Android逆向之Native层分析

yong夜 / 2019-07-16 09:08:00 / 浏览数 3667 安全技术 移动安全 顶(0) 踩(0)

---

[TOC]

# 引言

通常一些应用壳代码、算法协议什么的都会在so层，需要保护起来增加逆向难度。

所以本篇文章通过分析壳程序来熟悉一下Native层分析。

# 概述

本篇分析内容如下，主要分析libshella_3.0.0.0.so文件：

- init_array节解密JNI_OnLoad(通常so层保护就是对JNI_OnLoad进行保护，隐藏native方法注册)
- 遍历/proc/self/maps获取so文件基址

# 分析过程

## init_array节：解密操作

下面代码是init函数，作为解密JNI_OnLoad之用

### 初始化

- 这是每一个栈帧都会有的初始化操作：
  1. 保存上一个栈帧的帧地址和执行完当前栈帧后的返回地址到栈中
  2. 将当前栈帧(r11的值是栈地址)指向返回地址(LR的值是代码段地址)
  3. 压栈0x48，用来存放局部变量
  4. 参数入栈，R0、R1的值到栈顶处

```
libshella_3.0.0.0.so:763FA944 STMFD           SP!, {R11,LR}
libshella_3.0.0.0.so:763FA948 ADD             R11, SP, #4
libshella_3.0.0.0.so:763FA94C SUB             SP, SP, #0x48
libshella_3.0.0.0.so:763FA950 STR             R0, [R11,#var_48]
libshella_3.0.0.0.so:763FA954 STR             R1, [R11,#var_4C]
```

### for循环

for循环，条件判断之后，执行循环体，接着跳转到条件判断之前的位置，执行最后一个语句块，这里是var_8 -= 0x1000

```
la_3.0.0.0.so:75FFE954 STR             R1, [R11,#var_4C]
la_3.0.0.0.so:75FFE958 LDR             R3, =(sub_75FFE944 - 0x75FFE964)
la_3.0.0.0.so:75FFE95C ADD             R3, PC, R3 ; sub_75FFE944
la_3.0.0.0.so:75FFE960 BIC             R3, R3, #0xFF0
la_3.0.0.0.so:75FFE964 BIC             R3, R3, #0xF
la_3.0.0.0.so:75FFE968 STR             R3, [R11,#var_8]
la_3.0.0.0.so:75FFE96C B               loc_75FFE97C
la_3.0.0.0.so:75FFE970 ; ---------------------------------------------------------------
la_3.0.0.0.so:75FFE970
la_3.0.0.0.so:75FFE970 loc_75FFE970                            ; CODE XREF: sub_75FFE944
la_3.0.0.0.so:75FFE970 LDR             R3, [R11,#var_8]
la_3.0.0.0.so:75FFE974 SUB             R3, R3, #0x1000
la_3.0.0.0.so:75FFE978 STR             R3, [R11,#var_8]
la_3.0.0.0.so:75FFE97C
la_3.0.0.0.so:75FFE97C loc_75FFE97C                            ; CODE XREF: sub_75FFE944
la_3.0.0.0.so:75FFE97C LDR             R3, [R11,#var_8]
la_3.0.0.0.so:75FFE980 LDR             R2, [R3]
la_3.0.0.0.so:75FFE984 LDR             R3, =unk_464C457F
la_3.0.0.0.so:75FFE988 CMP             R2, R3
la_3.0.0.0.so:75FFE98C BNE             loc_75FFE970
la_3.0.0.0.so:75FFE990 MOV             R3, #0
la_3.0.0.0.so:75FFE994 STR             R3, [R11,#var_1C]
la_3.0.0.0.so:75FFE998 MOV             R3, #0
la_3.0.0.0.so:75FFE99C STR             R3, [R11,#var_20]
la_3.0.0.0.so:75FFE9A0 MOV             R3, #0
la_3.0.0.0.so:75FFE9A4 STR             R3, [R11,#var_24]
la_3.0.0.0.so:75FFE9A8 LDR             R3, [R11,#var_8]
la_3.0.0.0.so:75FFE9AC STR             R3, [R11,#var_28]
la_3.0.0.0.so:75FFE9B0 LDR             R3, [R11,#var_28]
la_3.0.0.0.so:75FFE9B4 LDR             R2, [R3,#0x1C]
la_3.0.0.0.so:75FFE9B8 LDR             R3, [R11,#var_8]
la_3.0.0.0.so:75FFE9BC ADD             R3, R2, R3
la_3.0.0.0.so:75FFE9C0 STR             R3, [R11,#var_C]
la_3.0.0.0.so:75FFE9C4 MOV             R3, #0
la_3.0.0.0.so:75FFE9C8 STR             R3, [R11,#var_10]
la_3.0.0.0.so:75FFE9CC B               loc_75FFEA94
la_3.0.0.0.so:75FFE9D0 ; ---------------------------------------------------------------
```



伪代码

```
unsigned int var_8;
for(var_8=init_array_addr & 0xFFFFF000; *(unkonwn *)var_8 != 0x464C457C; var_8 -= 0x1000)
   ;
```

初始化赋值，给变量var_8赋值libshella_3.0.0.0.so内存基址

1. 前两条指令：将init_array函数地址0x763FA944复制给R3
2. 接着用BIC将0xFFF(0xFF0+0xF)中值为1的位清除，置0。即一个&0xFFFFF000操作，得到init_array节所在libshella_3.0.0.0.so文件的内存基址0x763FA000
3. 将libshella_3.0.0.0.so内存基址保存到局部变量var_8(栈)中
4. 跳转到loc_763FA97C

```
libshella_3.0.0.0.so:763FA958 LDR            R3, =(init_array - 0x763FA964)
libshella_3.0.0.0.so:763FA95C ADD            R3, PC, R3 ; init_array
libshella_3.0.0.0.so:763FA960 BIC            R3, R3, #0xFF0
libshella_3.0.0.0.so:763FA964 BIC            R3, R3, #0xF
libshella_3.0.0.0.so:763FA968 STR            R3, [R11,#var_8]
libshella_3.0.0.0.so:763FA96C B              loc_763FA97C
```

判断，是否var_8变量存储的内存地址指向的数据是magic字段0x464C457C

1. libshella_3.0.0.0.so内存基址赋值给给R3
2. 取ELF头的前四字节，也就是magic赋值给R2

3. 将0x464C457F赋值给R3
4. 比较R2，R3，如果R2-R3！=0，则跳转到loc_763FA970

循环，将变量var_8减去0x1000

```
;■■
libshella_3.0.0.0.so:763FA970 loc_763FA970                            ; CODE XREF: init_array+48■j
libshella_3.0.0.0.so:763FA970 LDR             R3, [R11,#var_8]
libshella_3.0.0.0.so:763FA974 SUB             R3, R3, #0x1000
libshella_3.0.0.0.so:763FA978 STR             R3, [R11,#var_8]

;■■
loc_763FA97C                            ; CODE XREF: init_array+28■j
libshella_3.0.0.0.so:763FA97C LDR             R3, [R11,#var_8]
libshella_3.0.0.0.so:763FA980 LDR             R2, [R3]
libshella_3.0.0.0.so:763FA984 LDR             R3, =unk_464C457F
                           ; LDR             R3, =0x464C457F
libshella_3.0.0.0.so:763FA988 CMP             R2, R3
libshella_3.0.0.0.so:763FA98C BNE             loc_763FA970
;■■
```

- 给变量var_C赋值，这个值为程序头表的内存地址

伪代码

```
var_1C = 0;
var_20 = 0;
var_24 = 0;
var_C = *(_DWORD *)(var_8 + 0x1C) + var_8; //■■■■■■
var_10 = 0;
```

1. 对变量var_1C、var_20、var_24置0
2. 将变量var_8的值(libshella_3.0.0.0.so内存基址)赋值给var_28
3. 将libshella_3.0.0.0.so内存基址加0x1C得到的内存地址指向的值（ELF偏移0x1C的值为程序头表的偏移值）赋给R2，将var_8的值赋给R3
4. R3=R2+R3，即程序头表的内存地址赋值给R3
5. 变量var_C存储程序头表的偏移地址
6. 变量var_10置0

```
libshella_3.0.0.0.so:75FFE990 MOV             R3, #0
libshella_3.0.0.0.so:75FFE994 STR             R3, [R11,#var_1C]
libshella_3.0.0.0.so:75FFE998 MOV             R3, #0
libshella_3.0.0.0.so:75FFE99C STR             R3, [R11,#var_20]
libshella_3.0.0.0.so:75FFE9A0 MOV             R3, #0
libshella_3.0.0.0.so:75FFE9A4 STR             R3, [R11,#var_24]
libshella_3.0.0.0.so:75FFE9A8 LDR             R3, [R11,#var_8]
libshella_3.0.0.0.so:75FFE9AC STR             R3, [R11,#var_28]
libshella_3.0.0.0.so:75FFE9B0 LDR             R3, [R11,#var_28]
libshella_3.0.0.0.so:75FFE9B4 LDR             R2, [R3,#0x1C]
libshella_3.0.0.0.so:75FFE9B8 LDR             R3, [R11,#var_8]
libshella_3.0.0.0.so:75FFE9BC ADD             R3, R2, R3
libshella_3.0.0.0.so:75FFE9C0 STR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFE9C4 MOV             R3, #0
libshella_3.0.0.0.so:75FFE9C8 STR             R3, [R11,#var_10]
libshella_3.0.0.0.so:75FFE9CC B               loc_75FFEA94
libshella_3.0.0.0.so:75FFE9D0 ; ---------------------------------------------------------------------------
```

