

安全 sdk-so 保护

抹掉 Section 节表信息

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
000h:	7F	45	4C	46	01	01	01	00	00	00	00	00	00	00	00	00	.ELF.....
010h:	03	00	28	00	01	00	00	00	00	00	00	00	34	00	00	00	..(.....4...
020h:	F4	07	0B	00	00	02	00	05	34	00	20	00	09	00	28	00	8.....4...(. ...
030h:	1B	00	1A	00	06	00	00	00	34	00	00	00	34	00	00	004...4...
040h:	34	00	00	00	20	01	00	00	20	01	00	00	04	00	00	00	4... ..
050h:	04	00	00	00	03	00	00	00	54	01	00	00	54	01	00	00T...T...
060h:	54	01	00	00	13	00	00	00	13	00	00	00	04	00	00	00	T.....
070h:	01	00	00	00	01	00	00	00	00	00	00	00	00	00	00	00
080h:	00	00	00	00	34	D7	0A	00	34	D7	0A	00	05	00	00	004*..4*.....
090h:	00	10	00	00	01	00	00	00	70	E5	0A	00	70	F5	0A	00pă..põ..
0A0h:	70	F5	0A	00	84	20	00	00	7C	6D	00	00	06	00	00	00	põ... ..lm.....
0B0h:	00	10	00	00	02	00	00	00	DC	EA	0A	00	DC	FA	0A	00Ûè..Ûú..
0C0h:	DC	FA	0A	00	28	01	00	00	28	01	00	00	06	00	00	00	Ûú..(..(.....
0D0h:	04	00	00	00	04	00	00	00	68	01	00	00	68	01	00	00h...h...
0E0h:	68	01	00	00	24	00	00	00	24	00	00	00	04	00	00	00	h...\$...\$.....
0F0h:	04	00	00	00	51	E5	74	64	00	00	00	00	00	00	00	00	...Qâtd.....
100h:	00	00	00	00	00	00	00	00	00	00	00	00	06	00	00	00
110h:	00	00	00	00	01	00	00	70	E8	72	0A	00	E8	72	0A	00për..èr..
120h:	E8	72	0A	00	B8	1B	00	00	B8	1B	00	00	04	00	00	00	èr... ..
130h:	04	00	00	00	52	E5	74	64	70	E5	0A	00	70	F5	0A	00Râtdpă..põ..
140h:	70	F5	0A	00	90	0A	00	00	90	0A	00	00	06	00	00	00	põ

template Results - ELFTemplate1.bt

	Name
✓ struct file	
> struct elf_header	
> struct program_header_table	
✓ struct section_header_table	
> struct section_table_entry32_t section_table_element[0]	SHN_UNDEF
> struct section_table_entry32_t section_table_element[1]	

```
Output
Executing template 'D:\poke\010editor pt\ELFTemplate1.bt' on 'G:\alipay\libsg\libsgmainso-6.3.80.so'...
ERROR Line 480: Template passed end of file at variable 's data'.
```

