每个iterator：都（对应着）单个block=（basic）代码块
- 此处代码块有2种
  - 非原始函数代码 == `isAppCode=false`
    - 对应着应该是Stalker内部实现原理说的，copy拷贝出的代码
      - 其中会额外加上很多逻辑，用于实现Stalker的功能和逻辑
      - 估计就是这里说的这些内容
        - [原创] sktrace：基于 Frida Stalker 的 trace 工具-Android安全-看雪-安全社区|安全招聘|kanxue.com
          - 每当执行到一个基本块， `Stalker` 都会做以下几件事：
            1. 对于方法调用，保存 `lr` 等必要信息
            2. 重定位位置相关指令，例如： `ADR Xd, label`
            3. 建立此块的索引，如果此块在达到可信阈值后，内容未曾变化，下次将不再重新编译（为了加快速度）
            4. 根据 transform 函数，编译生成一个新的基本块 `GumExecBlock` ，保存到 `GumSlab` 。 `void transform(GumStalkerIterator iterator, GumStalkerOutput output, gpointer user_data)` 可以控制读取，改动，写入指令。
            5. transform 过程中还可通过 `void gum_stalker_iterator_put_callout (GumStalkerIterator self,GumStalkerCallout callout, gpointer`

`data, GDestroyNotify data_destroy)` 来设置一个当此位置被执行到时的 `callout` 。通过此 `void callout(GumCpuContext cpu_context, gpointer user_data)` 获取 `cpu` 信息。

6. 执行一个基本快 `GumExecBlock` ，开始下一个基本快

- 所以无需操作具体内部过程，直接忽略即可

- 原始函数代码 == `isAppCode=true`
  - 是实际运行的代码，才是我们所要关注的代码
    - 才会真正的去处理：比如判断是否是对应的（某个偏移量的）代码，然后去打印查看调试寄存器的值等等

此处去通过计算是否是 原始函数代码isAppCode 决定是否处理

具体详见示例代码：

- ___lldb_unnamed_symbol2575$$akd

`iterator.putCallout` 中的 `context` 的属性和含义

对于

```
transform: function (iterator) {
...
  iterator.putCallout((context) => {
...
```

中的 `context`，其定义是：

Stalker | Frida • A world-class dynamic instrumentation toolkit

```
typedef GumArm64CpuContext GumCpuContext;

struct _GumArm64CpuContext
{
  guint64 pc;
  guint64 sp;


  guint64 x[29];
  guint64 fp;
  guint64 lr;
  guint8 q[128];
};
```

某次调试时打印出的值是：

```
contextStr={"pc":"0x1041865a4","sp":"0x16bf46740","nzcv":1610612736,"x0":"0x0","x1":"0x
104ba0000","x2":"0xc","x3":"0x29","x4":"0x470b","x5":"0x0","x6":"0x0","x7":"0xec0","x8":
"0x0","x9":"0x3050500","x10":"0x16bf46838","x11":"0x6c245a87","x12":"0x6c245a3d","x13":
"0x1041864dc","x14":"0x45","x15":"0x0","x16":"0xd5709bdf0d7a48f0","x17":"0x123209fc0","
x18":"0x0","x19":"0xdf3221f55389ecc8","x20":"0x1233064b0","x21":"0x93dba5a2","x22":"0x0"
,"x23":"0x39207beb","x24":"0x6c245a87","x25":"0x1041a6cb0","x26":"0x6c245a87","x27":"0x
194d2c100","x28":"0xfffffffffffffffe","fp":"0x16bf46820","lr":"0x104186168","q0":{},"q1"
:{},"q2":{},"q3":{},"q4":{},"q5":{},"q6":{},"q7":{},"q8":{},"q9":{},"q10":{},"q11":{},"
q12":{},"q13":{},"q14":{},"q15":{},"q16":{},"q17":{},"q18":{},"q19":{},"q20":{},"q21":{}
,"q22":{},"q23":{},"q24":{},"q25":{},"q26":{},"q27":{},"q28":{},"q29":{},"q30":{},"q31":
{},"d0":0.5585262685374610e-308,"d1":0.0000169759663317e-308,"d2":7.9499288951273454e-2
75,"d3":0,"d4":0,"d5":0,"d6":0,"d7":0,"d8":0,"d9":0,"d10":0,"d11":0,"d12":0,"d13":0,"d1
4":0,"d15":0,"d16":1.0855813867524649e+251,"d17":0.000002084665455e-308,"d18":3.055469
8911305689e-152,"d19":0.000001322084152e-308,"d20":3.7696966457559071e-175,"d21":1.0004
894955531618e+128,"d22":9.5944260775693913e+225,"d23":-6.8154492514793903e-236,"d24":-2
.19309237443878 54e+50,"d25":-3.1153191556287330e+141,"d26":-1.0031492505499153e+302,"d2
7":-3.1584786060327890e-284,"d28":-1.7036104465458726e+239,"d29":-1.1538744009003510e-1
97,"d30":2.4944484440662111e-259,"d31":0,"s0":1.1035169354619356e-39,"s1":1.12103877145
98533e-44,"s2":3.8204714345426298e-37,"s3":0,"s4":0,"s5":0,"s6":0,"s7":0,"s8":0,"s9":0,
"s10":0,"s11":0,"s12":0,"s13":0,"s14":0,"s15":0,"s16":1.5548730664783592e-21,"s17":-1.3
084408235578860e+36,"s18":-8.6088756618229034e-23,"s19":-5.408784745233076e-20,"s20":4.
2151578028428229e+37,"s21":18362722504671230,"s22":3.372155249992253e+28,"s23":-7.67846
```

```
30118876258e-30,"s24":3715189.5,"s25":2.6426119211413095e+23,"s26":3.5724724140742379
e-23,"s27":1.6239861190373424e-18,"s28":205533200,"s29":0.008124308660626411,"s30":-
5.6424941355004989e-36,"s31":0}
```


格式化后：


```
{
    "pc": "0x1041865a4",
    "sp": "0x16bf46740",
    "nzcv": 1610612736,
    "x0": "0x0",
    "x1": "0x104ba0000",
    "x2": "0xc",
    "x3": "0x29",
    "x4": "0x470b",
    "x5": "0x0",
    "x6": "0x0",
    "x7": "0xec0",
    "x8": "0x0",
    "x9": "0x3050500",
    "x10": "0x16bf46838",
    "x11": "0x6c245a87",
    "x12": "0x6c245a3d",
    "x13": "0x1041864dc",
    "x14": "0x45",
    "x15": "0x0",
    "x16": "0xd5709bdf0d7a48f0",
    "x17": "0x123209fc0",
    "x18": "0x0",
    "x19": "0xdf3221f55389ecc8",
    "x20": "0x1233064b0",
    "x21": "0x93dba5a2",
    "x22": "0x0",
    "x23": "0x39207beb",
    "x24": "0x6c245a87",
    "x25": "0x1041a6cb0",
    "x26": "0x6c245a87",
    "x27": "0x194d2c100",
    "x28": "0xfffffffffffffffe",
    "fp": "0x16bf46820",
    "lr": "0x104186168",
    "q0": {},
    "q1": {},
    "q2": {},
    "q3": {},
    "q4": {},
    "q5": {},
    "q6": {},
    "q7": {},
    "q8": {},
    "q9": {},
    "q10": {},
    "q11": {},
    "q12": {},
```

```
    "q13": {},
    "q14": {},
    "q15": {},
    "q16": {},
    "q17": {},
    "q18": {},
    "q19": {},
    "q20": {},
    "q21": {},
    "q22": {},
    "q23": {},
    "q24": {},
    "q25": {},
    "q26": {},
    "q27": {},
    "q28": {},
    "q29": {},
    "q30": {},
    "q31": {},
    "d0": 0.5585262685374610e-308,
    "d1": 0.0000169759663317e-308,
    "d2": 7.9499288951273454e-275,
    "d3": 0,
    "d4": 0,
    "d5": 0,
    "d6": 0,
    "d7": 0,
    "d8": 0,
    "d9": 0,
    "d10": 0,
    "d11": 0,
    "d12": 0,
    "d13": 0,
    "d14": 0,
    "d15": 0,
    "d16": -1.0855813867524649e+251,
    "d17": 0.000002084565455e-308,
    "d18": 3.0554698911305689e-152,
    "d19": 0.000001322084152e-308,
    "d20": 3.7696966457559071e-175,
    "d21": 1.0004894955531618e+128,
    "d22": 9.5944260775693913e+225,
    "d23": -6.8154492514793903e-236,
    "d24": -2.1930923744387854e+50,
    "d25": -3.1153191556287330e+141,
    "d26": -1.0031492505499153e+302,
    "d27": -3.1584786060327890e-284,
    "d28": -1.7036104465458726e+239,
    "d29": -1.1538744009003510e-197,
    "d30": 2.4944484440662111e-259,
    "d31": 0,
    "s0": 1.1035169354619356e-39,
    "s1": 1.1210387714598533e-44,
    "s2": 3.8204714345426298e-37,
    "s3": 0,
    "s4": 0,
```

```
    "s5": 0,
    "s6": 0,
    "s7": 0,
    "s8": 0,
    "s9": 0,
    "s10": 0,
    "s11": 0,
    "s12": 0,
    "s13": 0,
    "s14": 0,
    "s15": 0,
    "s16": 1.5548730664783592e-21,
    "s17": -1.3084408235578860e+36,
    "s18": -8.6088756618229034e-23,
    "s19": -5.408784745233076e-20,
    "s20": 4.21515780284282229e+37,
    "s21": 18362722504671230,
    "s22": 3.372155249992253e+28,
    "s23": -7.6784630118876258e-30,
    "s24": -3715189.5,
    "s25": 2.64261192114130 95e+23,
    "s26": -3.5724724140742379e-23,
    "s27": -1.6239861190373424e-18,
    "s28": -205533200,
    "s29": -0.008124308660626411,
    "s30": -5.64249413550049 89e-36,
    "s31": 0
}
```


-》其中可以看到：各个寄存器的值

-》实现了调试的目的：查看寄存器等方面的值

Stalker.follow中的transform中的Instruction的属性

此处介绍 `Stalker.follow` 中 `transform` 中的 `指令` = `instruction` 的属性的详情：

Instruction有哪些属性

参考官网文档：Frida的Instruction（和其他相关资料），可以总结出：

- Instruction的属性
  - `address`
  - `next`
  - `size`
  - `mnemonic`
  - `opStr`
  - `operands`
  - `regsRead`
  - `regsWritten`
  - `groups`
  - `toString()`
  - `regsAccessed`
  - `toJSON()`