while循环

伪代码

```
var_10 = 0;
while(*(unkown *)(var_28 + 0x2c) > var_10)
{
   if(*(unkown*)var_C != 1 || *(unkown*)(var_C+0x18) != 5)
   {
       if(*(unkown*)var_C == 1 || *(unkown*)(var_C+0x18) == 6)
       {
           var_20 = *(_DWORD *)(var_C + 8) & 0xFFFFF000;
           var_24 = (*(_DWORD *)(var_C + 8) + *(_DWORD *)(var_C + 0x10) + 0xFFF) & 0xFFFFF000;
```

```
            break;
        }
    }
    else
    {
        var_1C = (*(unkown*)(var_C+0x8) + *(unkown*)(var_C+0x10) + 0xFFF) &0xFFFFF000;
    }
    *(unkown *)var10++;
    *(unkown *)var_C += 0x20;
}
```

while判断，如果var_28+0x2C指向的值(程序头表数量)大于变量var_10的值就循环

第一个if语句，判断当前程序头表类型是否为可加载类型或者程序头是否可读可执行(5h)，如果条件判断成立则跳转执行if语句块loc_75FFEA1C

第一个if语句的if语句块

 第二个if语句：判断是否当前段是可加载段或者段属性为可读可写(6h)，如果条件成立，跳出第二个if语句

 第二个if语句的if语句块：当前段的虚拟地址& 0xFFFFF000的结果赋值给var_20，下一个程序段的虚拟地址存储到var_24，跳出while循环

第一个if语句中的else语句块：当前段类型是可加载段或者是可读可执行权限，就将下一个段的虚拟地址根据0x1000对其后的值赋给变量var_1C，接着跳出if语句向下执行

var_10指向的值 + 1，var_C指向的值(var_C=var_28) + 0x20

```
;2. ■var_C■■■■■■■■(■■■■■■■)■■■■■■■■R3
;■■R3-1■■■■■■zbit(z■■■■)■■0■■R3■■■■1■■■■■■loc_75FFEA1C
;■var_C■■■■■■■■+0x18(p_flag)■■■■■■■R3
;■■R3■■■■■■5
libshella_3.0.0.0.so:75FFE9D0 loc_75FFE9D0                            ; CODE XREF: sub_75FFE944+164■j
libshella_3.0.0.0.so:75FFE9D0 LDR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFE9D4 LDR             R3, [R3]
libshella_3.0.0.0.so:75FFE9D8 CMP             R3, #1
libshella_3.0.0.0.so:75FFE9DC BNE             loc_75FFEA1C
libshella_3.0.0.0.so:75FFE9E0 LDR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFE9E4 LDR             R3, [R3,#0x18]
libshella_3.0.0.0.so:75FFE9E8 CMP             R3, #5
libshella_3.0.0.0.so:75FFE9EC BNE             loc_75FFEA1C

;■■■if■■■■else■■■
;■■■■■■■■0x8■■■■■R2(■■■■■■■),■■■■■■■■0x10■■(■■■■■■■■■)■■R3■R3 = R2+R3
;R3■■(■■■■■■■■■■)■(R3 + 0xFFF) &0xFFFFF000
;■(R3 + 0xFFF) &0xFFFFF000■■■■■■■var_1C
libshella_3.0.0.0.so:75FFE9F0 LDR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFE9F4 LDR             R2, [R3,#8]
libshella_3.0.0.0.so:75FFE9F8 LDR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFE9FC LDR             R3, [R3,#0x10]
libshella_3.0.0.0.so:75FFEA00 ADD             R3, R2, R3
libshella_3.0.0.0.so:75FFEA04 ADD             R3, R3, #0xFF0
libshella_3.0.0.0.so:75FFEA08 ADD             R3, R3, #0xF
libshella_3.0.0.0.so:75FFEA0C BIC             R3, R3, #0xFF0
libshella_3.0.0.0.so:75FFEA10 BIC             R3, R3, #0xF
libshella_3.0.0.0.so:75FFEA14 STR             R3, [R11,#var_1C]
libshella_3.0.0.0.so:75FFEA18 B               loc_75FFEA7C
libshella_3.0.0.0.so:75FFEA1C ; ---------------------------------------------------------------------------

;■■■if■■■if■■■
;■■■■■if■■■■■■■■■■8■■■■■■■■■■■■var_C■■■■■■■■1■■■■■var_C+0x18■■■■■■■■6■■■■■■■■■■■■■■if■■■■■■■loc_75FFEA7C■■■■■if■■else
libshella_3.0.0.0.so:75FFEA1C
libshella_3.0.0.0.so:75FFEA1C loc_75FFEA1C                            ; CODE XREF: sub_75FFE944+98■j
libshella_3.0.0.0.so:75FFEA1C                                         ; sub_75FFE944+A8■j
libshella_3.0.0.0.so:75FFEA1C LDR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFEA20 LDR             R3, [R3]
libshella_3.0.0.0.so:75FFEA24 CMP             R3, #1
libshella_3.0.0.0.so:75FFEA28 BNE             loc_75FFEA7C
libshella_3.0.0.0.so:75FFEA2C LDR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFEA30 LDR             R3, [R3,#0x18]
libshella_3.0.0.0.so:75FFEA34 CMP             R3, #6
libshella_3.0.0.0.so:75FFEA38 BNE             loc_75FFEA7C
```

```
;■■■if■■■if■■■
;var_C+0x8■■■■ & 0xFFFFF000■■■■■■■■■■var_20
;■■■■■■■■■■■■■■■var_24
;■■■■■■loc_75FFEAAC■■■while■■■■■■■■■break■■
libshella_3.0.0.0.so:75FFEA3C LDR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFEA40 LDR             R3, [R3,#8]
libshella_3.0.0.0.so:75FFEA44 BIC             R3, R3, #0xFF0
libshella_3.0.0.0.so:75FFEA48 BIC             R3, R3, #0xF
libshella_3.0.0.0.so:75FFEA4C STR             R3, [R11,#var_20]
libshella_3.0.0.0.so:75FFEA50 LDR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFEA54 LDR             R2, [R3,#8]
libshella_3.0.0.0.so:75FFEA58 LDR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFEA5C LDR             R3, [R3,#0x10]
libshella_3.0.0.0.so:75FFEA60 ADD             R3, R2, R3
libshella_3.0.0.0.so:75FFEA64 ADD             R3, R3, #0xFF0
libshella_3.0.0.0.so:75FFEA68 ADD             R3, R3, #0xF
libshella_3.0.0.0.so:75FFEA6C BIC             R3, R3, #0xFF0
libshella_3.0.0.0.so:75FFEA70 BIC             R3, R3, #0xF
libshella_3.0.0.0.so:75FFEA74 STR             R3, [R11,#var_24]
libshella_3.0.0.0.so:75FFEA78 B               loc_75FFEAAC
libshella_3.0.0.0.so:75FFEA7C ; ---------------------------------------------------------------------------

;3.■■■if■■■■■■■■■■■
;var_10■■■■■+1
;var_C■■■■■+0x20
libshella_3.0.0.0.so:75FFEA7C
libshella_3.0.0.0.so:75FFEA7C loc_75FFEA7C                            ; CODE XREF: sub_75FFE944+D4■j
libshella_3.0.0.0.so:75FFEA7C                                         ; sub_75FFE944+E4■j ...
libshella_3.0.0.0.so:75FFEA7C LDR             R3, [R11,#var_10]
libshella_3.0.0.0.so:75FFEA80 ADD             R3, R3, #1
libshella_3.0.0.0.so:75FFEA84 STR             R3, [R11,#var_10]
libshella_3.0.0.0.so:75FFEA88 LDR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFEA8C ADD             R3, R3, #0x20
libshella_3.0.0.0.so:75FFEA90 STR             R3, [R11,#var_C]
libshella_3.0.0.0.so:75FFEA94

;1.while■■■■■■■■■■
;■■■var_28■■(libshella_3.0.0.0.so■■■■■)■■■R3
;■var_28+0x2C■■■■■■■■■■■■■■■■■■■■■■■■R3■■■■■■■2■■■■■■■■■R2
;■■■var_10■■■R3
;■■■■■■R2-R3■■■■■■zbit■■1■R2>R3■■■■■■■■■■■
libshella_3.0.0.0.so:75FFEA94 loc_75FFEA94                            ; CODE XREF: sub_75FFE944+88■j
libshella_3.0.0.0.so:75FFEA94 LDR             R3, [R11,#var_28]
libshella_3.0.0.0.so:75FFEA98 LDRH            R3, [R3,#0x2C]
libshella_3.0.0.0.so:75FFEA9C MOV             R2, R3
libshella_3.0.0.0.so:75FFEAA0 LDR             R3, [R11,#var_10]
libshella_3.0.0.0.so:75FFEAA4 CMP             R2, R3
libshella_3.0.0.0.so:75FFEAA8 BHI             loc_75FFE9D0
```

逻辑移位

伪代码

```
char var_11 = 0x2B;
char var_12 = 0x99;
char var_13 = 0x20;
char var_14 = 0x15;
var_30 = var_8; //libshella_3.0.0.0.so■■■■
char var_29 = 0x91;
char var_2A = 0x91;
var_31  = 0x00;
var_38 = dword_76002008 >> 16;  //0x1000
var_3C = dword_76002008 << 16 >> 16; //0x2AB4
var_40 = var_3C - var_38;   //0x1AB4

mprotect(*(unkown *)var_30 + 0x1000, ((*(unkown *)var_30 / 0x1000)+1)*0x1000, 3);
```