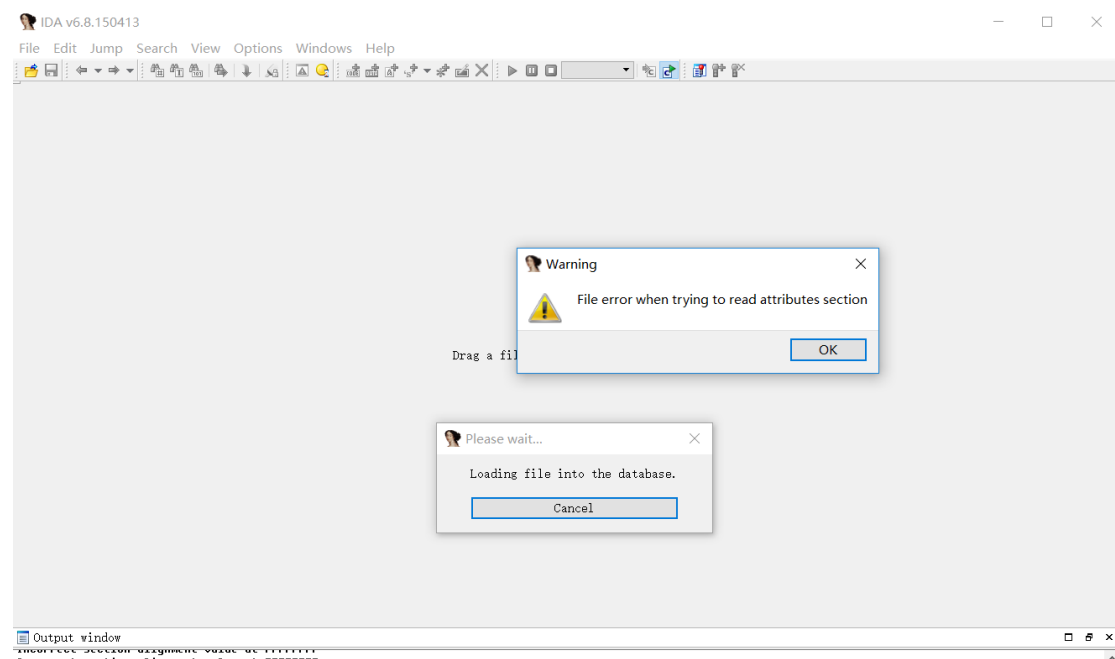
节头表信息:

000h:	7F 45 4C 46 01 01 01 00 00 00 00 00 00 00 00 00 00	S.....
010h:	03 00 28 00 01 00 00 00 00 00 00 00 00 34 00 00 00	Y.....
020h:	F4 07 0B 00 00 02 00 05 34 00 20 00 09 00 28 00	8.....4...(. ...
030h:	1B 00 1A 00 06 00 00 00 34 00 00 00 34 00 00 004...4...
040h:	34 00 00 00 20 01 00 00 20 01 00 00 04 00 00 00	4... ..
050h:	04 00 00 00 03 00 00 00 54 01 00 00 54 01 00 00T...T...
060h:	54 01 00 00 13 00 00 00 13 00 00 00 04 00 00 00	T.....
070h:	01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
080h:	00 00 00 00 34 D7 0A 00 34 D7 0A 00 05 00 00 004*..4*.....
090h:	00 10 00 00 01 00 00 00 70 E5 0A 00 70 F5 0A 00pă..põ..
0A0h:	70 F5 0A 00 84 20 00 00 7C 6D 00 00 06 00 00 00	põ... ..lm.....
0B0h:	00 10 00 00 02 00 00 00 DC EA 0A 00 DC FA 0A 00Ûè..Ûú..
0C0h:	DC FA 0A 00 28 01 00 00 28 01 00 00 06 00 00 00	Ûú..(..(.....
0D0h:	04 00 00 00 04 00 00 00 68 01 00 00 68 01 00 00h...h...
0E0h:	68 01 00 00 24 00 00 00 24 00 00 00 04 00 00 00	h...\$...\$.....
0F0h:	04 00 00 00 51 E5 74 64 00 00 00 00 00 00 00 00	...Qâtd.....
100h:	00 00 00 00 00 00 00 00 00 00 00 00 06 00 00 00
110h:	00 00 00 00 01 00 00 70 E8 72 0A 00 E8 72 0A 00për..èr..
120h:	E8 72 0A 00 B8 1B 00 00 B8 1B 00 00 04 00 00 00	èr... ..
130h:	04 00 00 00 52 E5 74 64 70 E5 0A 00 70 F5 0A 00Râtdpă..põ..
140h:	70 F5 0A 00 90 0A 00 00 90 0A 00 00 06 00 00 00	põ