如何打印Instruction的属性

打印对应的属性值的代码：

```
console.log("instruction: address=" + instruction.address
  + ",next=" + instruction.next
  + ",size=" + instruction.size
  + ",mnemonic=" + instruction.mnemonic
  + ",opStr=" + instruction.opStr
  + ",operands=" + JSON.stringify(instruction.operands)
  + ",regsRead=" + JSON.stringify(instruction.regsRead)
  + ",regsWritten=" + JSON.stringify(instruction.regsWritten)
  + ",groups=" + JSON.stringify(instruction.groups)
  + ",toString()=" + instruction.toString()
);
```

输出效果：

```
instruction: address=0x10f4ecef4,next=0x4,size=4,mnemonic=ldr,opStr=x0, #0x10f4ecf78,op
erands=[{"type":"reg","value":"x0","access":"w"},{"type":"imm","value":"4551790456","ac
cess":"r"}],regsRead=[],regsWritten=[],groups=[],toString()=ldr x0, #0x10f4ecf78
[0x10f4ecef4] ldr x0, #0x10f4ecf78

instruction: address=0x10f4ecef8,next=0x4,size=4,mnemonic=bl,opStr=0x1091a500c,operands
=[{"type":"imm","value":"4447686668","access":"r"}],regsRead=[],regsWritten=["lr"],grou
```

```
ps=["call","jump","branch_relative"],toString()=bl #0x1091a500c
[0x10f4ecef8] bl #0x1091a500c
```

FridaUtil的printInstructionInfo

以及后续新版中，加上：`regsAccessed`、`toJSON()`后，经过整理，已提取到工具类函数中：

https://github.com/crifan/JsFridaUtil/blob/main/frida/FridaUtil.js

中的：

```
  static printInstructionInfo(instruction){
    // Instruction: address=0x252c0edf8,toString()=br x10,next=0x4,size=4,mnemonic=br,o
pStr=x10,operands=[{"type":"reg","value":"x10","access":"r"}],regsAccessed={"read":["x1
0"],"written":[]},regsRead=[],regsWritten=[],groups=["jump"],toJSON()={"address":"0x252
c0edf8","next":"0x4","size":4,"mnemonic":"br","opStr":"x10","operands":[{"type":"reg","
value":"x10","access":"r"}],"regsAccessed":{"read":["x10"],"written":[]},"regsRead":[],
"regsWritten":[],"groups":["jump"]}
    console.log("Instruction: address=" + instruction.address
      + ",toString()=" + instruction.toString()
      + ",toJSON()=" + JSON.stringify(instruction.toJSON())
      // + ",next=" + instruction.next
      // + ",size=" + instruction.size
      // + ",mnemonic=" + instruction.mnemonic
      // + ",opStr=" + instruction.opStr
      // + ",operands=" + JSON.stringify(instruction.operands)
      // + ",regsAccessed=" + JSON.stringify(instruction.regsAccessed)
      // + ",regsRead=" + JSON.stringify(instruction.regsRead)
      // + ",regsWritten=" + JSON.stringify(instruction.regsWritten)
      // + ",groups=" + JSON.stringify(instruction.groups)
    )
  }
```

以及之前的某次输出的效果是：

```
Instruction: address=0x252c0eca4,toString()=stp x20, x19, [sp, #0x40],next=0x4,size=4,m
nemonic=stp,opStr=x20, x19, [sp, #0x40],operands=[{"type":"reg","value":"x20","access":
"r"},{"type":"reg","value":"x19","access":"r"},{"type":"mem","value":{"base":"sp","disp"
:64},"access":"rw"}],regsAccessed={"read":["x20","x19","sp"],"written":[]},regsRead=[],
regsWritten=[],groups=[],toJSON()={"address":"0x252c0eca4","next":"0x4","size":4,"mnemo
nic":"stp","opStr":"x20, x19, [sp, #0x40]","operands":[{"type":"reg","value":"x20","acc
ess":"r"},{"type":"reg","value":"x19","access":"r"},{"type":"mem","value":{"base":"sp",
"disp":64},"access":"rw"}],"regsAccessed":{"read":["x20","x19","sp"],"written":[]},"reg
sRead":[],"regsWritten":[],"groups":[]}
```

供参考。

Instruction的属性的来源

而这些属性的来源：

- 主要是 `Capstone`
  - Java接口
    - capstone/Capstone.java at next · capstone-engine/capstone · GitHub
  - Python接口
    - capstone/__init__.py at next · capstone-engine/capstone · GitHub
- 其次是 `Frida` 内部
  - frida/frida-gum
    - https://github.com/frida/frida-gum/blob/main/bindings/gumjs/gumquickinstruction.c

下面摘录其相关的核心代码：

capstone/Capstone.java

capstone/Capstone.java

```
  protected static class _cs_insn extends Structure {
    // instruction ID.
    public int id;
    // instruction address.
    public long address;
    // instruction size.
    public short size;
    // machine bytes of instruction.
    public byte[] bytes;
    // instruction mnemonic. NOTE: irrelevant for diet engine.
    public byte[] mnemonic;
    // instruction operands. NOTE: irrelevant for diet engine.
    public byte[] op_str;
    // detail information of instruction.
    public _cs_detail.ByReference cs_detail;

...

  public static class CsInsn {
    private Pointer csh;
    private CS cs;
    private _cs_insn raw;
    private int arch;

    // instruction ID.
    public int id;
    // instruction address.
    public long address;
    // instruction size.
    public short size;
    // Machine bytes of this instruction, with number of bytes indicated by size above
    public byte[] bytes;
    // instruction mnemonic. NOTE: irrelevant for diet engine.
    public String mnemonic;
    // instruction operands. NOTE: irrelevant for diet engine.
    public String opStr;
    // list of all implicit registers being read.
    public short[] regsRead;
```

```
    // list of all implicit registers being written.
    public short[] regsWrite;
    // list of semantic groups this instruction belongs to.
    public byte[] groups;
    public OpInfo operands;
```

capstone/__init__.py

capstone/__init__.py

中相关核心代码：

```
class _cs_insn(ctypes.Structure):
    _fields_ = (
        ('id', ctypes.c_uint),
        ('alias_id', ctypes.c_uint64),
        ('address', ctypes.c_uint64),
        ('size', ctypes.c_uint16),
        ('bytes', ctypes.c_ubyte * 24),
        ('mnemonic', ctypes.c_char * 32),
        ('op_str', ctypes.c_char * 160),
        ('is_alias', ctypes.c_bool),
        ('usesAliasDetails', ctypes.c_bool),
        ('illegal', ctypes.c_bool),
        ('detail', ctypes.POINTER(_cs_detail)),
    )

...

# Python-style class to disasm code
class CsInsn(object):
    def __init__(self, cs, all_info):
        self._raw = copy_ctypes(all_info)
        self._cs = cs
        if self._cs._detail and not self.is_invalid_insn():
            # save detail
            self._raw.detail = ctypes.pointer(all_info.detail._type_())
            ctypes.memmove(ctypes.byref(self._raw.detail[0]), ctypes.byref(all_info.det
ail[0]),
                           ctypes.sizeof(type(all_info.detail[0])))

    def __repr__(self):
        return '<CsInsn 0x%x [%s]: %s %s>' % (self.address, self.bytes.hex(), self.mnem
onic, self.op_str)

    # return if the instruction is invalid
    def is_invalid_insn(self):
        arch = self._cs.arch
        if arch == CS_ARCH_EVM:
            return self.id == evm.EVM_INS_INVALID
        else:
            return self.id == 0

    # return instruction's ID.
```

```
    @property
    def id(self):
        return self._raw.id

    # return instruction's address.
    @property
    def address(self):
        return self._raw.address

    # return instruction's size.
    @property
    def size(self):
        return self._raw.size

    # return instruction's is_alias flag
    @property
    def is_alias(self):
        return self._raw.is_alias

    # return instruction's illegal flag
    # Set if instruction can be decoded but is invalid
    # due to context or illegal operands.
    @property
    def illegal(self):
        return self._raw.illegal

    # return instruction's alias_id
    @property
    def alias_id(self):
        return self._raw.alias_id

    # return instruction's flag if it uses alias details
    @property
    def uses_alias_details(self):
        return self._raw.usesAliasDetails

    # return instruction's machine bytes (which should have @size bytes).
    @property
    def bytes(self):
        return bytearray(self._raw.bytes)[:self._raw.size]

    # return instruction's mnemonic.
    @property
    def mnemonic(self):
        if self._cs._diet:
            # Diet engine cannot provide @mnemonic.
            raise CsError(CS_ERR_DIET)

        return self._raw.mnemonic.decode('ascii')

    # return instruction's operands (in string).
    @property
    def op_str(self):
        if self._cs._diet:
            # Diet engine cannot provide @op_str.
            raise CsError(CS_ERR_DIET)
```

```
        return self._raw.op_str.decode('ascii')

    # return list of all implicit registers being read.
    @property
    def regs_read(self):
        if self.is_invalid_insn():
            raise CsError(CS_ERR_SKIPDATA)

        if self._cs._diet:
            # Diet engine cannot provide @regs_read.
            raise CsError(CS_ERR_DIET)

        if self._cs._detail:
            return self._raw.detail.contents.regs_read[:self._raw.detail.contents.regs_
read_count]

        raise CsError(CS_ERR_DETAIL)

    # return list of all implicit registers being modified
    @property
    def regs_write(self):
        if self.is_invalid_insn():
            raise CsError(CS_ERR_SKIPDATA)

        if self._cs._diet:
            # Diet engine cannot provide @regs_write
            raise CsError(CS_ERR_DIET)

        if self._cs._detail:
            return self._raw.detail.contents.regs_write[:self._raw.detail.contents.regs
_write_count]

        raise CsError(CS_ERR_DETAIL)

    # return list of semantic groups this instruction belongs to.
    @property
    def groups(self):
        if self.is_invalid_insn():
            raise CsError(CS_ERR_SKIPDATA)

        if self._cs._diet:
            # Diet engine cannot provide @groups
            raise CsError(CS_ERR_DIET)

        if self._cs._detail:
            return self._raw.detail.contents.groups[:self._raw.detail.contents.groups_c
ount]

        raise CsError(CS_ERR_DETAIL)

    # return whether instruction has writeback operands.
    @property
    def writeback(self):
        if self.is_invalid_insn():
            raise CsError(CS_ERR_SKIPDATA)
```

```
        if self._cs._diet:
            # Diet engine cannot provide @writeback.
            raise CsError(CS_ERR_DIET)

        if self._cs._detail:
            return self._raw.detail.contents.writeback

        raise CsError(CS_ERR_DETAIL)
```

frida-gum/bindings/gumjs/gumquickinstruction.c

frida-gum/bindings/gumjs/gumquickinstruction.c at main · frida/frida-gum

的相关核心代码：

```
static const JSCFunctionListEntry gumjs_instruction_entries[] =
{
  JS_CGETSET_DEF ("address", gumjs_instruction_get_address, NULL),
  JS_CGETSET_DEF ("next", gumjs_instruction_get_next, NULL),
  JS_CGETSET_DEF ("size", gumjs_instruction_get_size, NULL),
  JS_CGETSET_DEF ("mnemonic", gumjs_instruction_get_mnemonic, NULL),
  JS_CGETSET_DEF ("opStr", gumjs_instruction_get_op_str, NULL),
  JS_CGETSET_DEF ("operands", gumjs_instruction_get_operands, NULL),
  JS_CGETSET_DEF ("regsAccessed", gumjs_instruction_get_regs_accessed, NULL),
  JS_CGETSET_DEF ("regsRead", gumjs_instruction_get_regs_read, NULL),
  JS_CGETSET_DEF ("regsWritten", gumjs_instruction_get_regs_written, NULL),
  JS_CGETSET_DEF ("groups", gumjs_instruction_get_groups, NULL),
  JS_CFUNC_DEF ("toString", 0, gumjs_instruction_to_string),
  JS_CFUNC_DEF ("toJSON", 0, gumjs_instruction_to_json),
};
```