1. 变量赋值
2. 给libshella_3.0.0.0.so内存基址往后偏移0x1000开始，长度为0x2000的内存页，赋可读可写属性

```
;1. ■■■■
;■■■■var_14■■■■■■■0x2B■■■■■0x2B992015■■■var_14
;0x2B■■■■■■■■■0xFFFFFF91■■■■■■■0x91■var_2C■■■■■,■■■■0x91■■■■■var_2C■■■■■0x91914EB0
;var_8 = var_30
;■var_34■■■■0
;■■■■■■■■■dword_76002008■■■■■16■0x1000■■■var_38
;■■■■■■■■■dword_76002008■■■■■16■■■■16■0x2AB4■■var_3C
;■var_3C-var_38■■■■0x1AB4■■■var_40
;R0 = *var_30 + *var_38
;R1 = ((*var_40 / 0x1000)+1)*0x1000■■■■■■*var_30■■■■■■■■■■■■■0x2000
;R2 = 3
libshella_3.0.0.0.so:75FFEAAC loc_75FFEAAC                            ; CODE XREF: sub_75FFE944+134■j
libshella_3.0.0.0.so:75FFEAAC MOV             R3, #0x2B
libshella_3.0.0.0.so:75FFEAB0 STRB            R3, [R11,#var_11]
libshella_3.0.0.0.so:75FFEAB4 LDRB            R3, [R11,#var_11]
libshella_3.0.0.0.so:75FFEAB8 EOR             R3, R3, #0x45
libshella_3.0.0.0.so:75FFEABC MVN             R3, R3
libshella_3.0.0.0.so:75FFEAC0 STRB            R3, [R11,#var_29]
libshella_3.0.0.0.so:75FFEAC4 LDRB            R3, [R11,#var_29]
libshella_3.0.0.0.so:75FFEAC8 STRB            R3, [R11,#var_2A]
libshella_3.0.0.0.so:75FFEACC MOV             R3, #0xFFFFFF99
libshella_3.0.0.0.so:75FFEAD0 STRB            R3, [R11,#var_12]
libshella_3.0.0.0.so:75FFEAD4 MOV             R3, #0x20
libshella_3.0.0.0.so:75FFEAD8 STRB            R3, [R11,#var_13]
libshella_3.0.0.0.so:75FFEADC MOV             R3, #0x15
libshella_3.0.0.0.so:75FFEAE0 STRB            R3, [R11,#var_14]
libshella_3.0.0.0.so:75FFEAE4 LDR             R3, [R11,#var_8]
libshella_3.0.0.0.so:75FFEAE8 STR             R3, [R11,#var_30]
libshella_3.0.0.0.so:75FFEAEC MOV             R3, #0
libshella_3.0.0.0.so:75FFEAF0 STRB            R3, [R11,#var_31]
libshella_3.0.0.0.so:75FFEAF4 LDR             R3, =(dword_76002008 - 0x75FFEB00)
libshella_3.0.0.0.so:75FFEAF8 ADD             R3, PC, R3 ; dword_76002008
libshella_3.0.0.0.so:75FFEAFC LDR             R3, [R3]
libshella_3.0.0.0.so:75FFEB00 MOV             R3, R3,LSR#16
libshella_3.0.0.0.so:75FFEB04 STR             R3, [R11,#var_38]
libshella_3.0.0.0.so:75FFEB08 LDR             R3, =(dword_76002008 - 0x75FFEB14)
libshella_3.0.0.0.so:75FFEB0C ADD             R3, PC, R3 ; dword_76002008
libshella_3.0.0.0.so:75FFEB10 LDR             R3, [R3]
libshella_3.0.0.0.so:75FFEB14 MOV             R3, R3,LSL#16
libshella_3.0.0.0.so:75FFEB18 MOV             R3, R3,LSR#16
libshella_3.0.0.0.so:75FFEB1C STR             R3, [R11,#var_3C]
libshella_3.0.0.0.so:75FFEB20 LDR             R2, [R11,#var_3C]
libshella_3.0.0.0.so:75FFEB24 LDR             R3, [R11,#var_38]
libshella_3.0.0.0.so:75FFEB28 RSB             R3, R3, R2
libshella_3.0.0.0.so:75FFEB2C STR             R3, [R11,#var_40]
libshella_3.0.0.0.so:75FFEB30 LDR             R3, [R11,#var_38]
libshella_3.0.0.0.so:75FFEB34 LDR             R2, [R11,#var_30]
libshella_3.0.0.0.so:75FFEB38 ADD             R2, R2, R3
libshella_3.0.0.0.so:75FFEB3C LDR             R3, [R11,#var_40]
libshella_3.0.0.0.so:75FFEB40 ADD             R3, R3, #0xFF0
libshella_3.0.0.0.so:75FFEB44 ADD             R3, R3, #0xF
libshella_3.0.0.0.so:75FFEB48 BIC             R3, R3, #0xFF0
libshella_3.0.0.0.so:75FFEB4C BIC             R3, R3, #0xF
libshella_3.0.0.0.so:75FFEB50 MOV             R0, R2
libshella_3.0.0.0.so:75FFEB54 MOV             R1, R3
libshella_3.0.0.0.so:75FFEB58 MOV             R2, #3
libshella_3.0.0.0.so:75FFEB5C BL              mprotect_0
```

解密

伪代码

```
for(var_18=0x1000; var_18 <= var_3C; var_18++)
{
   var_31 = *(_BYTE *)(var_30 + var_18);
   var_2A = (unsigned __int8)(((ar12-var_13) ^ var_18 + var_14) ^ var_11);
   *(_BYTE *)(var_30 + var_18) = *(_BYTE *)(var_30 + var_18) ^ (char)var_2A;
   *(_BYTE *)(var_30 + var_18) += (unsigned __int8)((var_13 &var_14) ^ var_12);
   var_11 += (unsigned __int8)(var_12 + var_13 - var_14) & var_31 & var_18;
   var_12 += (unsigned __int8)(var_18 + var_11) ^ var_31;
```

```
   var_13 ^= (unsigned __int8)(var_31-var11) ^ var_18;
   var_14 += (unsigned __int8)(var_18)-(var_31 +var_11);

   var_18++;
}

mprotect(var_38+var_30, (var_40/0x1000 + 1)*0x1000, 5);
sub_B39E47E8(var_38+var_30, var_40);
dword_AF7A5008 = var_8; //libshella_3.0.0.0.so■■■■
```