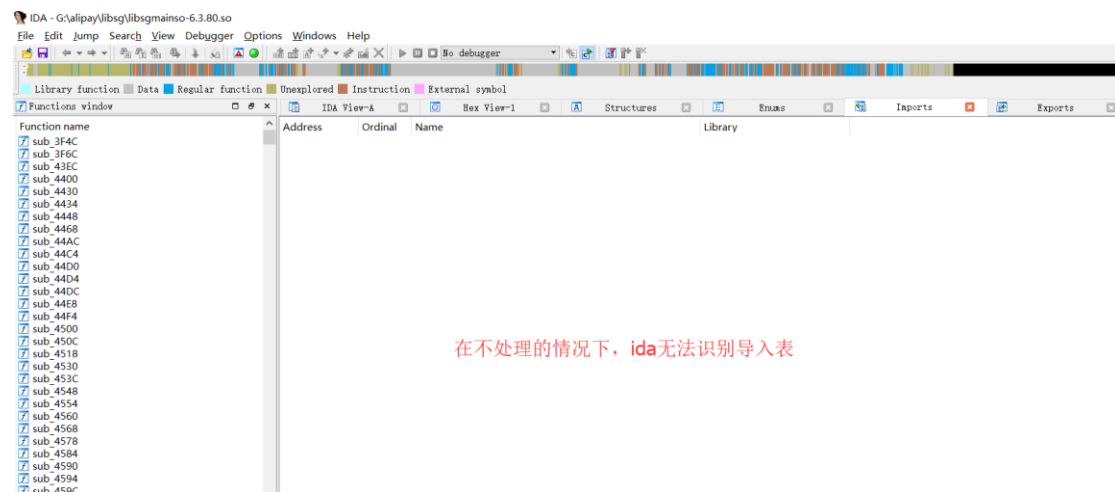
template Results - ELFTemplate.bt

	Name
✓ struct section_table_entry32_t section_table_element[1]	
> struct s_name32_t s_name	SHN_UNDEF
enum s_type32_e s_type	SHT_PROGBITS (1)