`Stalker.follow()` 内部实现原理

当用户调用 `Stalker.follow()` 时，内部调用：

- 要么是： `gum_stalker_follow_me()` ： 去跟踪当前的线程thread
  - 函数原型

```
GUM_API void gum_stalker_follow_me (GumStalker * self, GumStalkerTransformer *
transformer, GumEventSink * sink);
```

  - 底层 `JS` 引擎：是 `QuickJS` 或 `V8`
- 要么是： `gum_stalker_follow(thread_id)` ： 去跟踪当前process进程中的其他某个线程thread

`gum_stalker_follow_me` 的内部原理

```
#ifdef __APPLE__

.globl _gum_stalker_follow_me

_gum_stalker_follow_me:
#else
.globl gum_stalker_follow_me

.type gum_stalker_follow_me, %function

gum_stalker_follow_me:
#endif
stp x29, x30, [sp, -16]!

mov x29, sp

mov x3, x30

#ifdef __APPLE__

bl __gum_stalker_do_follow_me

#else
bl _gum_stalker_do_follow_me

#endif
ldp x29, x30, [sp], 16

br x0
```

-》

- 内部原理
  - LR=Link Register=X30=链接寄存器
    - AArch64架构中，根据LR去决定从哪里开始跟踪

        - 当遇到BR，BLR等跳转指令时，会去设置LR
            - LR被设置为，当前函数返回后，继续运行的地址
        - 由于只有一个LR，如果被调用函数调用了其他函数，此时LR的值就会被保存起来，比如保存到Stack栈上，后续当RET指令执行之前，会重新把LR从Stack栈中加载到寄存器中，最终返回到调用者
    - FP=Frame Pointer=X29=帧指针
        - FP始终指向Stack top栈的顶部，表示当前函数被调用时的栈的位置
        - 所以就可以通过固定的偏移量去访问到，所有通过Stack栈传入的参数和基于栈的局部变量
        - 且每个函数都有自己的FP，所以需要调用新函数时，保存之前的FP，返回之前函数时，恢复FP。
        - 在刚进入新函数后，备份FP后，就可以去设置：mov x29, sp，把SP给X29=FP了。

保持了原先传入 `x0-x2` 的3个参数，额外加上 `x3` = `x30` = `LR` ，所以再去调用函数，就对应上参数了：

```
gpointer gum_stalker_do_follow_me (GumStalker * self, GumStalkerTransformer * transform
er, GumEventSink * sink, gpointer ret_addr)
```

`gum_stalker_follow` 的内部原理

和 `gum_stalker_follow_me()` 类似，但有额外参数：thread_id

```
void
gum_stalker_follow (GumStalker * self,
                    GumThreadId thread_id,
                    GumStalkerTransformer * transformer,
                    GumEventSink * sink)
{
  if (thread_id == gum_process_get_current_thread_id ())
  {
    gum_stalker_follow_me (self, transformer, sink);
  }
  else
  {
    GumInfectContext ctx;

    ctx.stalker = self;
    ctx.transformer = transformer;
    ctx.sink = sink;

    gum_process_modify_thread (thread_id, gum_stalker_infect, &ctx);
  }
}
```

其中： `gum_process_modify_thread()` ，不属于 `Stalker` ，但属于 `Gum`

回调callback会去修改： `GumCpuContext`

GumCpuContext

GumCpuContext的定义：

```
typedef GumArm64CpuContext GumCpuContext;

struct _GumArm64CpuContext
{
  guint64 pc;
  guint64 sp;


  guint64 x[29];
  guint64 fp;
  guint64 lr;
  guint8 q[128];
};
```

相关：

```
static void
gum_stalker_infect (GumThreadId thread_id,
                    GumCpuContext * cpu_context,
                    gpointer user_data)
```

- `gum_process_modify_thread()` 内部实现
  - Linux/Android：ptrace
    - GDB也用的这个：挂载到进程上，读写寄存器

Stalker每次只处理一个代码块block

内部机制：

新申请一块内存，写入给原始代码中加了调试代码后的代码

加的指令，用于生成事件、提供其他Stalker所支持的功能。

以及根据情况去relocate重定位指令代码。

比如对于下面代码，要重定位：

- ADR Address of label at a PC-relative offset.
- `ADR Xd, label`
- `Xd` Is the 64-bit name of the general-purpose destination register, in the range 0 to 31.
- `label` Is the program label whose address is to be calculated. It is an offset from the address of this instruction, in the range ±1MB.

底层通过Gum的Relocator

frida-gum/gumarm64relocator.c at main · frida/frida-gum · GitHub

现在，回想一下我们说过潜行者一次工作一个块。那么我们如何检测下一个块呢？我们还记得每个块也以分支指令结尾，如果我们修改这个分支以分支回 Stalker 引擎，但确保我们存储分支打算结束的目的地，我们可以检测下一个块并在那里重定向执行。这个相同的简单过程可以一个接一个地继续。

Stalker=潜行者

现在，这个过程可能有点慢，因此我们可以应用一些优化。首先，如果我们多次执行相同的代码块（例如循环，或者可能只是一个多次调用的函数），我们不必重新检测它。我们可以重新执行相同的检测代码。出于这个原因，我们保留了一个哈希表，其中包含我们之前遇到的所有块以及我们放置块的检测副本的位置。

其次，当遇到呼叫指令时，在发出检测的呼叫后，我们随后会发出一个着陆板，我们可以返回该着陆板而无需重新进入 Stalker。Stalker使用记录真实返回地址（real_address）和此着陆垫（code_address）的GumExecFrame结构构建了一个侧堆栈。当一个函数返回时，我们发出代码，该代码将根据real_address检查侧堆栈中的返回地址，如果匹配，它可以简单地返回到code_address，而无需重新进入运行时。这个着陆板最初将包含进入 Stalker 引擎以检测下一个块的代码，但稍后可以将其反向修补以直接分支到该块。这意味着可以处理整个返回序列，而无需输入和离开 Stalker。

如果返回地址与存储的GumExecFrame real_address不匹配，或者我们在侧堆栈中的空间不足，我们只需从头开始重新构建一个新的。我们需要在应用程序代码执行时保留 LR 的值，以便应用程序不能使用它来检测 Stalker 的存在（反调试），或者如果它将其用于除简单返回之外的任何其他目的（例如，在代码部分中引用内联数据）。此外，我们希望 Stalker 能够随时取消关注，所以我们不想不得不返回我们的堆栈来更正我们在此过程中修改的 LR 值。

最后，虽然我们总是用对 Stalker 的调用来替换分支以检测下一个块，但根据 Stalker.trustThreshold 的配置，我们可能会对此类检测代码进行反向修补，以将调用替换为下一个检测块的直接分支。确定性分支（例如，目的地是固定的，分支不是有条件的）很简单，我们可以将分支替换为下一个块。但是我们也可以处理条件分支，如果我们检测两个代码块（一个是分支，一个是不是）。然后，我们可以将原始条件分支替换为一个条件分支，该条件分支将控制流定向到获取分支时遇到的块的检测版本，然后是另一个检测块的无条件分支。我们还可以部分处理目标不是静态的分支。假设我们的分支是这样的：

BR X0

这种指令在调用函数指针或类方法时很常见。虽然 X0 的值可以更改，但通常它实际上总是相同的。在这种情况下，我们可以将最终的分支指令替换为代码，该代码将 X0 的值与我们的已知函数进行比较，如果它将分支与代码的检测副本的地址匹配。然后，如果不匹配，则可以将无条件分支返回到 Stalker 引擎。因此，如果函数指针 say 的值发生了变化，那么代码仍然有效，无论我们最终到达哪里，我们都将重新输入 Stalker 和乐器。但是，如果如我们预期的那样它保持不变，那么我们可以完全绕过 Stalker 引擎并直接进入仪器化功能。

Stalker中函数地址和指令地址匹配不上

对于之前Stalker代码

___lldb_unnamed_symbol2575$$akd

的某次调试输出日志是：

```
===== dynamicDebug frida scripts fridaStalker_akdSymbol2575.js =====
funcRelativeStartAddr 656480, functionSize 9416, funcRelativeEndAddr 665896
moduleName akd, moduleBaseAddress 0x10054c000
funcRealStartAddr 0x1005ec460, funcRealEndAddr 0x1005ec4609416
[iPhone::PID::13098 ]-> ----- arg0 0xfffffffffffffffe, arg1 0x16fe26838, arg2 0x16fe268
38, arg3 0xfffffffffffffffe

curTid= 26635

+++ into iterator=
[0x1071dbcd8] b #0x1071dbce8
+++ into iterator=
[0x1071dbce8] str wzr, [x19, #0x90]
[0x1071dbcec] ldr x0, [x19, #0xa0]
[0x1071dbcf0] str xzr, [x19, #0xa0]
[0x1071dbcf4] cbz x0, #0x1071dbcfc
+++ into iterator=
[0x1071dbcf8] bl #0x1070dfef8
+++ into iterator=
[0x1071dbcfc] ldr x0, [x19, #0x98]
[0x1071dbd00] str xzr, [x19, #0x98]
[0x1071dbd04] cbz x0, #0x1071dbd14
...
```

-》看到2个函数地址不匹配：

- 函数实际内存起始地址：`0x1005ec460`
  - `0x1005exxxx` 之类的地址
- Stalker中运行期间的函数地址：`0x1071dbcd8`
  - `0x1071dxxxx` 之类的地址