1. for循环的初始化操作：var_18 = var_38 = 0x1000
2. for循环的判断条件：var_18 <= var_3C（0x2AB4）

```
1. ;■■■
libshella_3.0.0.0.so:AF7A1B60 LDR             R3, [R11,#var_38]
libshella_3.0.0.0.so:AF7A1B64 STR             R3, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1B68 B               loc_AF7A1D24
libshella_3.0.0.0.so:AF7A1B6C ; ---------------------------------------------------------------------------
libshella_3.0.0.0.so:AF7A1B6C
libshella_3.0.0.0.so:AF7A1B6C loc_AF7A1B6C                            ; CODE XREF: sub_AF7A1944+3EC■j

;3. ■■■■
;var_30+var_18■■■■■■■■■■■■■■■var_31(var_34■■■■)
;var_12■■var_13■■■■■■■■■■■■■R2■(unsigned __int8)(var12-var_13)
;var_18■■■■■■■■■■R2■■■■■■■■■■R3■■R3■8■■■■■R2
;var_14■■■■■R2■■■■■■■■■R2
;var_11■■■■R2■■■■■■■■■■■■■■■var_2A
;var_30■■■■■var_18■■■■■■R3
;var_30■■■■■var_18■■■■■■1■■■■■■■■■■R1
;var_2A■■■■R1■■■■■■■■■■■■■■■R3■■■■■■■■■
libshella_3.0.0.0.so:AF7A1B6C LDR             R3, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1B70 LDR             R2, [R11,#var_30]
libshella_3.0.0.0.so:AF7A1B74 ADD             R3, R2, R3
libshella_3.0.0.0.so:AF7A1B78 LDRB            R3, [R3]
libshella_3.0.0.0.so:AF7A1B7C STRB            R3, [R11,#var_31]
libshella_3.0.0.0.so:AF7A1B80 LDRB            R2, [R11,#var_12]
libshella_3.0.0.0.so:AF7A1B84 LDRB            R3, [R11,#var_13]
libshella_3.0.0.0.so:AF7A1B88 RSB             R3, R3, R2
libshella_3.0.0.0.so:AF7A1B8C AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1B90 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1B94 LDR             R3, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1B98 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1B9C EOR             R3, R2, R3
libshella_3.0.0.0.so:AF7A1BA0 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1BA4 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1BA8 LDRB            R3, [R11,#var_14]
libshella_3.0.0.0.so:AF7A1BAC ADD             R3, R2, R3
libshella_3.0.0.0.so:AF7A1BB0 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1BB4 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1BB8 LDRB            R3, [R11,#var_11]
libshella_3.0.0.0.so:AF7A1BBC EOR             R3, R2, R3
libshella_3.0.0.0.so:AF7A1BC0 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1BC4 STRB            R3, [R11,#var_2A]
libshella_3.0.0.0.so:AF7A1BC8 LDR             R3, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1BCC LDR             R2, [R11,#var_30]
libshella_3.0.0.0.so:AF7A1BD0 ADD             R3, R2, R3
libshella_3.0.0.0.so:AF7A1BD4 LDR             R2, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1BD8 LDR             R1, [R11,#var_30]
libshella_3.0.0.0.so:AF7A1BDC ADD             R2, R1, R2
libshella_3.0.0.0.so:AF7A1BE0 LDRB            R1, [R2]
libshella_3.0.0.0.so:AF7A1BE4 LDRB            R2, [R11,#var_2A]
libshella_3.0.0.0.so:AF7A1BE8 EOR             R2, R1, R2
libshella_3.0.0.0.so:AF7A1BEC AND             R2, R2, #0xFF
libshella_3.0.0.0.so:AF7A1BF0 STRB            R2, [R3]

;■■■■
;var_30■■■■■var_18■■■■■■R3■■■■■■■■■■■■■■■
;var_30■■■■■var_18■■■■■■1■■■■■■■■■■R1
;var_13 & var_14■■■■■■■■■■■R0
```

```
;var_12■R0■■■■■■■■■■■■■■■R1■■■■■■■■■■■■■■■R3■■■■■■
libshella_3.0.0.0.so:AF7A1BF4 LDR             R3, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1BF8 LDR             R2, [R11,#var_30]
libshella_3.0.0.0.so:AF7A1BFC ADD             R3, R2, R3
libshella_3.0.0.0.so:AF7A1C00 LDR             R2, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1C04 LDR             R1, [R11,#var_30]
libshella_3.0.0.0.so:AF7A1C08 ADD             R2, R1, R2
libshella_3.0.0.0.so:AF7A1C0C LDRB            R1, [R2]
libshella_3.0.0.0.so:AF7A1C10 LDRB            R0, [R11,#var_13]
libshella_3.0.0.0.so:AF7A1C14 LDRB            R2, [R11,#var_14]
libshella_3.0.0.0.so:AF7A1C18 AND             R2, R0, R2
libshella_3.0.0.0.so:AF7A1C1C AND             R0, R2, #0xFF
libshella_3.0.0.0.so:AF7A1C20 LDRB            R2, [R11,#var_12]
libshella_3.0.0.0.so:AF7A1C24 EOR             R2, R0, R2
libshella_3.0.0.0.so:AF7A1C28 AND             R2, R2, #0xFF
libshella_3.0.0.0.so:AF7A1C2C ADD             R2, R1, R2
libshella_3.0.0.0.so:AF7A1C30 AND             R2, R2, #0xFF
libshella_3.0.0.0.so:AF7A1C34 STRB            R2, [R3]

;var_12■■var_13,■■■■■■■■R2
;R2■■var_14■■■■■■■■R2
;var_31■R2■■■■■■■■■■■■■■■R2
;var_18■R2■■■■■■■■■■■■■■■R2
;var_11■■R2■■■■■■■var_11
libshella_3.0.0.0.so:AF7A1C38 LDRB            R2, [R11,#var_12]
libshella_3.0.0.0.so:AF7A1C3C LDRB            R3, [R11,#var_13]
libshella_3.0.0.0.so:AF7A1C40 ADD             R3, R2, R3
libshella_3.0.0.0.so:AF7A1C44 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1C48 LDRB            R3, [R11,#var_14]
libshella_3.0.0.0.so:AF7A1C4C RSB             R3, R3, R2
libshella_3.0.0.0.so:AF7A1C50 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1C54 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1C58 LDRB            R3, [R11,#var_31]
libshella_3.0.0.0.so:AF7A1C5C AND             R3, R2, R3
libshella_3.0.0.0.so:AF7A1C60 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1C64 LDR             R3, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1C68 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1C6C AND             R3, R2, R3
libshella_3.0.0.0.so:AF7A1C70 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1C74 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1C78 LDRB            R3, [R11,#var_11]
libshella_3.0.0.0.so:AF7A1C7C ADD             R3, R2, R3
libshella_3.0.0.0.so:AF7A1C80 STRB            R3, [R11,#var_11]

;var_18■■■■■■■■var_11■■■■■■■■R2, (unsigned __int8)(var_18 + var_11)
;var_31■R2■■■■■■■■■■■■var_12,■■■■■■var_12■var_12 += (unsigned __int8)(var_18 + var_11) ^ var_31
libshella_3.0.0.0.so:AF7A1C84 LDR             R3, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1C88 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1C8C LDRB            R3, [R11,#var_11]
libshella_3.0.0.0.so:AF7A1C90 ADD             R3, R2, R3
libshella_3.0.0.0.so:AF7A1C94 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1C98 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1C9C LDRB            R3, [R11,#var_31]
libshella_3.0.0.0.so:AF7A1CA0 EOR             R3, R2, R3
libshella_3.0.0.0.so:AF7A1CA4 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1CA8 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1CAC LDRB            R3, [R11,#var_12]
libshella_3.0.0.0.so:AF7A1CB0 ADD             R3, R2, R3
libshella_3.0.0.0.so:AF7A1CB4 STRB            R3, [R11,#var_12]

;var_13 ^= (unsigned __int8)(var_31-var11) ^ var_18
libshella_3.0.0.0.so:AF7A1CB8 LDRB            R2, [R11,#var_31]
libshella_3.0.0.0.so:AF7A1CBC LDRB            R3, [R11,#var_11]
libshella_3.0.0.0.so:AF7A1CC0 RSB             R3, R3, R2
libshella_3.0.0.0.so:AF7A1CC4 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1CC8 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1CCC LDR             R3, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1CD0 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1CD4 EOR             R3, R2, R3
```

```
libshella_3.0.0.0.so:AF7A1CD8 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1CDC LDRB            R3, [R11,#var_13]
libshella_3.0.0.0.so:AF7A1CE0 EOR             R3, R2, R3
libshella_3.0.0.0.so:AF7A1CE4 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1CE8 STRB            R3, [R11,#var_13]

;var_14 += (unsigned __int8)(var_18)-(var_31 +var_11)
libshella_3.0.0.0.so:AF7A1CEC LDR             R3, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1CF0 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1CF4 LDRB            R1, [R11,#var_31]
libshella_3.0.0.0.so:AF7A1CF8 LDRB            R3, [R11,#var_11]
libshella_3.0.0.0.so:AF7A1CFC ADD             R3, R1, R3
libshella_3.0.0.0.so:AF7A1D00 AND             R3, R3, #0xFF
libshella_3.0.0.0.so:AF7A1D04 RSB             R3, R3, R2
libshella_3.0.0.0.so:AF7A1D08 AND             R2, R3, #0xFF
libshella_3.0.0.0.so:AF7A1D0C LDRB            R3, [R11,#var_14]
libshella_3.0.0.0.so:AF7A1D10 ADD             R3, R2, R3
libshella_3.0.0.0.so:AF7A1D14 STRB            R3, [R11,#var_14]

;var_18++
libshella_3.0.0.0.so:AF7A1D18 LDR             R3, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1D1C ADD             R3, R3, #1
libshella_3.0.0.0.so:AF7A1D20 STR             R3, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1D24
libshella_3.0.0.0.so:AF7A1D24 loc_AF7A1D24                            ; CODE XREF: sub_AF7A1944+224■j

;2.for■■■■■■
;var_18-var_3C■■■■■■zbit■1■■■(cbit xor vbit)■1■■■■■■■■■■■■
libshella_3.0.0.0.so:AF7A1D24 LDR             R2, [R11,#var_18]
libshella_3.0.0.0.so:AF7A1D28 LDR             R3, [R11,#var_3C]
libshella_3.0.0.0.so:AF7A1D2C CMP             R2, R3
libshella_3.0.0.0.so:AF7A1D30 BLE             loc_AF7A1B6C

;4.■■
;■■■■■■■■■■■■■■■■■■■■■■■■■■■■mprotect(var_38+var_30, (var_40/0x1000 + 1)*0x1000, 5)
libshella_3.0.0.0.so:AF7A1D34 LDR             R3, [R11,#var_38]
libshella_3.0.0.0.so:AF7A1D38 LDR             R2, [R11,#var_30]
libshella_3.0.0.0.so:AF7A1D3C ADD             R2, R2, R3
libshella_3.0.0.0.so:AF7A1D40 LDR             R3, [R11,#var_40]
libshella_3.0.0.0.so:AF7A1D44 ADD             R3, R3, #0xFF0
libshella_3.0.0.0.so:AF7A1D48 ADD             R3, R3, #0xF
libshella_3.0.0.0.so:AF7A1D4C BIC             R3, R3, #0xFF0
libshella_3.0.0.0.so:AF7A1D50 BIC             R3, R3, #0xF
libshella_3.0.0.0.so:AF7A1D54 MOV             R0, R2
libshella_3.0.0.0.so:AF7A1D58 MOV             R1, R3
libshella_3.0.0.0.so:AF7A1D5C MOV             R2, #5
libshella_3.0.0.0.so:AF7A1D60 BL              mprotect_0

;■■■■■■■■■■■■■■■■■sub_B39E47E8(var_38+var_30, var_40)
libshella_3.0.0.0.so:AF7A1D64 LDR             R3, [R11,#var_38]
libshella_3.0.0.0.so:AF7A1D68 LDR             R2, [R11,#var_30]
libshella_3.0.0.0.so:AF7A1D6C ADD             R2, R2, R3
libshella_3.0.0.0.so:AF7A1D70 LDR             R3, [R11,#var_40]
libshella_3.0.0.0.so:AF7A1D74 MOV             R0, R2
libshella_3.0.0.0.so:AF7A1D78 MOV             R1, R3
libshella_3.0.0.0.so:AF7A1D7C BL              sub_B39E47E8

;■var_8■■■■dword_AF7A5008■■■
libshella_3.0.0.0.so:AF7A1D80 LDR             R2, [R11,#var_8]
libshella_3.0.0.0.so:AF7A1D84 LDR             R3, =(dword_AF7A5008 - 0xAF7A1D90)
libshella_3.0.0.0.so:AF7A1D88 ADD             R3, PC, R3 ; dword_AF7A5008
libshella_3.0.0.0.so:AF7A1D8C STR             R2, [R3]
libshella_3.0.0.0.so:AF7A1D90 BL              unk_AF7A1898
libshella_3.0.0.0.so:AF7A1D94 SUB             SP, R11, #4
libshella_3.0.0.0.so:AF7A1D98 LDMFD           SP!, {R11,PC}
libshella_3.0.0.0.so:AF7A1D98 ; End of function sub_AF7A1944
```