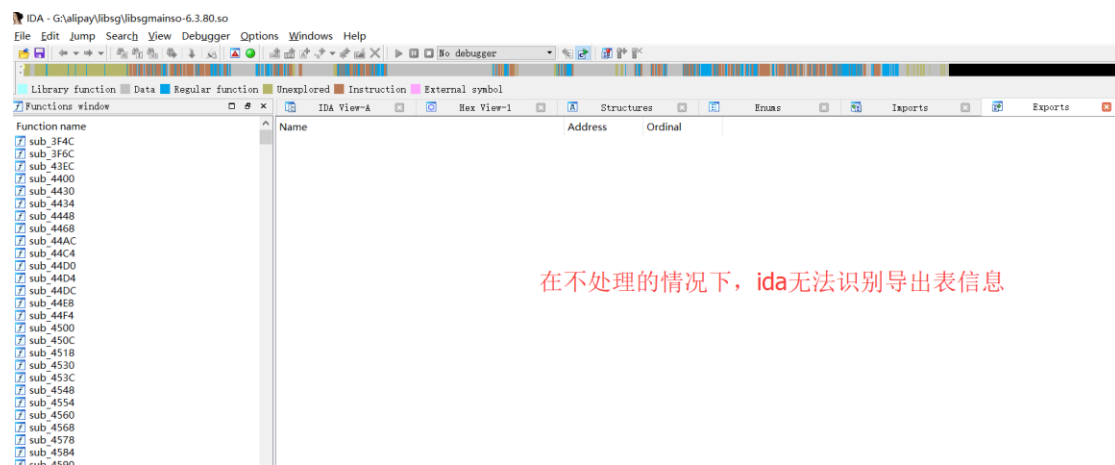
抹掉节头表导致 IDA（类型的基于节头表解析的逆向工具）加载失败



IDA 无法识别导入导出表



在不处理的情况下，ida无法识别导入表



在不处理的情况下，ida无法识别导出表信息

```

12  libsgmainso_6.3.80.so:E93DDA68
libsgmainso_6.3.80.so:E93DDA68 000 70 B5 PUSH {R4-R6,LR}
libsgmainso_6.3.80.so:E93DDA6A 010 09 A4 ADR R4, unk_E93DDA90
libsgmainso_6.3.80.so:E93DDA6C 010 02 25 20 35 MOVS R5, #0x22
libsgmainso_6.3.80.so:E93DDA70 010 64 19 ADDS R4, R4, R5
libsgmainso_6.3.80.so:E93DDA72 010 64 1C ADDS R4, R4, #1
libsgmainso_6.3.80.so:E93DDA74 010 01 D0 BEQ loc_E93DDA7A
libsgmainso_6.3.80.so:E93DDA76 010 00 D1 BNE loc_E93DDA7A
libsgmainso_6.3.80.so:E93DDA78 010 7F BD POP {R0-R6,PC}
libsgmainso_6.3.80.so:E93DDA7A
libsgmainso_6.3.80.so:E93DDA7A loc_E93DDA7A
libsgmainso_6.3.80.so:E93DDA7A ; CODE XREF: sub_E93DDA68+CTJ
libsgmainso_6.3.80.so:E93DDA7A ; sub_E93DDA68+ETJ
libsgmainso_6.3.80.so:E93DDA7C 010 03 94 STR R4, [SP,#0x10+var_4]
libsgmainso_6.3.80.so:E93DDA7E 010 00 9C LDR R4, [SP,#0x10+var_10]
libsgmainso_6.3.80.so:E93DDA80 010 01 9D LDR R5, [SP,#0x10+var_C]
libsgmainso_6.3.80.so:E93DDA82 010 6E 46 MOV R6, SP
libsgmainso_6.3.80.so:E93DDA84 010 08 36 ADDS R6, #8
libsgmainso_6.3.80.so:E93DDA86 010 B5 46 MOV SP, R6
libsgmainso_6.3.80.so:E93DDA88 010 40 BD POP {R6,PC}
libsgmainso_6.3.80.so:E93DDA8A ; End of function sub_E93DDA68
libsgmainso_6.3.80.so:E93DDA8C
libsgmainso_6.3.80.so:E93DDA8E

```

```

R12 libsgmainso_6.3.80.so:E93DDA68
libsgmainso_6.3.80.so:E93DDA68 000 70 B5 PUSH {R4-R6,LR}
libsgmainso_6.3.80.so:E93DDA6A 010 09 A4 ADR R4, unk_E93DDA90
libsgmainso_6.3.80.so:E93DDA6C 010 02 25 20 35 MOVS R5, #0x22
libsgmainso_6.3.80.so:E93DDA70 010 64 19 ADDS R4, R4, R5
libsgmainso_6.3.80.so:E93DDA72 010 64 1C ADDS R4, R4, #1
libsgmainso_6.3.80.so:E93DDA74 010 01 D0 BEQ loc_E93DDA7A
libsgmainso_6.3.80.so:E93DDA76 010 00 D1 BNE loc_E93DDA7A
libsgmainso_6.3.80.so:E93DDA76
libsgmainso_6.3.80.so:E93DDA78 010 7F BD DCW 0xBD7F
libsgmainso_6.3.80.so:E93DDA7A
libsgmainso_6.3.80.so:E93DDA7A loc_E93DDA7A
libsgmainso_6.3.80.so:E93DDA7A ; CODE XREF: sub_E93DDA68+Cfj
libsgmainso_6.3.80.so:E93DDA7A ; sub_E93DDA68+Efj
libsgmainso_6.3.80.so:E93DDA7A 010 03 94 STR R4, [SP,#0x10+var_4]
libsgmainso_6.3.80.so:E93DDA7C 010 00 9C LDR R4, [SP,#0x10+var_10]
libsgmainso_6.3.80.so:E93DDA7E 010 01 9D LDR R5, [SP,#0x10+var_1C]
libsgmainso_6.3.80.so:E93DDA80 010 6E 46 MOV R6, SP
libsgmainso_6.3.80.so:E93DDA82 010 08 36 ADDS R6, #8
libsgmainso_6.3.80.so:E93DDA84 010 B5 46 MOV SP, R6
libsgmainso_6.3.80.so:E93DDA86 010 40 BD POP {R6,PC}
libsgmainso_6.3.80.so:E93DDA86 ; End of function sub_E93DDA68
libsgmainso_6.3.80.so:E93DDA86