后来明白原因了：

Stalker内部的hook指令的原理就是：

把当前代码拷贝一份，加上hook代码及其他逻辑

而此处的：拷贝一份，到别处，别的一段内存地址空间，此处就是指的是：

- `0x1071dbcd8`
  - `0x1071dxxxx` 之类的地址

所以，原始函数代码和被Stalker去hook的代码，两个地址是不同的，是可以理解的，没有出错，是正常的。

16字节之前的开始那段代码无法调试

Frida的Stalker，触发的时候，往往第一行代码，确切的说是前面一小段代码，是无法打印出来的

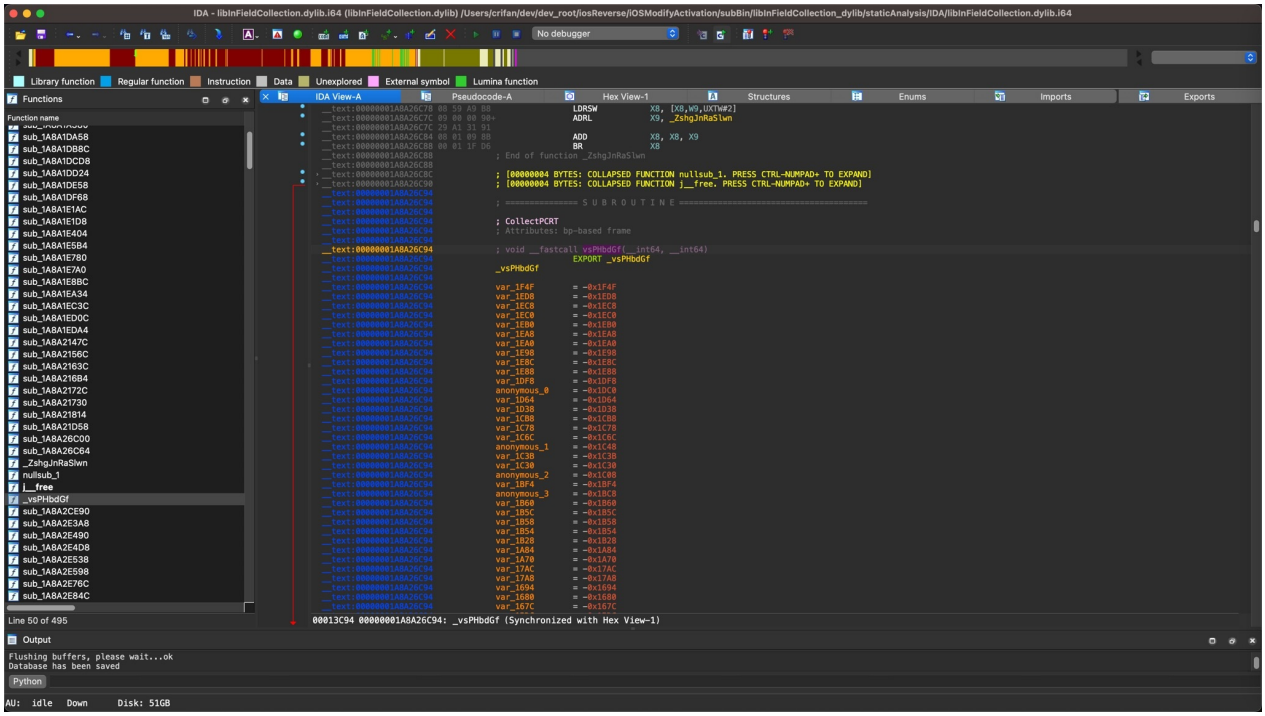
而能开始打印的时候，往往是代码向后偏移大概0x10=16字节之后，才能打印出来

举例

想要去hook函数：

- libInFieldCollection.dylib的vsPHbdGf

IDA中，起始地址是：



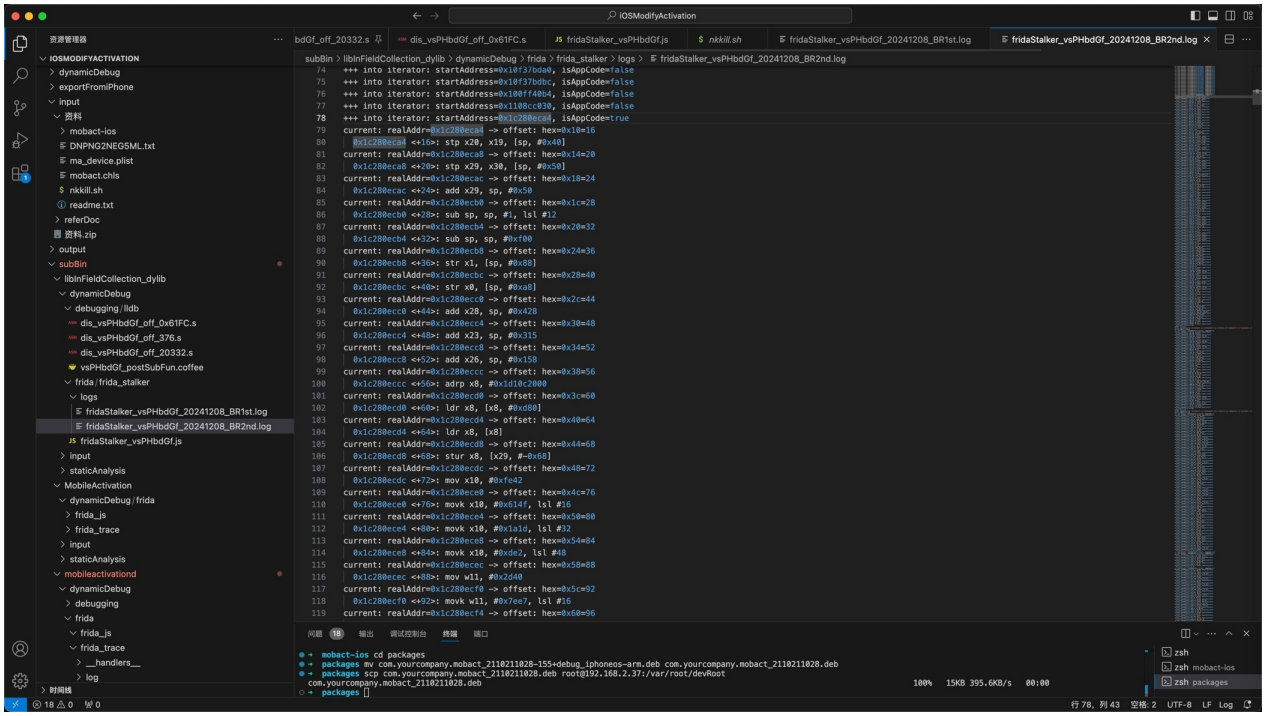

```
__text:00000001A8A26C94                 ; CollectPCRT
__text:00000001A8A26C94                 ; Attributes: bp-based frame
__text:00000001A8A26C94
__text:00000001A8A26C94                 ; void __fastcall vsPHbdGf(__int64, __int64)
__text:00000001A8A26C94                                 EXPORT _vsPHbdGf
...
__text:00000001A8A26C94 FC 6F BA A9                     STP             X28, X27, [SP,# 0x10
var_50]
__text:00000001A8A26C98 FA 67 01 A9                     STP             X26, X25, [SP,#0x50
var_40]
__text:00000001A8A26C9C F8 5F 02 A9                     STP             X24, X23, [SP,#0x50
var_30]
__text:00000001A8A26CA0 F6 57 03 A9                     STP             X22, X21, [SP,#0x50
var_20]
__text:00000001A8A26CA4 F4 4F 04 A9                     STP             X20, X19, [SP,#0x50
var_10]
__text:00000001A8A26CA8 FD 7B 05 A9                     STP             X29, X30, [SP,#0x50
var_s0]
```

```
__text:00000001A8A26CAC FD 43 01 91                 ADD             X29, SP, #0x50
__text:00000001A8A26CB0 FF 07 40 D1                 SUB             SP, SP, #1,LSL#12
```

libInFieldCollection.dylib的vsPHbdGf

中的输出是：

```
into hookNative:
into stalkerHookNative_vsPHbdGf:
libraryName libInFieldCollection.dylib -> moduleBaseAddress 0x1c27fb000
origFuncPtr 0x1c280ec94
funcRelativeStartAddr 81044
functionSize 25084 0x61fc, argNum 2
Frida Stalker hook: module: baseAddress 0x1c27fb000
function: relativeStartAddr 0x13c94, size 25084 0x61fc, relativeEndAddr 0x19e90
funcRealStartAddr 0x1c280ec94, funcRealEndAddr 0x1c2814e90
curTid null
[iPhone: mobileactivationd ]-> ============= Trigged addr: relative [0x13c94] = real [0x
1c280ec94] =============
arg[0] 0x16fa8a988
arg[1] 0x16fa8a984
curTid 21019
+++ into iterator: startAddress 0x10f3c1768, isAppCode=false
+++ into iterator: startAddress 0x10f3c1778, isAppCode=false
...
+++ into iterator: startAddress 0x1108cc030, isAppCode=false
+++ into iterator: startAddress 0x1c280eca4, isAppCode=true
current: realAddr 0x1c280eca4 -> offset: hex=0x10=16
    0x1c280eca4 <+16>: stp x20, x19, [sp, #0x40]
current: realAddr 0x1c280eca8 -> offset: hex=0x14=20
    0x1c280eca8 <+20>: stp x29, x30, [sp, #0x50]
current: realAddr 0x1c280ecac -> offset: hex=0x18=24
    0x1c280ecac <+24>: add x29, sp, #0x50
...
```



此时：

- 函数的第一行代码=函数起始地址是：`0x1c280ec94`
- 真正开始打印log日志的地址是：`0x1c280eca4`
  - 已经是起始地址的偏移量：`0x10 = 16` 了

原因

猜测是和Frida的Stalker的内部实现机制有关系：

好像大概意思是，从原始代码code拷贝到新的地方，再去hook和打印

使得好像无法实现最初代码的hook？？

影响和结果

无法实现，去函数最开始的几行（4行）代码去hook（比如打印输出参数时的寄存器值等等）

即：

此处开始打印hook的代码，已经是：

```
__text:00000001A8A26C94 FC 6F BA A9                 STP             X28, X27, [SP,# 0x10
var_50]
__text:00000001A8A26C98 FA 67 01 A9                 STP             X26, X25, [SP,#0x50
var_40]
__text:00000001A8A26C9C F8 5F 02 A9                 STP             X24, X23, [SP,#0x50
var_30]
__text:00000001A8A26CA0 F6 57 03 A9                 STP             X22, X21, [SP,#0x50
var_20]
__text:00000001A8A26CA4 F4 4F 04 A9                 STP             X20, X19, [SP,#0x50
var_10]
```

中的：

```
__text:00000001A8A26CA4 F4 4F 04 A9                 STP             X20, X19, [SP,#0x50+var_10]
```

对应着输出的：

- 偏移量是：`0x10` = `16` 的 `0x1c280eca4`

```
+++ into iterator: startAddress 0x1c280eca4, isAppCode=true
current: realAddr 0x1c280eca4 -> offset: hex=0x10=16
    0x1c280eca4 <+16>: stp x20, x19, [sp, #0x40]
```