遍历/proc/self/maps获取so库基址

初始化

1. 保存上一个栈的栈帧，并将R11指向栈底的返回地址
2. 参数入栈

```
libshella_3.0.0.0.so:A54EF5B4 00 48 2D E9 STMFD           SP!, {R11,LR}
libshella_3.0.0.0.so:A54EF5B8 04 B0 8D E2 ADD             R11, SP, #4
libshella_3.0.0.0.so:A54EF5BC 83 DE 4D E2 SUB             SP, SP, #0x830
libshella_3.0.0.0.so:A54EF5C0 18 08 0B E5 STR             R0, [R11,#var_818]
```

文件操作，开辟内存

```
;■■■■■■■■■■■■■■■■■■■■var_8■
libshella_3.0.0.0.so:A54EF5C4 70 31 9F E5 LDR             R3, =(aProcSelfMaps - 0xA54EF5D0)
libshella_3.0.0.0.so:A54EF5C8 03 30 8F E0 ADD             R3, PC, R3              ; "/proc/self/maps"
libshella_3.0.0.0.so:A54EF5CC 03 00 A0 E1 MOV             R0, R3
libshella_3.0.0.0.so:A54EF5D0 68 31 9F E5 LDR             R3, =(unk_A54EFA5C - 0xA54EF5DC)
libshella_3.0.0.0.so:A54EF5D4 03 30 8F E0 ADD             R3, PC, R3 ; unk_A54EFA5C
libshella_3.0.0.0.so:A54EF5D8 03 10 A0 E1 MOV             R1, R3
libshella_3.0.0.0.so:A54EF5DC 61 F8 FF EB BL              fopen
libshella_3.0.0.0.so:A54EF5E0 08 00 0B E5 STR             R0, [R11,#var_8]

;■■■■■■■■■■■■■■■■■■■0x400■■■■■■■
;■■■■4■■■■■■■■■■■■■■■8■■■■■■■■(var_8,var_C)
libshella_3.0.0.0.so:A54EF5E4 01 3B 4B E2 SUB             R3, R11, #-var_400
libshella_3.0.0.0.so:A54EF5E8 04 30 43 E2 SUB             R3, R3, #4
libshella_3.0.0.0.so:A54EF5EC 08 30 43 E2 SUB             R3, R3, #8
libshella_3.0.0.0.so:A54EF5F0 01 2B A0 E3 MOV             R2, #0x400
libshella_3.0.0.0.so:A54EF5F4 03 00 A0 E1 MOV             R0, R3
libshella_3.0.0.0.so:A54EF5F8 00 10 A0 E3 MOV             R1, #0
libshella_3.0.0.0.so:A54EF5FC 32 F8 FF EB BL              memset_0

;■■■■■■■■■■■0x400■■■■■■■
libshella_3.0.0.0.so:A54EF600 02 3B 4B E2 SUB             R3, R11, #-var_800
libshella_3.0.0.0.so:A54EF604 04 30 43 E2 SUB             R3, R3, #4
libshella_3.0.0.0.so:A54EF608 08 30 43 E2 SUB             R3, R3, #8
libshella_3.0.0.0.so:A54EF60C 01 2B A0 E3 MOV             R2, #0x400
libshella_3.0.0.0.so:A54EF610 03 00 A0 E1 MOV             R0, R3
libshella_3.0.0.0.so:A54EF614 00 10 A0 E3 MOV             R1, #0
libshella_3.0.0.0.so:A54EF618 2B F8 FF EB BL              memset_0

;■var_810■var_814■0
;■dword_A54F1008■■■■■var_C
libshella_3.0.0.0.so:A54EF61C 00 30 A0 E3 MOV             R3, #0
libshella_3.0.0.0.so:A54EF620 10 38 0B E5 STR             R3, [R11,#var_810]
libshella_3.0.0.0.so:A54EF624 00 30 A0 E3 MOV             R3, #0
libshella_3.0.0.0.so:A54EF628 14 38 0B E5 STR             R3, [R11,#var_814]
libshella_3.0.0.0.so:A54EF62C 10 31 9F E5 LDR             R3, =(dword_A54F1008 - 0xA54EF638)
libshella_3.0.0.0.so:A54EF630 03 30 8F E0 ADD             R3, PC, R3 ; dword_A54F1008
libshella_3.0.0.0.so:A54EF634 00 30 93 E5 LDR             R3, [R3]
libshella_3.0.0.0.so:A54EF638 0C 30 0B E5 STR             R3, [R11,#var_C]
libshella_3.0.0.0.so:A54EF63C 33 00 00 EA B               loc_A54EF710
```