```

```

jsgmainso_6.3.80.so:EF416A88 010 97 46 04 B5 DCD 0x05044697
jsgmainso_6.3.80.so:EF416A8C 010 07 BD C0 46 DCD 0x46C0BD07
jsgmainso_6.3.80.so:EF416A90
jsgmainso_6.3.80.so:EF416A90
jsgmainso_6.3.80.so:EF416A90 010 11 60
jsgmainso_6.3.80.so:EF416A92 010 70 BD
jsgmainso_6.3.80.so:EF416A92
jsgmainso_6.3.80.so:EF416A92
jsgmainso_6.3.80.so:EF416A94
jsgmainso_6.3.80.so:EF416A94 11 60
jsgmainso_6.3.80.so:EF416A96 08 BD
jsgmainso_6.3.80.so:EF416A98
jsgmainso_6.3.80.so:EF416A98 B7 46
jsgmainso_6.3.80.so:EF416A9A
jsgmainso_6.3.80.so:EF416A9A 0C B5
jsgmainso_6.3.80.so:EF416A9C 00 00
jsgmainso_6.3.80.so:EF416A9E 0C B5
jsgmainso_6.3.80.so:EF416AA0 F0 47
jsgmainso_6.3.80.so:EF416AA0
jsgmainso_6.3.80.so:EF416AA2
jsgmainso_6.3.80.so:EF416AA2 38
jsgmainso_6.3.80.so:EF416AA3 BD
jsgmainso_6.3.80.so:EF416AA4 11

```

loc_EF416A90 ; DATA XREF: JNI_OnLoad+270
 STR R1, [R2]
 POP {R4-R6,PC}
 ; End of function JNI_OnLoad
 ;
 STR R1, [R2]
 POP {R3,PC}
 ;
 MOV PC, R6
 ;
 PUSH {R2,R3,LR}
 MOVS R3, R1
 PUSH {R2,R3,LR}
 BLX LR
 ;
 CODE32
 DCB 0x38 ; 8
 DCB 0xBD ;
 DCB 0x11

垃圾数据，迷惑反汇编器

```

SUB      SP, SP, #8
SUB      SP, SP, #8
PUSH     {R0,R1,LR}
BLX      sub_4964
LSLS     R2, R5, #3
MOVS     R0, R0
POP      {R1}
CMP      R0, #0
ADD      R1, PC
LDR      R1, [R1]
PUSH     {R0,R1,LR}

```

混淆指令

另一種迷惑代碼:

```
var_4 = -4
```

```

PUSH     {R0-R2,LR}
ADR      R1, 0xB130
MOVS     R1, R1
SUBS     R1, #5
MOVS     R0, R0
MOVS     R0, R1
MOVS     R2, R2
ADDS     R0, #0x10
STR      R0, [SP,#0xC]
POP      {R0-R2,PC}

```

End of function JNI_OnLoad

	MOV	R1, LR	
	POP	{R5,R6,PC}	
	PUSH	{R2,R3}	
	BLX	loc_B130	
	NOP		
loc_B130	MOVS	R3, R3	; CODE XREF: LOAD:000001
	POP	{R2,PC}	
	PUSH	{R1,R2}	
	MOV	PC, R4	
	MOVS	R0, R2	
	PUSH	{R0,R1,LR}	