而要去hook前面（16字节=0x10=占用4个指令的大小）4行代码：

```
__text:00000001A8A26C94 FC 6F BA A9                 STP             X28, X27, [SP,# 0x10+var_50]
__text:00000001A8A26C98 FA 67 01 A9                 STP             X26, X25, [SP,#0x50+var_40]
__text:00000001A8A26C9C F8 5F 02 A9                 STP             X24, X23, [SP,#0x50+var_30]
__text:00000001A8A26CA0 F6 57 03 A9                 STP             X22, X21, [SP,#0x50+var_20]
```

目前看来，就无法实现了

如何解决

暂时无解。

除非以后对Stalker机制有深入了解，看看是否有机会解决此，感觉算是Stalker的bug了。

Stalker.exclude

此处介绍 `Stalker.exclude` 的使用心得。

用Stalker.exclude提高效率和减少干扰

现象

libInFieldCollection.dylib的vsPHbdGf

中的Frida的Stalker的核心部分代码：

```
function stalkerHookNative_vsPHbdGf(){
  console.log("into stalkerHookNative_vsPHbdGf:")


  var libraryName = "libInFieldCollection.dylib"
  const moduleBaseAddress = Module.findBaseAddress(libraryName)
  console.log("libraryName=" + libraryName + " -> moduleBaseAddress=" + moduleBaseAddre
ss)
  if (null == moduleBaseAddress) {
    console.error("Failed to find library " + libraryName)
    return
  }


  var origFuncPtr = Module.findExportByName(libraryName, "vsPHbdGf")
  console.log("origFuncPtr=" + origFuncPtr)

  var funcRelativeStartAddr = origFuncPtr - moduleBaseAddress
  console.log("funcRelativeStartAddr=" + funcRelativeStartAddr)

  var functionSize = 0x61FC

  // CollectPCRT == void __fastcall vsPHbdGf(__int64 a1, __int64 a2)
  var argNum = 2
  console.log("functionSize=" + functionSize + "=" + JsUtil.intToHexStr(functionSize) +
", argNum=" + argNum)

    Stalker.exclude(module_system)
  }


  let hookFuncMap = {
    0x164: // +356
      function (context) {
        var contextStr = JSON.stringify(context)
        console.log("[0x164] contextStr=" + contextStr)
        var x10 = context.x10
        console.log("[0x164] x10=" + x10)
      },
    0x1DC: // +476
      function (context) {
        var contextStr = JSON.stringify(context)
        console.log("[0x1DC] contextStr=" + contextStr)
        var x10 = context.x10
        console.log("[0x1DC] x10=" + x10)
```

```
      },
  }

  FridaUtil.stalkerHookUnnameNative(moduleBaseAddress, funcRelativeStartAddr, functionS
ize, argNum, hookFuncMap)
}


function hookNative(){
  console.log("into hookNative:")
  stalkerHookNative_vsPHbdGf()
}

hookNative()
```

和对应的:

`FridaUtil` 的 `stalkerHookUnnameNative`

```
  // Frida Stalker hoo unknown name native function
  static stalkerHookUnnameNative(moduleBaseAddress, funcRelativeStartAddr, functionSize,
 argNum, hookFuncMap){
    console.log("Frida Stalker hook: module: baseAddress=" + moduleBaseAddress)


    var functionSizeHexStr = JsUtil.intToHexStr(functionSize)
    var funcRelativeStartAddrHexStr = JsUtil.intToHexStr(funcRelativeStartAddr)
    var funcRelativeEndAddr = funcRelativeStartAddr + functionSize
    var funcRelativeEndAddrHexStr = JsUtil.intToHexStr(funcRelativeEndAddr)
    console.log("function: relativeStartAddr=" + funcRelativeStartAddrHexStr + ", size="
 + functionSize + "=" + functionSizeHexStr + ", relativeEndAddr=" + funcRelativeEndAddr
HexStr)


    const funcRealStartAddr = moduleBaseAddress.add(funcRelativeStartAddr)
    // var funcRealEndAddr = funcRealStartAddr + functionSize
    const funcRealEndAddr = funcRealStartAddr.add(functionSize)
    console.log("funcRealStartAddr=" + funcRealStartAddr + ", funcRealEndAddr=" + funcR
ealEndAddr)
    var curTid = null
    console.log("curTid=" + curTid)
    Interceptor.attach(funcRealStartAddr, {
      onEnter: function(args) {
        JsUtil.logStr("Trigged addr: relative [" + funcRelativeStartAddrHexStr + "] = r
eal [" + funcRealStartAddr + "]")


        for(var i = 0; i < argNum; i ++) {
          var curArg = args[i]
          console.log("arg[" + i + "]=" + curArg)
        }


        var curTid = Process.getCurrentThreadId()
```

```
        console.log("curTid=" + curTid)
        Stalker.follow(curTid, {
            events: {
              call: false, // CALL instructions: yes please
              ret: true, // RET instructions
              exec: false, // all instructions: not recommended as it's
              block: false, // block executed: coarse execution trace
              compile: false // block compiled: useful for coverage
            },
            // onReceive: Called with `events` containing a binary blob comprised of on
e or more GumEvent structs. See `gumevent.h` for details about the format. Use `Stalker
.parse()` to examine the data.
            onReceive(events) {
              var parsedEvents = Stalker.parse(events)
              // var parsedEventsStr = JSON.stringify(parsedEventsStr)
              // console.log(">>> into onReceive: parsedEvents=" + parsedEvents + ", pa
rsedEventsStr=" + parsedEventsStr);
              console.log(">>> into onReceive: parsedEvents=" + parsedEvents);
            },


            // transform: (iterator: StalkerArm64Iterator) => {
            transform: function (iterator) {
              // https://www.radare.org/doc/frida/interfaces/StalkerArmIterator.html


              // console.log("iterator=" + iterator)
              var instruction = iterator.next()
              const startAddress = instruction.address
              // console.log("+++ into iterator: startAddress=" + startAddress)
              // const isAppCode = startAddress.compare(funcRealStartAddr) >= 0 && star
tAddress.compare(funcRealEndAddr) === -1
              // const isAppCode = (startAddress.compare(funcRealStartAddr) >= 0) && (s
tartAddress.compare(funcRealEndAddr) < 0)
              const gt_realStartAddr = startAddress.compare(funcRealStartAddr) >= 0
              const lt_realEndAddr = startAddress.compare(funcRealEndAddr) < 0
              var isAppCode = gt_realStartAddr && lt_realEndAddr
              console.log("+++ into iterator: startAddress=" + startAddress + ", isAppC
ode=" + isAppCode)


              // // for debug
              // isAppCode = true


              // console.log("isAppCode=" + isAppCode + ", gt_realStartAddr=" + gt_real
StartAddr + ", lt_realEndAddr=" + lt_realEndAddr)
              do {
                if (isAppCode) {
                  // is origal function code = which we focus on


                  // console.log("instruction: address=" + instruction.address
                  //     + ",next=" + instruction.next()
                  //     + ",size=" + instruction.size
                  //     + ",mnemonic=" + instruction.mnemonic
```

```
                    //     + ",opStr=" + instruction.opStr
                    //     + ",operands=" + JSON.stringify(instruction.operands)
                    //     + ",regsAccessed=" + JSON.stringify(instruction.regsAccessed)
                    //     + ",regsRead=" + JSON.stringify(instruction.regsRead)
                    //     + ",regsWritten=" + JSON.stringify(instruction.regsWritten)
                    //     + ",groups=" + JSON.stringify(instruction.groups)
                    //     + ",toString()=" + instruction.toString()
                    //     + ",toJSON()=" + instruction.toJSON()
                    // );


                    var curRealAddr = instruction.address
                    // console.log("curRealAddr=" + curRealAddr)
                    // const isAppCode = curRealAddr.compare(funcRealStartAddr) >= 0 && c
urRealAddr.compare(funcRealEndAddr) === -1
                    // console.log(curRealAddr + ": isAppCode=" + isAppCode)
                    var curOffsetHexPtr = curRealAddr.sub(funcRealStartAddr)
                    var curOffsetInt = curOffsetHexPtr.toInt32()
                    console.log("current: realAddr=" + curRealAddr + " -> offset: hex=" +
 curOffsetHexPtr + "=" + curOffsetInt)


                    // var instructionStr = instruction.mnemonic + " " + instruction.opStr

                    var instructionStr = instruction.toString()
                    // console.log("\t" + curRealAddr + ": " + instructionStr);
                    // console.log("\t" + curRealAddr + " <+" + curOffsetHexPtr + ">: " +
 instructionStr)
                    console.log("\t" + curRealAddr + " <+" + curOffsetInt + ">: " + instr
uctionStr)


                    if (curOffsetInt in hookFuncMap){
                      console.log("offset: " + curOffsetHexPtr + "=" + curOffsetInt)
                      // let curHookFunc = hookFuncMap.get(curOffsetInt)
                      var curHookFunc = hookFuncMap[curOffsetInt]
                      // console.log("curOffsetInt=" + curOffsetInt + " -> curHookFunc="
+ curHookFunc)


                      // putCallout -> https://www.radare.org/doc/frida/interfaces/Stalke
rArmIterator.html#putCallout
                      // StalkerScriptCallout -> https://www.radare.org/doc/frida/types/S
talkerScriptCallout.html
                      // CpuContext -> https://www.radare.org/doc/frida/types/CpuContext.
html
                      // Arm64CpuContext -> https://www.radare.org/doc/frida/interfaces/A
rm64CpuContext.html


                      // work: normal
                      iterator.putCallout(curHookFunc)


                      // var extraDataDict = {
                      //   "curOffsetInt": curOffsetInt
```

```
                    // }
                    // Not work: abnormal
                    // iterator.putCallout((context) => {
                    // // iterator.putCallout((context, extraDataDict) => {
                    //   // console.log("match offset: " + curOffsetHexPtr + ", curReal
Addr=" + curRealAddr)
                    //   // curHookFunc(context, curOffsetInt, moduleBaseAddress)
                    //   // context.curOffsetInt = curOffsetInt
                    //   // context.curOffsetHexPtr = curOffsetHexPtr
                    //   // context.moduleBaseAddress = moduleBaseAddress
                    //   // context[curOffsetInt] = curOffsetInt
                    //   // context[curOffsetHexPtr] = curOffsetHexPtr
                    //   // context[moduleBaseAddress] = moduleBaseAddress
                    //   // curHookFunc(context, extraDataDict)
                    //   curHookFunc(context)
                    // })
                  }


                }
                iterator.keep()
              } while ((instruction = iterator.next()) !== null)
            }
          });


          // function needDebug(context) {
          //     console.log("into needDebug")
          //     // console.log("into needDebug: context=" + context)
          //     // var contextStr = JSON.stringify(context, null, 2)
          //     // console.log("context=" + contextStr)
          //     // var x9Value1 = context.x9
          //     // var x9Value2 = context["x9"]
          //     // console.log("x9Value1=" + x9Value1 + ", x9Value2=" + x9Value2)
          // }
        },
        onLeave: function(retval) {
          console.log("addr: relative [" + funcRelativeStartAddrHexStr + "] real [" + fun
cRealStartAddr + "] -> retval=" + retval)
          if (curTid != null) {
            Stalker.unfollow(curTid)
            console.log("Stalker.unfollow curTid=", curTid)
          }
        }
      })
    }
```