while循环

```
libshella_3.0.0.0.so:A54EF640             ; ---------------------------------------------------------------------------
libshella_3.0.0.0.so:A54EF640
libshella_3.0.0.0.so:A54EF640             loc_A54EF640                            ; CODE XREF: sub_A54EF5B4+16C■j

;2.while■■■
;■■■■■■■■■■■■■0x400■■■■■■■■■■■■■■■fgets■■■■■■■■■■■■■■■
libshella_3.0.0.0.so:A54EF640 01 3B 4B E2 SUB             R3, R11, #-var_400
libshella_3.0.0.0.so:A54EF644 04 30 43 E2 SUB             R3, R3, #4
libshella_3.0.0.0.so:A54EF648 08 30 43 E2 SUB             R3, R3, #8
libshella_3.0.0.0.so:A54EF64C 03 00 A0 E1 MOV             R0, R3
libshella_3.0.0.0.so:A54EF650 01 1B A0 E3 MOV             R1, #0x400
libshella_3.0.0.0.so:A54EF654 08 20 1B E5 LDR             R2, [R11,#var_8]
libshella_3.0.0.0.so:A54EF658 45 F8 FF EB BL              fgets
```

```
;■var_40C■■■■R2
;■var_810■■■■R3
;■var_814■■■■R12
;■var_80C■■■■var_834(SP)■var_830(SP+4)■var_82C(SP+8)■var_828(SP+C)
;■var_818■■var_824
;■■sscanf(var_40C, "%lx-%lx %s %s %s %s %s", &var_810, &var_814, var_834, var_830, var_82C, var_828, var_824);
libshella_3.0.0.0.so:A54EF65C 01 2B 4B E2 SUB             R2, R11, #-var_400
libshella_3.0.0.0.so:A54EF660 04 20 42 E2 SUB             R2, R2, #4
libshella_3.0.0.0.so:A54EF664 08 20 42 E2 SUB             R2, R2, #8
libshella_3.0.0.0.so:A54EF668 02 3B 4B E2 SUB             R3, R11, #-var_800
libshella_3.0.0.0.so:A54EF66C 04 30 43 E2 SUB             R3, R3, #4
libshella_3.0.0.0.so:A54EF670 0C 30 43 E2 SUB             R3, R3, #0xC
libshella_3.0.0.0.so:A54EF674 81 CE 4B E2 SUB             R12, R11, #-var_810
libshella_3.0.0.0.so:A54EF678 04 C0 4C E2 SUB             R12, R12, #4
libshella_3.0.0.0.so:A54EF67C 02 1B 4B E2 SUB             R1, R11, #-var_800
libshella_3.0.0.0.so:A54EF680 04 10 41 E2 SUB             R1, R1, #4
libshella_3.0.0.0.so:A54EF684 08 10 41 E2 SUB             R1, R1, #8
libshella_3.0.0.0.so:A54EF688 00 10 8D E5 STR             R1, [SP,#0x834+var_834]
libshella_3.0.0.0.so:A54EF68C 02 1B 4B E2 SUB             R1, R11, #-var_800
libshella_3.0.0.0.so:A54EF690 04 10 41 E2 SUB             R1, R1, #4
libshella_3.0.0.0.so:A54EF694 08 10 41 E2 SUB             R1, R1, #8
libshella_3.0.0.0.so:A54EF698 04 10 8D E5 STR             R1, [SP,#0x834+var_830]
libshella_3.0.0.0.so:A54EF69C 02 1B 4B E2 SUB             R1, R11, #-var_800
libshella_3.0.0.0.so:A54EF6A0 04 10 41 E2 SUB             R1, R1, #4
libshella_3.0.0.0.so:A54EF6A4 08 10 41 E2 SUB             R1, R1, #8
libshella_3.0.0.0.so:A54EF6A8 08 10 8D E5 STR             R1, [SP,#0x834+var_82C]
libshella_3.0.0.0.so:A54EF6AC 02 1B 4B E2 SUB             R1, R11, #-var_800
libshella_3.0.0.0.so:A54EF6B0 04 10 41 E2 SUB             R1, R1, #4
libshella_3.0.0.0.so:A54EF6B4 08 10 41 E2 SUB             R1, R1, #8
libshella_3.0.0.0.so:A54EF6B8 0C 10 8D E5 STR             R1, [SP,#0x834+var_828]
libshella_3.0.0.0.so:A54EF6BC 18 18 1B E5 LDR             R1, [R11,#var_818]
libshella_3.0.0.0.so:A54EF6C0 10 10 8D E5 STR             R1, [SP,#0x834+var_824]
libshella_3.0.0.0.so:A54EF6C4 02 00 A0 E1 MOV             R0, R2
libshella_3.0.0.0.so:A54EF6C8 78 20 9F E5 LDR             R2, =(aLxLxSSSSS - 0xA54EF6D4)
libshella_3.0.0.0.so:A54EF6CC 02 20 8F E0 ADD             R2, PC, R2              ; "%lx-%lx %s %s %s %s %s"
libshella_3.0.0.0.so:A54EF6D0 02 10 A0 E1 MOV             R1, R2
libshella_3.0.0.0.so:A54EF6D4 03 20 A0 E1 MOV             R2, R3
libshella_3.0.0.0.so:A54EF6D8 0C 30 A0 E1 MOV             R3, R12
libshella_3.0.0.0.so:A54EF6DC 27 F8 FF EB BL              sscanf

;■■■■■■■■■■■
libshella_3.0.0.0.so:A54EF6E0 10 28 1B E5 LDR             R2, [R11,#var_810]
libshella_3.0.0.0.so:A54EF6E4 0C 30 1B E5 LDR             R3, [R11,#var_C]
libshella_3.0.0.0.so:A54EF6E8 03 00 52 E1 CMP             R2, R3
libshella_3.0.0.0.so:A54EF6EC 07 00 00 8A BHI             loc_A54EF710
libshella_3.0.0.0.so:A54EF6F0 14 38 1B E5 LDR             R3, [R11,#var_814]
libshella_3.0.0.0.so:A54EF6F4 0C 20 1B E5 LDR             R2, [R11,#var_C]
libshella_3.0.0.0.so:A54EF6F8 03 00 52 E1 CMP             R2, R3
libshella_3.0.0.0.so:A54EF6FC 03 00 00 2A BCS             loc_A54EF710
libshella_3.0.0.0.so:A54EF700 08 00 1B E5 LDR             R0, [R11,#var_8]
libshella_3.0.0.0.so:A54EF704 20 F8 FF EB BL              unk_A54ED78C
libshella_3.0.0.0.so:A54EF708 00 30 A0 E3 MOV             R3, #0
libshella_3.0.0.0.so:A54EF70C 07 00 00 EA B               loc_A54EF730
libshella_3.0.0.0.so:A54EF710             ; ---------------------------------------------------------------------------
libshella_3.0.0.0.so:A54EF710
libshella_3.0.0.0.so:A54EF710             loc_A54EF710                            ; CODE XREF: sub_A54EF5B4+88■j
libshella_3.0.0.0.so:A54EF710                                                     ; sub_A54EF5B4+138■j ...

;1.while■■■■
;■■var_8■■■■■■■■■■■■■■■■■■
libshella_3.0.0.0.so:A54EF710 08 00 1B E5 LDR             R0, [R11,#var_8]
libshella_3.0.0.0.so:A54EF714 1F F8 FF EB BL              feof
libshella_3.0.0.0.so:A54EF718 00 30 A0 E1 MOV             R3, R0
libshella_3.0.0.0.so:A54EF71C 00 00 53 E3 CMP             R3, #0
libshella_3.0.0.0.so:A54EF720 C6 FF FF 0A BEQ             loc_A54EF640
libshella_3.0.0.0.so:A54EF724 08 00 1B E5 LDR             R0, [R11,#var_8]
libshella_3.0.0.0.so:A54EF728 17 F8 FF EB BL              fclose
libshella_3.0.0.0.so:A54EF72C 00 30 E0 E3 MOV             R3, #0xFFFFFFFF
libshella_3.0.0.0.so:A54EF730
```

```
libshella_3.0.0.0.so:A54EF730             loc_A54EF730                            ; CODE XREF: sub_A54EF5B4+158■j
libshella_3.0.0.0.so:A54EF730 03 00 A0 E1 MOV             R0, R3
libshella_3.0.0.0.so:A54EF734 04 D0 4B E2 SUB             SP, R11, #4
libshella_3.0.0.0.so:A54EF738 00 88 BD E8 LDMFD           SP!, {R11,PC}
libshella_3.0.0.0.so:A54EF738             ; End of function sub_A54EF5B4
```

伪代码

```
var_8 = fopen("/proc/self/maps", "r");
memset(&var_40C ,0, 0x400);
memset(&var_80C ,0, 0x400);
var_C = dword_A54F1008; //libshella_3.0.0.0.so■■■■■■■■■0x1000■■■■■■■■■■■■■■■

while(feof(var_8)==0)
{
   fgets(&var_40C, 0x400; var_8);
   sscanf(var_40C, "%lx-%lx %s %s %s %s %s", &var_810, &var_814, var_834, var_830, var_82C, var_828, var_824);
   if(var_810>var_C || var_814 <= var_C)
   {
       fclose(var_8);
       return 0;
   }
}
fclose(var_8);
return 1;
```

## 解密section

### 初始化

 参数入栈：so路径，一个常量

```
STMFD           SP!, {R11,LR}
ADD             R11, SP, #4
SUB             SP, SP, #0x22C0
SUB             SP, SP, #0x28
LDR             R3, =0xFFFFDD2C         ;0x22d4
SUB             R2, R11, #-var_4
STR             R0, [R2,R3]
LDR             R3, =0xFFFFDD28
SUB             R12, R11, #-var_4
STR             R1, [R12,R3]
```

### 打印sdk_version

```
SUB             R2, R11, #-var_A0
MOV             R3, #0x40
MOV             R0, R2
MOV             R1, #0
MOV             R2, R3
BL              memset_0
SUB             R3, R11, #-var_A0
LDR             R2, =(aRo_build_versi - 0xB391868C)
ADD             R2, PC, R2 ; "ro.build.version.sdk"
MOV             R0, R2
MOV             R1, R3
BL              property_get
SUB             R3, R11, #-var_A0
MOV             R0, R3
BL              atoi
MOV             R3, R0
MOV             R2, R3
LDR             R3, =(dword_B391B004 - 0xB39186B4)
ADD             R3, PC, R3 ; dword_B391B004
STR             R2, [R3]
LDR             R3, =(dword_B391B004 - 0xB39186C0)
ADD             R3, PC, R3 ; dword_B391B004
LDR             R3, [R3]
MOV             R0, #6
```

```
LDR             R2, =(aTxtag - 0xB39186D0)
ADD             R2, PC, R2 ; "txtag"
MOV             R1, R2
LDR             R2, =(aVersionD - 0xB39186DC)
ADD             R2, PC, R2 ; "version:%d"
BL              printf_log
```

do-while，打开本地so文件

```
loc_B39186DC
LDR             R3, =0xFFFFDD2C
SUB             R1, R11, #-var_4
LDR             R0, [R1,R3]
MOV             R1, #0x80000
BL              open
STR             R0, [R11,#var_28]
LDR             R3, [R11,#var_28]
CMN             R3, #1
BEQ             loc_B39186DC
```

for循环

```
SUB             R2, R11, #-var_F8
MOV             R3, #0x58
MOV             R0, R2
MOV             R1, #0
MOV             R2, R3
BL              memset_0
MOV             R3, #0
STR             R3, [R11,#var_8]
B               loc_B391874C
; ---------------------------------------------------------------------------
loc_B3918724
SUB             R2, R11, #-var_F8
LDR             R3, =0xFFFFDD28
LDR             R0, [R11,#var_28]
MOV             R1, R2
MOV             R2, #0x58
SUB             R12, R11, #-var_4
LDR             R3, [R12,R3]
BL              sub_B39176FC
MOV             R3, R0
STR             R3, [R11,#var_8]

loc_B391874C
LDR             R3, [R11,#var_8]
CMP             R3, #0x57
BLS             loc_B3918724
```

函数调用约定，参数传递

参数小于4个，用寄存器R1-R3存储并传入函数

参数大于4个，多余的参数传到栈顶

```
LDR             R2, [R11,#var_8]
LDR             R3, =0xFFFFDD2C
LDR             R1, =0xFFFFDD28
SUB             R0, R11, #-var_4
LDR             R1, [R0,R1]
STR             R1, [SP,#0x22EC+var_22EC]
STR             R2, [SP,#0x22EC+var_22E8]
MOV             R0, #6
LDR             R2, =(aTxtag - 0xB3918784)
ADD             R2, PC, R2 ; "txtag"
MOV             R1, R2
LDR             R2, =(aLoadLibrarySAt - 0xB3918790)
ADD             R2, PC, R2 ; "load library %s at offset %x read count"...
SUB             R12, R11, #-var_4
LDR             R3, [R12,R3]
```

```
BL              printf_log
SUB             R3, R11, #-var_29C
MOV             R0, R3
MOV             R1, #0
MOV             R2, #0x1A0
BL              memset_0
LDR             R3, [R11,#var_F8]
LDR             R2, [R11,#var_F4]
STR             R2, [SP,#0x22EC+var_22EC]
MOV             R0, #6
LDR             R2, =(aTxtag - 0xB39187C8)
ADD             R2, PC, R2 ; "txtag"
MOV             R1, R2
LDR             R2, =(aMin_vaddrXSize - 0xB39187D4)
ADD             R2, PC, R2 ; "min_vaddr:%x size:%x\n"
BL              printf_log
```