垃圾指令，擾亂ida

擾亂 ida 分析

```
EXPORT JNI_OnLoad

JNI_OnLoad
var_4 = -4

PUSH    {R0-R2,LR}
ADR     R1, 0xB130
MOVS    R1, R1
SUBS    R1, #5
MOVS    R0, R0
MOVS    R0, R1
MOVS    R2, R2
ADDS    R0, #0x10
STR     R0, [SP, #0xC]
POP     {R0-R2,PC}
; End of function JNI_OnLoad
; -----
```

pc需要計算，但
ida靜態分析遇
到pop就以爲函
數結束了

函數之前插入計算 pc 和垃圾數據，擾亂 ida

```

PUSH      {R0-R2,LR}
ADR       R1, 0x18FF8
MOVS      R1, R1
SUBS      R1, #5
MOVS      R0, R0
MOVS      R0, R1
MOVS      R2, R2
ADDS      R0, #0x28 ; '('
STR       R0, [SP,#0xc]
POP       {R0-R2,PC}

End of function sub_18FD8

```

pc需要計算，不是顯示的，擾亂ida分析，並且在函數之前，在函數之前還插入了一些垃圾數據做指令，擾亂ida分析

```

-----
POP       {R2-R5,PC}
-----
jcc_18FF8
PUSH      {R1,R5}
MOV       R4, LR
BLX       loc_18FF8
NOP
; CODE XREF: LOAD:00018FF2↑p
CMP       R3, #5
MOV       PC, R1
POP       {R1,R4-R6,PC}
-----
MOV       PC, R0
MOV       R5, LR
STR       R1, [R2]
MOV       PC, R1
MOV       PC, R0
MOV       PC, R6
MOV       LR, R3
POP       {R4,R5,PC}
MOV       R0, R1
MOV       R6, LR
PUSH      {R2,R4}
POP       {R5,PC}
POP       {R2-R5,PC}
-----
MOVS      R0, R1
PUSH.W    {R4-R8,LR}

```

真正函數開始

pop xxx , push xxx 垃圾數據指令

```

-----
POP       {R0,PC}
-----
POP       {R0,PC}
-----
PUSH      {R0,R2}
NOP
BX        R6
-----
BX        R4
-----
PUSH      {R1,R3}
MOV       PC, R6
-----
LDR       R4, [R5]
PUSH      {R1,R2}
MOVS      R0, R5
PUSH      {R0,R5}
POP       {R1-R3,PC}