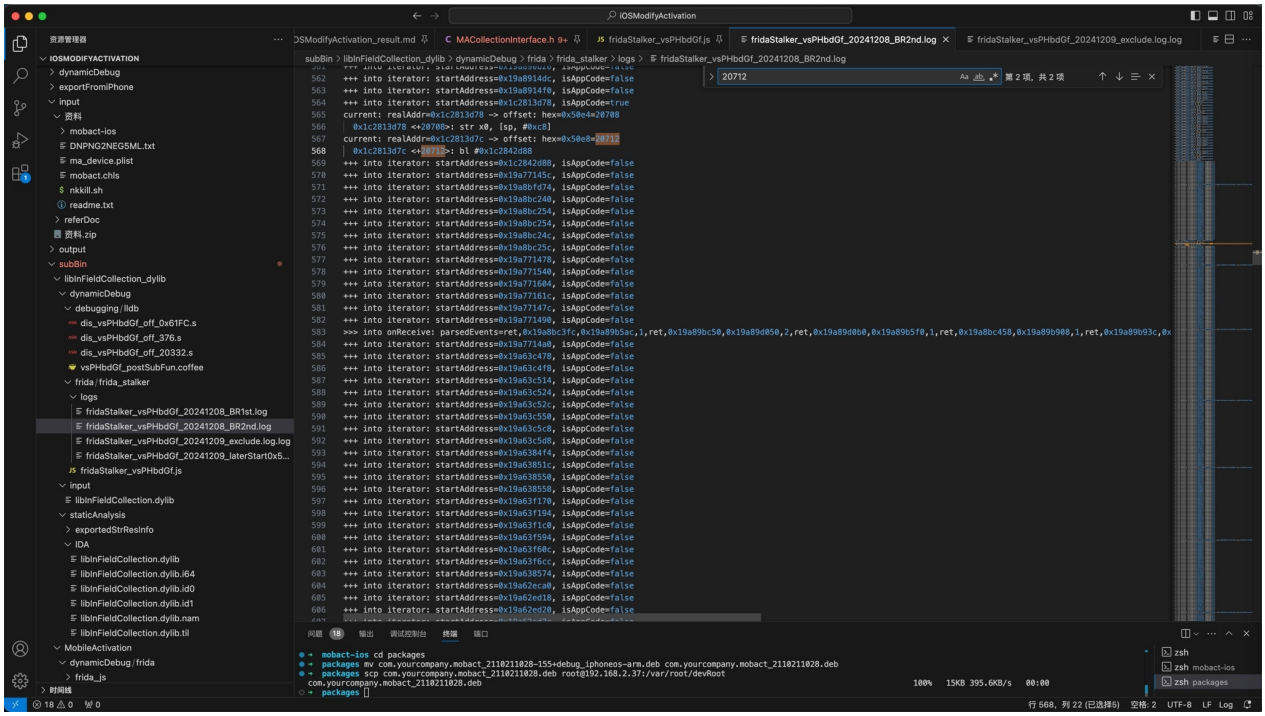
注：其最新代码详见：FridaUtil.js

但是hook不到，该函数 `vsPHbdGf` 的全部代码

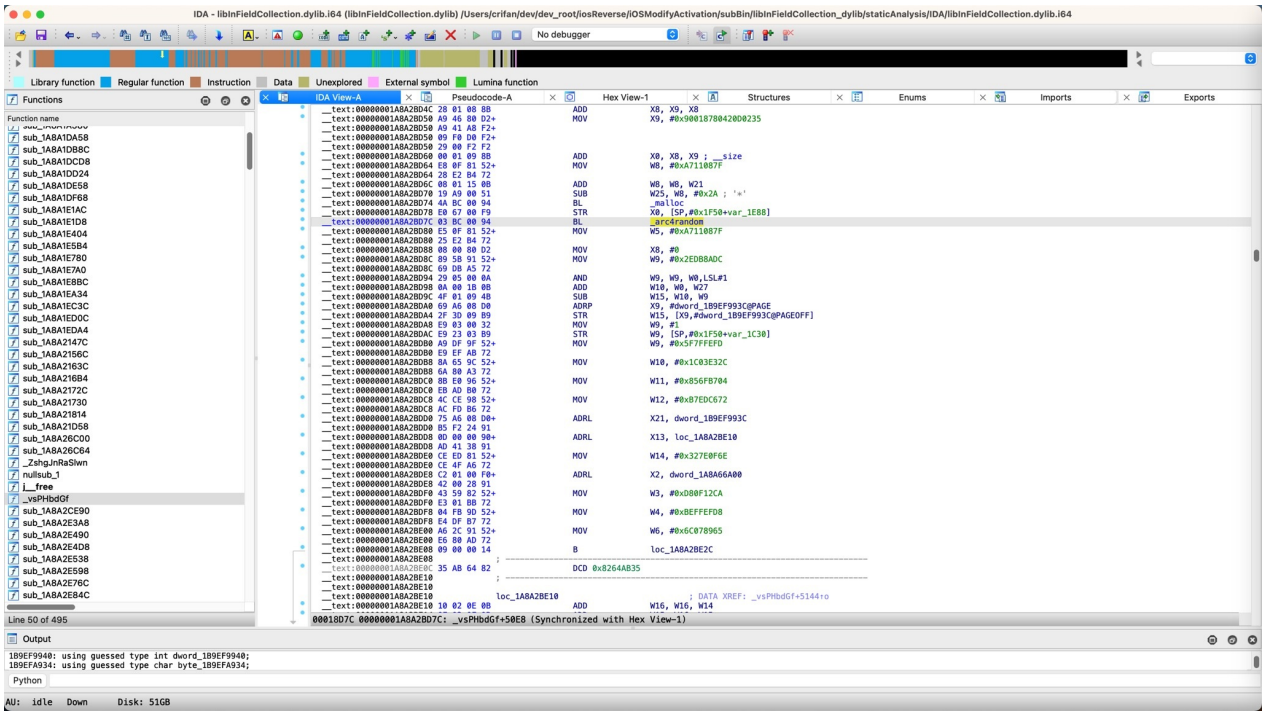
具体现象就是： vsPHbdGf函数代码很多，但是Stalker输出的代码执行逻辑，最后，最多，只到：

- `frida/frida_stalker/logs/fridaStalker_vsPHbdGf_20241208_BR2nd.log`

```
[iPhone::mobileactivationd ]-> %reload
into hookNative:
into stalkerHookNative_vsPHbdGf:
libraryName libInFieldCollection.dylib -> moduleBaseAddress=0x1c27fb000
origFuncPtr=0x1c280ec94
funcRelativeStartAddr=81044
functionSize=25084=0x61fc, argNum=2
Frida Stalker hook: module: baseAddress=0x1c27fb000
function: relativeStartAddr=0x13c94, size=25084=0x61fc, relativeEndAddr=0x19e90
funcRealStartAddr=0x1c280ec94, funcRealEndAddr=0x1c2814e90
curTid=null
[iPhone::mobileactivationd ]-> ============= Trigged addr: relative [0x13c94] = real [0x
1c280ec94] =============
arg[0]=0x16fa8a988
arg[1]=0x16fa8a984
curTid=21019
+++ into iterator: startAddress=0x10f3c1768, isAppCode=false
+++ into iterator: startAddress=0x10f3c1778, isAppCode=false
...
+++ into iterator: startAddress=0x1c280eca4, isAppCode=true
current: realAddr=0x1c280eca4 -> offset: hex=0x10=16
    0x1c280eca4 <+16>: stp x20, x19, [sp, #0x40]
...
current: realAddr=0x1c2813d74 -> offset: hex=0x50e0=20704
    0x1c2813d74 <+20704>: bl #0x1c2842e9c
+++ into iterator: startAddress=0x1c2842e9c, isAppCode=false
+++ into iterator: startAddress=0x19a8914c0, isAppCode=false
...
+++ into iterator: startAddress=0x19a8914f0, isAppCode=false
+++ into iterator: startAddress=0x1c2813d78, isAppCode=true
current: realAddr=0x1c2813d78 -> offset: hex=0x50e4=20708
    0x1c2813d78 <+20708>: str x0, [sp, #0xc8]
current: realAddr=0x1c2813d7c -> offset: hex=0x50e8=20712
    0x1c2813d7c <+20712>: bl #0x1c2842d88
+++ into iterator: startAddress=0x1c2842d88, isAppCode=false
+++ into iterator: startAddress=0x19a77145c, isAppCode=false
+++ into iterator: startAddress=0x19a8bfd74, isAppCode=false
...
```



对应着 `IDA` 中的 `D7C` 结尾的位置：



后续的，之后的代码，就看不到运行了。

原因

基本上确定就是：

由于（ `Stalker` ）去hook的内容太多，导致出现异常，即没有hook到后续代码的运行

解决办法

用Stalker.exclude排除掉其他的、无需关心的代码

具体步骤

- 先找到，要排除掉的代码的范围
  - 核心思路是：从你的函数中，真实额外调用了第三方（尤其是系统库中的函数），去寻找要排除掉的范围
  - 举例
    - 此处iOS的程序，且调试期间的确看到了后续代码会调用到 `malloc` 、 `arc4random` 等函数
      - 其中，好像是，这些函数就是属于 ObjC或System相关的库中?
    - 且此处的函数vsPHbdGf所属的库libInFieldCollection.dylib，所依赖的外部库，即import的库
      - libInFieldCollection_dylib/staticAnalysis/exportedStrResInfo/libInFieldCollection_rabin2_l_libraries.coffee
        - `/System/Library/Frameworks/IOKit.framework/Versions/A/IOKit`
        - `/System/Library/Frameworks/Foundation.framework/Foundation`
        - `/System/Library/Frameworks/CoreFoundation.framework/CoreFoundation`
        - `/usr/lib/libMobileGestalt.dylib`
        - `/usr/lib/libobjc.A.dylib`
        - `/usr/lib/libSystem.B.dylib`
    - 其中就有
      - iOS的常见的底层通用的库
        - `/usr/lib/libobjc.A.dylib`
        - `/usr/lib/libSystem.B.dylib`
    - 所以，基本上可以确定就是：
      - 此处后续要排除的范围，就是： `libobjc.A.dylib` 、 `libSystem.B.dylib` 等系统基础的底层的库
- 再去用 `Stalker.exclude` 排除掉（整个库）