do-while+if

```
LDR             R3, [R11,#var_F8]
STR             R3, [R11,#var_2C]

loc_B39187DC
LDR             R3, [R11,#var_F4]
MOV             R2, #0xFFFFFFFF
STR             R2, [SP,#0x22EC+var_22EC]
MOV             R2, #0
STR             R2, [SP,#0x22EC+var_22E8]
LDR             R0, [R11,#var_2C]
MOV             R1, R3
MOV             R2, #0
MOV             R3, #0x22
BL              sub_B3917708
STR             R0, [R11,#var_C]
LDR             R2, [R11,#var_C]
LDR             R3, =0x457FFFFF
CMP             R2, R3
BHI             loc_B391886C
LDR             R2, [R11,#var_C]
LDR             R3, [R11,#var_F4]
RSB             R3, R3, #0x40000000
CMP             R2, R3
BLS             loc_B391886C
LDR             R3, =(dword_B391B004 - 0xB3918838)
ADD             R3, PC, R3 ; dword_B391B004
LDR             R3, [R3]
CMP             R3, #0xA
BHI             loc_B391886C
MOV             R0, #6
LDR             R3, =(aTxtag - 0xB3918850)
ADD             R3, PC, R3 ; "txtag"
MOV             R1, R3
LDR             R3, =(aAddrP - 0xB391885C)
ADD             R3, PC, R3 ; "addr:%p"
MOV             R2, R3
LDR             R3, [R11,#var_C]
BL              printf_log
MOV             R3, #0xFFFFFFFF
STR             R3, [R11,#var_C]

loc_B391886C
LDR             R3, [R11,#var_C]
CMN             R3, #1
BEQ             loc_B39187DC
```

malloc出0x30空间

```
LDR             R2, [R11,#var_C]
LDR             R3, [R11,#int_0]
```

```
RSB             R3, R3, R2
STR             R3, [R11,#var_30]
LDR             R3, [R11,#var_C]
STR             R3, [R11,#pMap]
LDR             R3, [R11,#var_30]
STR             R3, [R11,#var_180]
LDR             R2, [R11,#var_E8]
LDR             R3, [R11,#var_30]
ADD             R3, R2, R3
STR             R3, [R11,#var_1F0]
LDR             R2, [R11,#var_E4]
LDR             R3, [R11,#var_30]
ADD             R3, R2, R3
STR             R3, [R11,#var_1EC]
LDR             R3, [R11,#var_E0]
CMP             R3, #0
BEQ             loc_B39188D4
LDR             R2, [R11,#var_E0]
LDR             R3, [R11,#var_30]
ADD             R3, R2, R3
B               loc_B39188D8
; ---------------------------------------------------------------------------
loc_B39188D4
MOV             R3, #0
loc_B39188FC
STR             R3, [R11,#var_1BC]
LDRH            R3, [R11,#var_CE]
STR             R3, [R11,#var_1B8]
LDR             R3, [R11,#var_C4]
STR             R3, [R11,#var_1E8]
LDR             R2, [R11,#var_BC]
LDR             R3, [R11,#var_30]
ADD             R3, R2, R3
STR             R3, [R11,#var_1E0]
LDR             R3, [R11,#var_C0]
STR             R3, [R11,#var_1E4]
LDR             R2, [R11,#var_BC]
LDR             R3, [R11,#var_30]
ADD             R2, R2, R3
LDR             R3, [R11,#var_C4]
MOV             R3, R3,LSL#2
ADD             R3, R2, R3
STR             R3, [R11,#var_1DC]
LDR             R3, [R11,#var_A8]
STR             R3, [R11,#var_1A4]
LDRH            R3, [R11,#var_A4]
STR             R3, [R11,#var_1A0]
LDR             R2, [R11,#var_B8]
LDR             R3, [R11,#var_30]
ADD             R3, R2, R3
STR             R3, [R11,#var_1D4]
LDR             R3, [R11,#var_B4]
STR             R3, [R11,#var_1D0]
LDR             R2, [R11,#var_AC]
LDR             R3, [R11,#var_30]
ADD             R3, R2, R3
STR             R3, [R11,#var_1CC]
LDR             R3, [R11,#var_B0]
STR             R3, [R11,#var_1C8]
LDR             R3, [R11,#var_30]
LDR             R2, [R11,#var_C]
STR             R2, [SP,#0x22EC+var_22EC]
MOV             R0, #6
LDR             R2, =(aTxtag - 0xB39189A0)
ADD             R2, PC, R2 ; "txtag"
MOV             R1, R2
LDR             R2, =(aLoad_biasPBase - 0xB39189AC)
ADD             R2, PC, R2 ; "load_bias:%p base:%p\n"
BL              printf_log
```

```
LDRH            R3, [R11,#var_EE]
MOV             R2, R3
MOV             R3, R2
MOV             R3, R3,LSL#1
ADD             R3, R3, R2
MOV             R3, R3,LSL#3
MOV             R0, R3
BL              malloc
```

for循环，继续读取配置

```
loc_B39189E8
LDRH            R3, [R11,#var_EE]
MOV             R2, R3
MOV             R3, R2
MOV             R3, R3,LSL#1
ADD             R3, R3, R2
MOV             R3, R3,LSL#3
MOV             R2, R3
LDRH            R3, [R11,#var_F0]
MOV             R1, R3
LDR             R3, =0xFFFFDD28
SUB             R0, R11, #-var_4
LDR             R3, [R0,R3]
ADD             R3, R1, R3
LDR             R0, [R11,#var_28]
LDR             R1, [R11,#var_10]
BL              pread
MOV             R3, R0
STR             R3, [R11,#var_8]
LDR             R3, [R11,#var_8]
MOV             R0, #6
LDR             R2, =(aTxtag - 0xB3918A44)
ADD             R2, PC, R2 ; "txtag"
MOV             R1, R2
LDR             R2, =(aReadCountX - 0xB3918A50)
ADD             R2, PC, R2 ; "read count:%x"
BL              printf_log

loc_B3918A50
LDRH            R3, [R11,#var_EE]
MOV             R2, R3
MOV             R3, R2
MOV             R3, R3,LSL#1
ADD             R3, R3, R2
MOV             R3, R3,LSL#3
MOV             R2, R3
LDR             R3, [R11,#var_8]
CMP             R2, R3
BHI             loc_B39189E8

loc_B3918A84
LDR             R3, [R11,#var_10]
LDR             R2, [R3]
LDR             R3, [R11,#var_30]
ADD             R3, R2, R3
STR             R3, [R11,#var_38]
LDR             R3, [R11,#var_10]
LDR             R2, [R3,#4]
LDR             R3, [R11,#var_38]
ADD             R3, R2, R3
STR             R3, [R11,#var_3C]
LDR             R3, [R11,#var_38]
BIC             R3, R3, #0xFF0
BIC             R3, R3, #0xF
STR             R3, [R11,#var_40]
LDR             R3, [R11,#var_3C]
ADD             R3, R3, #0xFF0
ADD             R3, R3, #0xF
```

```
BIC             R3, R3, #0xFF0
BIC             R3, R3, #0xF
STR             R3, [R11,#var_44]
LDR             R2, [R11,#var_44]
LDR             R3, [R11,#var_40]
RSB             R3, R3, R2
STR             R3, [R11,#var_48]
MOV             R3, #0
STR             R3, [R11,#var_8]
MOV             R3, #0
STR             R3, [R11,#var_18]
LDR             R3, [R11,#var_10]
LDR             R3, [R3,#0xC]
CMP             R3, #0
BEQ             loc_B3918E54



loc_B3918ECC
LDRH            R3, [R11,#var_EE]
MOV             R2, R3
LDR             R3, [R11,#var_14]
CMP             R2, R3
BGT             loc_B3918A84
```