```

無用的垃圾數據，特徵都是push xxx, pop xxx這個樣子

动态计算目标地址

```
libsgmainso_6.3.80.so:EED08048 01 10 CE E3 BIC R1, LR, #1
libsgmainso_6.3.80.so:EED0804C 00 01 91 E7 LDR R0, [R1,R0,LSL#2]
libsgmainso_6.3.80.so:EED08050 0E E0 80 E0 ADD LR, R0, LR
libsgmainso_6.3.80.so:EED08054 08 10 9D E5 LDR R1, [SP,#8]
libsgmainso_6.3.80.so:EED08058 08 E0 8D E5 STR LR, [SP,#8]
libsgmainso_6.3.80.so:EED0805C 01 E0 A0 E1 MOV LR, R1
libsgmainso_6.3.80.so:EED08060 03 80 BD E8 LDMFD SP!, {R0,R1,PC}
libsgmainso_6.3.80.so:EED08064 0E 10 A0 E1 MOV R1, LR
libsgmainso_6.3.80.so:EED08068 A1 10 A0 E1 MOV R1, R1,LSR#1
libsgmainso_6.3.80.so:EED0806C 81 10 A0 E1 MOV R1, R1,LSL#1
libsgmainso_6.3.80.so:EED08070 01 00 A0 E1 MOV R0, R1
libsgmainso_6.3.80.so:EED08074 00 10 91 E5 LDR R1, [R1]
libsgmainso_6.3.80.so:EED08078 00 10 81 E0 ADD R1, R1, R0
libsgmainso_6.3.80.so:EED0807C 00 00 91 E5 LDR R0, [R1]
libsgmainso_6.3.80.so:EED08080 10 00 8D E5 STR R0, [SP,#0x10]
libsgmainso_6.3.80.so:EED08084 04 E0 8E E2 ADD LR, LR, #4
libsgmainso_6.3.80.so:EED08088 0C E0 8D E5 STR LR, [SP,#0xC]
libsgmainso_6.3.80.so:EED0808C 03 40 BD E8 LDMFD SP!, {R0,R1,LR}
libsgmainso_6.3.80.so:EED08090 04 F0 9D E4 LDR PC, [SP],#4
libsgmainso_6.3.80.so:EED08094 03 00 2D E9 STMFD SP!, {R0,R1}
libsgmainso_6.3.80.so:EED08098 0E 10 A0 E1 MOV R1, LR
libsgmainso_6.3.80.so:EED0809C A1 10 A0 E1 MOV R1, R1,LSR#1
libsgmainso_6.3.80.so:EED080A0 00 01 A0 E1 MOV R0, R0,LSL#2
libsgmainso_6.3.80.so:EED080A4 81 10 A0 E1 MOV R1, R1,LSL#1
libsgmainso_6.3.80.so:EED080A8 00 10 91 E7 LDR R1, [R1,R0]
libsgmainso_6.3.80.so:EED080AC 81 10 A0 E1 MOV R1, R1,LSL#1
```

很多，这里只是截三幅图

这样会导致调用不连续，只有计算出 pc 值才能分析清楚程序走向。

pc 非顯示的

```
EXPORT JMI_OnLoad
JNI_OnLoad
var_4 = -4
PUSH {R0-R2,LR}
ADR R1, 0xB130
MOVS R1, R1
SUBS R1, #5
MOVS R0, R0
MOVS R0, R1
MOVS R2, R2
ADDS R0, #0x10
STR R0, [SP,#0xC]
POP {R0-R2,PC}
; End of function JNI_OnLoad
```

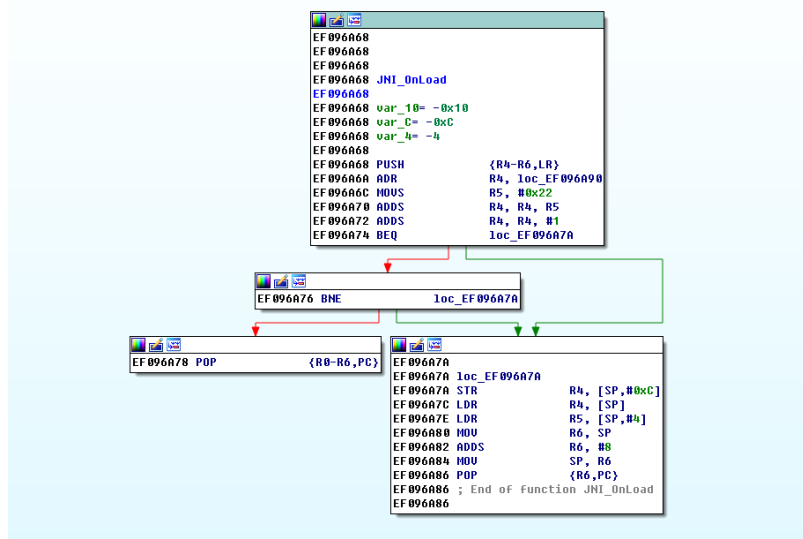
pc需要計算，但
ida靜態分析遇
到pop就以爲函
數結束了

Ida 无法解析出正确的流程图和参数（垃圾代码）

下面的 JNI_OnLoad 是我动态调试找到的，但是 ida 明显解析错了参数，和函数流程。

```
void __fastcall JNI_OnLoad(int a1, int a2, int a3, int a4, int a5, int a6, int a7, int a8)
{
    int v8; // [sp+4h] [bp-Ch]@0
    if ( &loc_EF096AB2 )
    {
        if ( !&loc_EF096AB2 )
            JUMPOUT(__CS__, a8);
        JUMPOUT(__CS__, v8);
    }
}
```