然后再去代码中加上：

```
// try exclude not-concern functions inside common libs

// var moduleName_libObjc = "/usr/lib/libobjc.A.dylib"
var moduleName_libObjc = "libobjc.A.dylib"
// FridaUtil.printModuleInfo(moduleName_libObjc)
var module_libObjc = Process.getModuleByName(moduleName_libObjc)
console.log("module_libObjc=" + module_libObjc)
if (null != module_libObjc) {
  FridaUtil.printModuleBasicInfo(module_libObjc)

  Stalker.exclude(module_libObjc)
}

// var moduleName_system = "/usr/lib/libSystem.B.dylib"
var moduleName_system = "libSystem.B.dylib"
// FridaUtil.printModuleInfo(moduleName_system)
var module_system = Process.getModuleByName(moduleName_system)
console.log("module_system=" + module_system)
if (null != module_system) {
```

```
    FridaUtil.printModuleBasicInfo(module_system)

    Stalker.exclude(module_system)
  }
```

最新代码是：

```
function stalkerHookNative_vsPHbdGf(){
  console.log("into stalkerHookNative_vsPHbdGf:")

  var libraryName = "libInFieldCollection.dylib"
  const moduleBaseAddress = Module.findBaseAddress(libraryName)
  console.log("libraryName=" + libraryName + " -> moduleBaseAddress=" + moduleBaseAddre
ss)
  if (null == moduleBaseAddress) {
    console.error("Failed to find library " + libraryName)
    return
  }

  var origFuncPtr = Module.findExportByName(libraryName, "vsPHbdGf")
  console.log("origFuncPtr=" + origFuncPtr)

  var funcRelativeStartAddr = origFuncPtr - moduleBaseAddress
  console.log("funcRelativeStartAddr=" + funcRelativeStartAddr)
  // for debug
  /*
    [iPhone::mobileactivationd ]-> ============= Trigged addr: relative [0x13c94] = real
 [0x1c280ec94]  =============
    ...
    +++ into iterator: startAddress=0x1c2813d78, isAppCode=true
    current: realAddr=0x1c2813d78 -> offset: hex=0x50e4=20708
        0x1c2813d78 <+20708>: str x0, [sp, #0xc8]
    current: realAddr=0x1c2813d7c -> offset: hex=0x50e8=20712
        0x1c2813d7c <+20712>: bl #0x1c2842d88
  */
  // // use later address as start address, try to avoid only hook part code == not hoo
k all code
  // funcRelativeStartAddr = funcRelativeStartAddr + 0x50e4

  var functionSize = 0x61FC
  // for debug: make larger, tyr to support hook more code
  // var functionSize = 0x10000

  // CollectPCRT == void __fastcall vsPHbdGf(__int64 a1, __int64 a2)
  var argNum = 2
  console.log("functionSize=" + functionSize + "=" + JsUtil.intToHexStr(functionSize) +
", argNum=" + argNum)

  // try exclude not-concern functions inside common libs

  // var moduleName_libObjc = "/usr/lib/libobjc.A.dylib"
  var moduleName_libObjc = "libobjc.A.dylib"
  // FridaUtil.printModuleInfo(moduleName_libObjc)
  var module_libObjc = Process.getModuleByName(moduleName_libObjc)
  console.log("module_libObjc=" + module_libObjc)
```

```
  if (null != module_libObjc) {
    FridaUtil.printModuleBasicInfo(module_libObjc)

    Stalker.exclude(module_libObjc)
  }

  // var moduleName_system = "/usr/lib/libSystem.B.dylib"
  var moduleName_system = "libSystem.B.dylib"
  // FridaUtil.printModuleInfo(moduleName_system)
  var module_system = Process.getModuleByName(moduleName_system)
  console.log("module_system=" + module_system)
  if (null != module_system) {
    FridaUtil.printModuleBasicInfo(module_system)

    Stalker.exclude(module_system)
  }

  let hookFuncMap = {
    0x164: // +356
      function (context) {
        var contextStr = JSON.stringify(context)
        console.log("[0x164] contextStr=" + contextStr)
        var x10 = context.x10
        console.log("[0x164] x10=" + x10)
      },
    0x1DC: // +476
      function (context) {
        var contextStr = JSON.stringify(context)
        console.log("[0x1DC] contextStr=" + contextStr)
        var x10 = context.x10
        console.log("[0x1DC] x10=" + x10)
      },
  }

  FridaUtil.stalkerHookUnnameNative(moduleBaseAddress, funcRelativeStartAddr, functionS
ize, argNum, hookFuncMap)
}

function hookNative(){
  console.log("into hookNative:")
  stalkerHookNative_vsPHbdGf()
}

hookNative()
```

最开始的输出中，关于库的输出是：

```
module_libObjc [object Object]
Module: name libobjc.A.dylib, base 0x199e82000, size7897088, path /usr/lib/libobjc.A.dy
lib
module_system [object Object]
Module: name libSystem.B.dylib, base 0x199e17000, size8192, path /usr/lib/libSystem.B.d
ylib
Frida Stalker hook: module: baseAddress 0x1c27fb000
```

最终结果是：

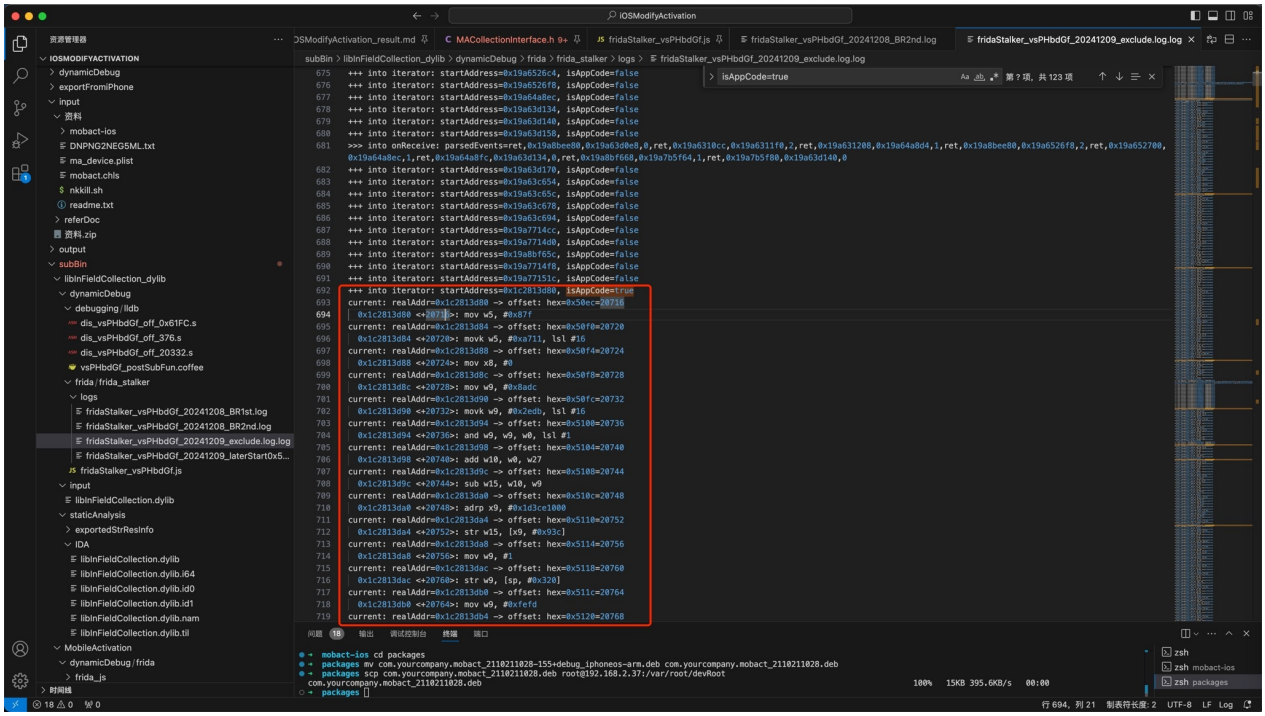
即可顺利hook出后续代码

- （ `d7c` 结尾的） `+20712` 的后续代码
  - 从（ `d80` 结尾的） `+20716` 开始的后续的代码

的执行了：

- `frida/frida_stalker/logs/fridaStalker_vsPHbdGf_20241209_exclude.log.log`

```
...
+++ into iterator: startAddress=0x19a77151c, isAppCode=false
+++ into iterator: startAddress=0x1c2813d80, isAppCode=true
current: realAddr=0x1c2813d80 -> offset: hex=0x50ec=20716
    0x1c2813d80 <+20716>: mov w5, #0x87f
current: realAddr=0x1c2813d84 -> offset: hex=0x50f0=20720
    0x1c2813d84 <+20720>: movk w5, #0xa711, lsl #16
current: realAddr=0x1c2813d88 -> offset: hex=0x50f4=20724
    0x1c2813d88 <+20724>: mov x8, #0
```



心得

刚开始以为是Frida的Stalker的bug呢，导致函数的后续代码没触发

后来也看到了，官网有提到：

Stalker | Frida • A world-class dynamic instrumentation toolkit

> **Excluded Ranges**

> Stalker also has the API `Stalker.exclude(range)` that's passed a base and limit used to prevent Stalker from instrumenting code within these regions. Consider, for example, your thread calls `malloc()` inside `libc`. You most likely don't care about the inner workings of the heap and this is not only going to slow down performance, but also generate a whole lot of extraneous events you don't care about. One thing to consider, however, is that as soon as a call is made to an excluded range, stalking of that thread is stopped until it returns. That means, if that thread were to call a function which is not inside a restricted range, a callback perhaps, then this would not be captured by Stalker. Just as this can be used to stop the stalking of a whole library, it can be used to stop stalking a given function (and its callees) too. This can be particularly useful if your target application is statically linked. Here, was cannot simply ignore all calls to `libc`, but we can find the symbol for `malloc()` using `Module.enumerateSymbols()` and ignore that single function.

以及找到了对应的API接口描述

JavaScript API | Frida • A world-class dynamic instrumentation toolkit

> **Stalker**
>
> `Stalker.exclude(range)`: marks the specified memory `range` as excluded, which is an object with `base` and `size` properties – like the properties in an object returned by e.g. `Process.getModuleByName()`.
>
> This means Stalker will not follow execution when encountering a call to an instruction in such a range. You will thus be able to observe/modify the arguments going in, and the return value coming back, but won't see the instructions that happened between.
>
> Useful to improve performance and reduce noise.