while循环解密

解密出待解压的数据，通过zlib库的inflate_0函数解压到指定位置

```
loc_B3918BAC
SUB             R3, R11, #-var_22C0
SUB             R3, R3, #4
SUB             R3, R3, #0x10
MOV             R0, R3
MOV             R1, #0xFFFFFFF1
LDR             R3, =(a1_2_3 - 0xB3918BCC)
ADD             R3, PC, R3 ; "1.2.3"
MOV             R2, R3
MOV             R3, #0x38
BL              inflateInit2
MOV             R3, R0
CMP             R3, #0
BNE             loc_B3918BAC
B               loc_B3918DAC

loc_B3918BE4
LDR             R3, [R11,#var_8]
ADD             R2, R3, #0x1000
LDR             R3, [R11,#var_10]
LDR             R3, [R3,#0xC]
CMP             R2, R3
BLS             loc_B3918C10
LDR             R3, [R11,#var_10]
LDR             R2, [R3,#0xC]
LDR             R3, [R11,#var_8]
RSB             R3, R3, R2
B               loc_B3918C14

loc_B3918C10
MOV             R3, #0x1000

loc_B3918C14
STR             R3, [R11,#var_4C]
LDR             R3, [R11,#var_8]
ADD             R2, R3, #0x1000
LDR             R3, [R11,#var_10]
LDR             R3, [R3,#0x14]
CMP             R2, R3
BLS             loc_B3918C44
```

```
LDR             R3, [R11,#var_10]
LDR             R2, [R3,#0x14]
LDR             R3, [R11,#var_8]
RSB             R3, R3, R2
B               loc_B3918C48

loc_B3918C44
MOV             R3, #0x1000

loc_B3918C48
STR             R3, [R11,#var_50]
LDR             R3, [R11,#var_10]
LDR             R2, [R3,#8]
LDR             R3, =0xFFFFDD28
SUB             R0, R11, #-var_4
LDR             R3, [R0,R3]
ADD             R2, R2, R3
LDR             R3, [R11,#var_8]
ADD             R3, R2, R3
MOV             R12, R3
SUB             R3, R11, #-var_2280
SUB             R3, R3, #4
SUB             R3, R3, #0x18
LDR             R0, [R11,#var_28]
MOV             R1, R3
LDR             R2, [R11,#var_4C]
MOV             R3, R12
BL              pread
MOV             R3, R0
STR             R3, [R11,#var_4C]
SUB             R3, R11, #-var_2280
SUB             R3, R3, #4
SUB             R3, R3, #0x18
LDR             R2, =(aTx12345tx12345 - 0xB3918CB0)
ADD             R2, PC, R2 ; "Tx:12345Tx:12345"
MOV             R0, R2
MOV             R1, R3
LDR             R2, [R11,#var_50]
MOV             R3, #0x10
BL              decrypt
LDR             R2, =0xFFFFDD30
MOV             R3, #4
SUB             R1, R11, #-var_4
ADD             R2, R1, R2
ADD             R3, R2, R3
LDR             R2, [R11,#var_4C]
STR             R2, [R3]
LDR             R2, =0xFFFFDD30
SUB             R3, R11, #-var_2280
SUB             R3, R3, #4
SUB             R3, R3, #0x18
SUB             R12, R11, #-var_4
STR             R3, [R12,R2]
LDR             R3, [R11,#var_4C]
MOV             R0, #6
LDR             R2, =(aTxtag - 0xB3918D08)
ADD             R2, PC, R2 ; "txtag"
MOV             R1, R2
LDR             R2, =(aReadCountX - 0xB3918D14)
ADD             R2, PC, R2 ; "read count:%x"
BL              printf_log
LDR             R2, =0xFFFFDD30
MOV             R3, #0x10
SUB             R0, R11, #-var_4
ADD             R2, R0, R2
ADD             R3, R2, R3
MOV             R2, #0x100000
STR             R2, [R3]
LDR             R2, [R11,#var_18]
```

```
LDR             R3, [R11,#var_38]
ADD             R3, R2, R3
MOV             R2, R3
LDR             R1, =0xFFFFDD30
MOV             R3, #0xC
SUB             R12, R11, #-var_4
ADD             R1, R12, R1
ADD             R3, R1, R3
STR             R2, [R3]
SUB             R3, R11, #-var_22C0
SUB             R3, R3, #4
SUB             R3, R3, #0x10
MOV             R0, R3
MOV             R1, #0
BL              inflate_0
STR             R0, [R11,#var_54]
LDR             R2, [R11,#var_18]
LDR             R1, =0xFFFFDD30
MOV             R3, #0x10
SUB             R0, R11, #-var_4
ADD             R1, R0, R1
ADD             R3, R1, R3
LDR             R3, [R3]
RSB             R3, R3, R2
ADD             R3, R3, #0x100000
STR             R3, [R11,#var_18]
LDR             R2, [R11,#var_8]
LDR             R3, [R11,#var_4C]
ADD             R3, R2, R3
STR             R3, [R11,#var_8]
```

伪代码

打印sdk版本号

将原始so文件偏移var_22DC(0x6D88，文件尾部附加数据开头)，长度为0x58的数据存入栈中var_F8里，很明显这个数据段中(var_F8-var_a0)是一些配置信息

根据上面的0x58字节数据，映射出一块大小为0x20C9C内存

根据0x58数据的配置var_ee，随机分配var_ee*24（48）字节内存空间，返回指针变量var_34

申请0x30内存块，接着从so结尾读取数据(紧接着0x58之后)，数据读取到var_10（范围var_10-var_40）

继续在var_40映射一块长度为var_48(0x1E000)的内存块

初始化解压函数用到的结构体var_22AC（z_streamp ）

循环每次读取0x1000字节数据到var_229C里，总共读取长度由0x10+0xC和0x10+0x14处的配置信息确定的

解密var_229C中的压缩数据，用zlib库进行解压缩操作，解压到var_38指定内存中

后面会有一些符号替换的操作来讲JNI_OnLoad地址重新定位

```
int __fastcall sub_B391863C(int soPath, int offset)
{
   var_22D8 = soPath;
   var_22DC = offset;

   memset(&var_A0, 0, 0x40);
   property_get("ro.build.version.sdk", &var_A0);
   android_printf_log(6, "txtag", "version:%d", atoi(&var_A0));

   do
   {
       var_28 = open(var_22D8, 0x80000);
   }
   while(var_28 == -1);

   memset(&var_F8, 0, 0x58);
   for(var_8=0; var_8<=0x57;var8=pread(var_28, &var_F8, 0x58, var_22DC))
```

```
    ;

_android_log_printf(6, "txtag", "load library %s at offset %x read count %x\n", var_22D8, var_22EC, var_22E8);
memset(&var_29C, 0, 0x1A0);
_android_log_printf(6, "txtag", "min_vaddr:%x size:%x\n", var_F8, var_F4);

var_2C = var_F8;

do
{
    var_C = mmap(var_2C, var_F4, 0, 0x22, 0XFFFFFFFF, 0);
    if(var_C <= 0x457FFFFF && var_C > 0x40000000-var_F4 && dword_B391B004 <=0xA)
    {
        _android_log_printf(6, "txtag", "addr:%p", var_C);
        var_C = -1;
    }
}while(var_C == -1);

var_30 = var_C - var_2C;    //var_C
var_210 = var_C;            //var_C
var_180 = var_30;           //var_C
var_1F0 = var_E8 + var_30;
var_1EC = var_E4 + var_30;

if(var_E0==0)
    var_1AC = 0;
else
    var_1AC = var_E0 + var_30;

if(var_DC == 0)
    var_1BC = 0;
else
    var_1BC = var_DC +var_30;

var_1B8 = (__int16)var_CE;
var_1E8 = var_C4;
var_1E0 = var_BC + var_30;
var_1E4 = var_C0;
var_1DC = var_BC + var_30 + var_C4 * 4;
var_1A4 = var_A8;
var_1A0 = (__int16)var_A6;
var_1D4 = var_B8, var_30;
var_1D0 = var_B4;
var_1CC = var_AC + var_30;

_android_log_printf(6, "txtag", "load_bias:%p base:%p\n", var_30, var_C);

var_34 = malloc(var_ee * 24);
var_10 = var_34;

var_8 = 0;
for(var_8=0; var_ee*24 > var_8; var_8 = pread(var_28, var_10, var_ee*24, var_22DC+(__int16)var_F0))
{
    _android_log_printf(6, "txtag", "read count:%x", var_8);
}

for(var_14=0; var_14<var_ee; var_14++)
{
    var_38 = *(unkown *)var_10 + var_30;
    var_40 = var_38 & 0xFFFFF000;
    var_3C = *(unkown *)(var_10 + 4) + var_38;
    var_44 = (var3C + 0xFFF ) & 0xFFFFF000;
    var_48 = var_44 - var_40;

    if(*(unkown *)(var_10+0xC) != 0)
    {
        mmap(var_40, var_48, 3, 0x32, 0xFFFFFFFF, 0);

        var_8 = 0;
```

```
            while(deflateInit(&var_22AC, 0xFFFFFFF1, "1.2.3", 0x38) != 0)
                ;

            while(*(unkown *)(var_10+0xC) > var_8)
            {
                if(var_8 + 0x1000 <= *(unkown *)(var_10+0xC))
                    var_4C = 0x1000;
                else
                    var_4C = *(unkown *)(var_10+0xC) - var_8;

                if(var_8 + 0x1000 <= *(unkown *)(var_10+0x14))
                    var_50 = 0x1000;
                else
                    var_50 = *(unkown *)(var_10+0x14) - var_8;

                pread(var_28, &var_229C, var_4C, var_8+var_22DC+*(unkown *)(var_10+8));
                decrypt("Tx:12345Tx:12345", &var_229C, var_50, 0x10);
                _android_log_printf(6, "txtag", "read count:%x", var_4C);
                var_54 = inflate_0(&var_229C, 0);


                var_8 += var_4C;
            }

            inflateEnd_0(&var_229C);
            for(var_1C = var_38;var_18+var_38>var_1C;var_1C+=0x400)
                cacheflush(var_1C, 0x400);
        }
        var_10 += 0x18;
    }
}
```

## 小结

看汇编代码，先看控制流程，找CMP、CMN等确定是什么语句，接着看blx，查看需要多少参数，接着往上找参数值

对于大量LDR、STR、MOV指令连续指令，确定是局部变赋值，直接高亮STR查看

【1】汇编语句中，for循环和while循环的相同点，判断语句都在整个语句块底部。不同点：while循环中，条件判断成功之后就直接跳转到循环体。for循环条件判断成功之

【2】ADD Rd,0xFFF，BIC Rd,0xFFF，可以计算出Rd跟0x1000对其的值

【3】程序执行过程中，需要记住的是栈保存的参数和变量，寄存器随时发生变化

【4】调用规范：每次进行Branch分支跳转的时候，都会将下一条语句地址存入LR

【5】调用规范：开始调用栈时，参数放入栈顶附近(如果当前栈帧内的函数调用参数都小于4，那么参数入栈就在栈顶，否则要留有足够空间给参数)，局部变量放在栈底

## 参考

【1】ARM指令查询 https://sourceware.org/cgen/gen-doc/arm-thumb-insn.html#insn-bhi

目录