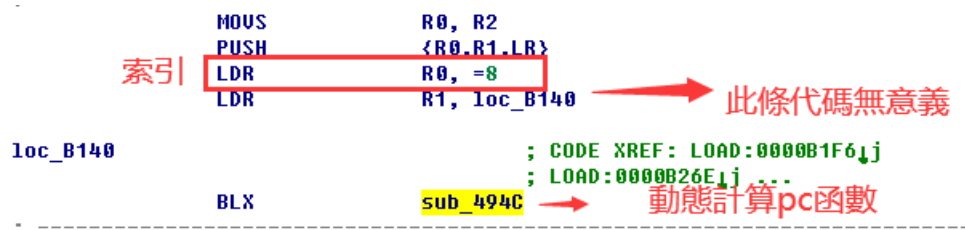
流程图：



这导致你必须单步调试，一边调试，一边帮助 ida 修正错误。

非顯示 pc，動態跳轉

pc 是非顯示的，需要進行查表計算；同時有多種計算 pc 的方式，其中一種跳轉特徵：



跳轉表：


```

LOAD:00000140          BLX          sub_494C          ; DATA XREF: LOAD:0000013C↑r
LOAD:00000140          ; -----
LOAD:00000144          dword_B144          DCD 8
LOAD:00000148          DCD 0xA8
LOAD:0000014C          DCD 0xBC
LOAD:00000150          DCD 0xCC
LOAD:00000154          DCD 0xE0
LOAD:00000158          DCD 0xF0
LOAD:0000015C          DCD 0x100
LOAD:00000160          DCD 0x110
LOAD:00000164          DCD 0x120
LOAD:00000168          DCD 0x134
LOAD:0000016C          DCD 0x14C
LOAD:00000170          DCD 0x168
LOAD:00000174          DCD 0x17C
LOAD:00000178          DCD 0x194
LOAD:0000017C          DCD 0x1AC
LOAD:00000180          DCD 0x1D4
LOAD:00000184          DCD 0x1EC
LOAD:00000188          DCD 0x208
LOAD:0000018C          DCD 0x224
LOAD:00000190          DCD 0x240
LOAD:00000194          DCD 0x268
LOAD:00000198          DCD 0x27C
LOAD:0000019C          DCD 0x290
LOAD:000001A0          DCD 0x2A4
LOAD:000001A4          DCD 0x2B8
LOAD:000001A8          DCD 0x2CC
LOAD:000001AC          DCD 0x2E0
LOAD:000001B0          DCD 0x2FC
LOAD:000001B4          DCD 0x310
LOAD:000001B8          DCD 0x324
LOAD:000001BC          DCD 0x33C
LOAD:000001C0          DCD 0x358
LOAD:000001C4          DCD 0x36C

```

另一種跳轉特徵：

```

; -----
SUB          SP, SP, #8
PUSH        {R0,R1,LR}
BLX         sub_4964
LSLS        R2, R5, #3
MOVS        R0, R0
POP         {R1}
CMP         R0, #0
ADD         R1, PC
LDR         R1, [R1]
PUSH        {R0,R1,LR}

```

字符串加密

幾乎所有 class 的字符串都被加密了

多種加密算法

該 so 用到了多種加密算法，如 rc4， aes 等，而且為了防止這些加密算法被直接識別，它還被稍作了修改。

补充

由于写这些文档时已经停止了分析工作，so 保护应该还有我没有发现的部分，此文档只是提供我发现的保护措施供大家分享。