其中最后一行明显是：

- 可以用来：提高性能和降低噪音。

-》但是差点，仍旧怀疑是Stalker的bug，而没去尝试Stalker.exclude，差点错过了这个Stalker.exclude

-》实践证明：`Stalker.exclude`是可以解决此问题的

最后的心得，就是：

- 以后遇到这类，觉得有一点希望的思路，只要时间允许，还是尽量也去试试
  - 且同时也是增加技术能力的，此处即新学习了个Stalker.exclude的用法

Stalker工具类函数

此处整理，Frida的Stalker期间，整理出的，工具类函数。

stalkerHookUnnameNative

- 函数：`stalkerHookUnnameNative`
- 作用：实现Frida的Stalker去hook某个位置的代码的通用逻辑
- （调用）举例
  - libmetasec_ml.so的ms_bd_c_l_a
    - `FridaUtil.stalkerHookUnnameNative(moduleBaseAddress, funcRelativeStartAddr, functionSize, argNum, hookFuncMap)`
- 最新代码
  - 详见：FridaUtil.js

`stalkerHookUnnameNative` 的代码

```
  // Frida Stalker hoo unknown name native function
  static stalkerHookUnnameNative(moduleBaseAddress, funcRelativeStartAddr, functionSize, argNum, hookFuncMap){
    console.log("Frida Stalker hook: module: baseAddress=" + moduleBaseAddress + ", isShowOpcode=" + FridaUtil.isShowOpcode)

    var functionSizeHexStr = JsUtil.intToHexStr(functionSize)
    var funcRelativeStartAddrHexStr = JsUtil.intToHexStr(funcRelativeStartAddr)
    var funcRelativeEndAddr = funcRelativeStartAddr + functionSize
    var funcRelativeEndAddrHexStr = JsUtil.intToHexStr(funcRelativeEndAddr)
    console.log("function: relativeStartAddr=" + funcRelativeStartAddrHexStr + ", size=" + functionSize + "=" + functionSizeHexStr + ", relativeEndAddr=" + funcRelativeEndAddrHexStr)

    const funcRealStartAddr = moduleBaseAddress.add(funcRelativeStartAddr)
    // var funcRealEndAddr = funcRealStartAddr + functionSize
    const funcRealEndAddr = funcRealStartAddr.add(functionSize)
    console.log("funcRealStartAddr=" + funcRealStartAddr + ", funcRealEndAddr=" + funcRealEndAddr)
    var curTid = null
    console.log("curTid=" + curTid)
    Interceptor.attach(funcRealStartAddr, {
      onEnter: function(args) {
        JsUtil.logStr("Trigged addr: relative [" + funcRelativeStartAddrHexStr + "] = real [" + funcRealStartAddr + "]")

        for(var i = 0; i < argNum; i ++) {
          var curArg = args[i]
          console.log("arg[" + i + "]=" + curArg)
        }

        var curTid = Process.getCurrentThreadId()
        console.log("curTid=" + curTid)
        Stalker.follow(curTid, {
            events: {
              call: false, // CALL instructions: yes please
              ret: true, // RET instructions
```

```
            exec: false, // all instructions: not recommended as it's
            block: false, // block executed: coarse execution trace
            compile: false // block compiled: useful for coverage
          },
          // onReceive: Called with `events` containing a binary blob comprised of on
e or more GumEvent structs. See `gumevent.h` for details about the format. Use `Stalker
.parse()` to examine the data.
          onReceive(events) {
            var parsedEvents = Stalker.parse(events)
            // var parsedEventsStr = JSON.stringify(parsedEventsStr)
            // console.log(">>> into onReceive: parsedEvents=" + parsedEvents + ", pa
rsedEventsStr=" + parsedEventsStr);
            console.log(">>> into onReceive: parsedEvents=" + parsedEvents);
          },

          // transform: (iterator: StalkerArm64Iterator) => {
          transform: function (iterator) {
            // https://www.radare.org/doc/frida/interfaces/StalkerArmIterator.html

            // console.log("iterator=" + iterator)
            var instruction = iterator.next()
            const startAddress = instruction.address
            // console.log("+++ into iterator: startAddress=" + startAddress)
            // const isAppCode = startAddress.compare(funcRealStartAddr) >= 0 && star
tAddress.compare(funcRealEndAddr) === -1
            // const isAppCode = (startAddress.compare(funcRealStartAddr) >= 0) && (s
tartAddress.compare(funcRealEndAddr) < 0)
            const gt_realStartAddr = startAddress.compare(funcRealStartAddr) >= 0
            const lt_realEndAddr = startAddress.compare(funcRealEndAddr) < 0
            var isAppCode = gt_realStartAddr && lt_realEndAddr
            console.log("+++ into iterator: startAddress=" + startAddress + ", isAppC
ode=" + isAppCode)

            // // for debug
            // isAppCode = true

            // console.log("isAppCode=" + isAppCode + ", gt_realStartAddr=" + gt_real
StartAddr + ", lt_realEndAddr=" + lt_realEndAddr)
            do {
              if (isAppCode) {
                // is origal function code = which we focus on
                // FridaUtil.printInstructionInfo(instruction)

                var curRealAddr = instruction.address
                // console.log("curRealAddr=" + curRealAddr)
                // const isAppCode = curRealAddr.compare(funcRealStartAddr) >= 0 && c
urRealAddr.compare(funcRealEndAddr) === -1
                // console.log(curRealAddr + ": isAppCode=" + isAppCode)
                var curOffsetHexPtr = curRealAddr.sub(funcRealStartAddr)
                var curOffsetInt = curOffsetHexPtr.toInt32()
                console.log("current: realAddr=" + curRealAddr + " -> offset: hex=" +
 curOffsetHexPtr + "=" + curOffsetInt)

                // var instructionStr = instruction.mnemonic + " " + instruction.opStr

                var instructionStr = instruction.toString()
```

```
                    // console.log("\t" + curRealAddr + ": " + instructionStr);
                    // console.log("\t" + curRealAddr + " <+" + curOffsetHexPtr + ">: " +
 instructionStr)
                    // console.log("\t" + curRealAddr + " <+" + curOffsetInt + ">: " + in
structionStr)

                    var opcodeStr = ""
                    if (FridaUtil.isShowOpcode) {
                      opcodeStr = " " + FridaUtil.genInstructionOpcodeStr(instruction)
                    }
                    var instructionFullLogStr = "\t" + curRealAddr + " <+" + curOffsetInt
 + ">" + opcodeStr + ": " + instructionStr
                    console.log(instructionFullLogStr)
                    // 0x252c0edf8 <+356>: br x10
                    // 0x252c0edf8 <+356> 40 01 1F D6: br x10

                    if (curOffsetInt in hookFuncMap){
                      console.log("offset: " + curOffsetHexPtr + "=" + curOffsetInt)
                      // let curHookFunc = hookFuncMap.get(curOffsetInt)
                      var curHookFunc = hookFuncMap[curOffsetInt]
                      // console.log("curOffsetInt=" + curOffsetInt + " -> curHookFunc="
 + curHookFunc)

                      // putCallout -> https://www.radare.org/doc/frida/interfaces/Stalke
rArmIterator.html#putCallout
                      // StalkerScriptCallout -> https://www.radare.org/doc/frida/types/S
talkerScriptCallout.html
                      // CpuContext -> https://www.radare.org/doc/frida/types/CpuContext.
html
                      // Arm64CpuContext -> https://www.radare.org/doc/frida/interfaces/A
rm64CpuContext.html

                      // work: normal
                      iterator.putCallout(curHookFunc)

                      // var extraDataDict = {
                      //   "curOffsetInt": curOffsetInt
                      // }
                      // Not work: abnormal
                      // iterator.putCallout((context) => {
                      // // iterator.putCallout((context, extraDataDict) => {
                      //   // console.log("match offset: " + curOffsetHexPtr + ", curReal
Addr=" + curRealAddr)
                      //   // curHookFunc(context, curOffsetInt, moduleBaseAddress)
                      //   // context.curOffsetInt = curOffsetInt
                      //   // context.curOffsetHexPtr = curOffsetHexPtr
                      //   // context.moduleBaseAddress = moduleBaseAddress
                      //   // context[curOffsetInt] = curOffsetInt
                      //   // context[curOffsetHexPtr] = curOffsetHexPtr
                      //   // context[moduleBaseAddress] = moduleBaseAddress
                      //   // curHookFunc(context, extraDataDict)
                      //   curHookFunc(context)
                      // })
                    }

                }
```

```
                iterator.keep()
            } while ((instruction = iterator.next()) !== null)
        }
    });

    // function needDebug(context) {
    //     console.log("into needDebug")
    //     // console.log("into needDebug: context=" + context)
    //     // var contextStr = JSON.stringify(context, null, 2)
    //     // console.log("context=" + contextStr)
    //     // var x9Value1 = context.x9
    //     // var x9Value2 = context["x9"]
    //     // console.log("x9Value1=" + x9Value1 + ", x9Value2=" + x9Value2)
    // }
  },
  onLeave: function(retval) {
    console.log("addr: relative [" + funcRelativeStartAddrHexStr + "] real [" + fun
cRealStartAddr + "] -> retval=" + retval)
    if (curTid != null) {
      Stalker.unfollow(curTid)
      console.log("Stalker.unfollow curTid=", curTid)
    }
  }
})
}
```

附录

下面列出相关参考资料。

参考资料

- 【基本解决】Frida的Stalker.follow中events的含义和用法
- 【已解决】Frida中Stalker如何在Stalker.follow中写代码追踪指令代码运行
- 【已解决】Mac中如何使用frida的Stalker的js脚本
- 【未解决】Frida的stalker trace
- 【基本解决】Frida的Stalker.follow中events的含义和用法
- 【已解决】Frida去hook函数报错：Error unable to intercept function at please file a bug
- 【无法解决】Frida的Stalker中transform中的instruction是否可以获取到bytes即opcode
- 【已解决】优化Frida的Stalker的代码追踪逻辑
- 【已解决】Frida的Stalker中函数地址和指令地址匹配不上
- 【已解决】搞懂Frida的Stalker.follow的transform的调试指令运行的逻辑
- 【已解决】Frida的Stalker.follow中transform中的指令instruction
- 【记录】分析Frida的Stalker对于___lldb_unnamed_symbol2575代码追踪的输出结果
- 【记录】学习Frida官网文档Stalker搞懂基本用法和逻辑
- 【已解决】Frida的Stalker中去判断是否是函数的代码指令的逻辑
- 【整理】Frida的Stalker的利用方式
- 【未解决】尝试Frida的stalker能否修复抖音AwemeCore中函数名常量字符串
- 【整理】Frida的Stalker经验心得：用Stalker.exclude提高效率和减少干扰
-
- Stalker介绍
- Stalker
- Stalker - JavaScript API | Frida
- DefinitelyTyped/index.d.ts
- [原创] sktrace：基于 Frida Stalker 的 trace 工具-Android安全-看雪-安全社区|安全招聘|kanxue.com
- Stalker的API
- 【iOS逆向】某营业厅算法分析_小陈_InfoQ写作社区
- misc/frida-stalker-example.py at master · poxyran/misc · GitHub
- Frida Stalker 是什么？-腾讯云开发者社区-腾讯云
- 某小说App返回数据 解密分析 - 奋飞安全
- Frida Stalker 是什么？ - 奋飞安全
- Crashed when hooking too many functions · Issue #244 · frida/frida
- Frida 4.4.0 Stalker.follow() Crashes process. Win8.1 x86_64, JS Bindings · Issue #63 · frida/frida
-