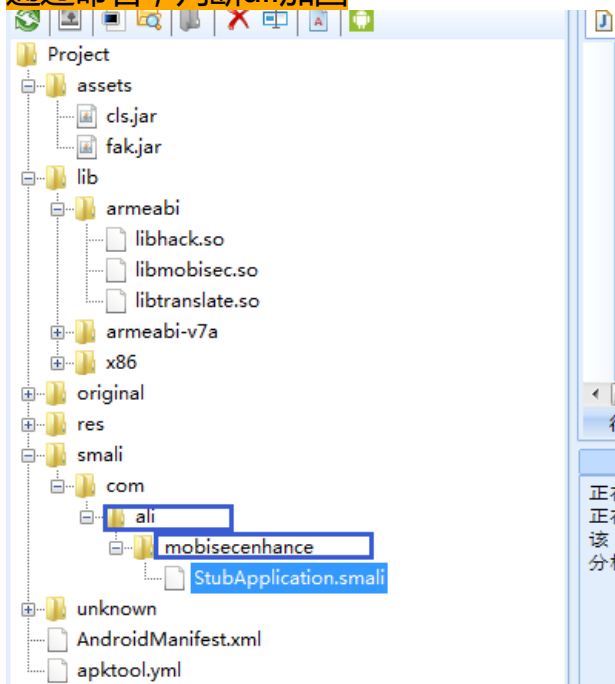


简单dex壳实例

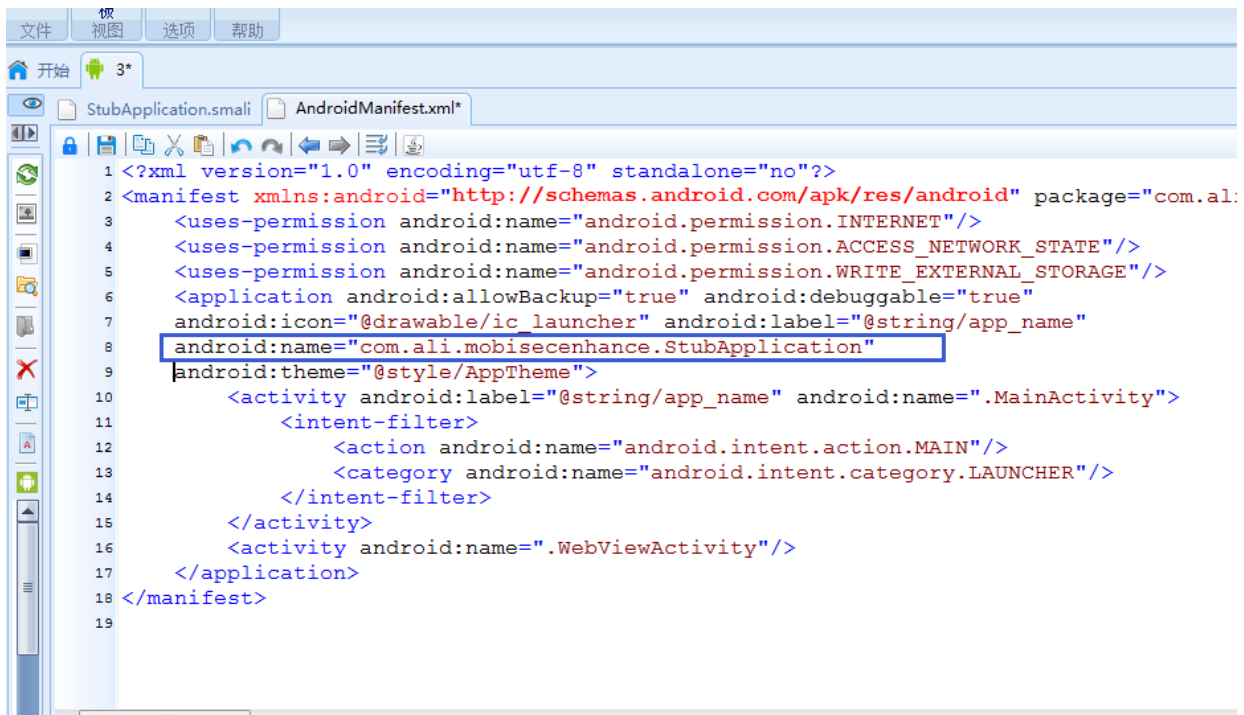
? 3.apk



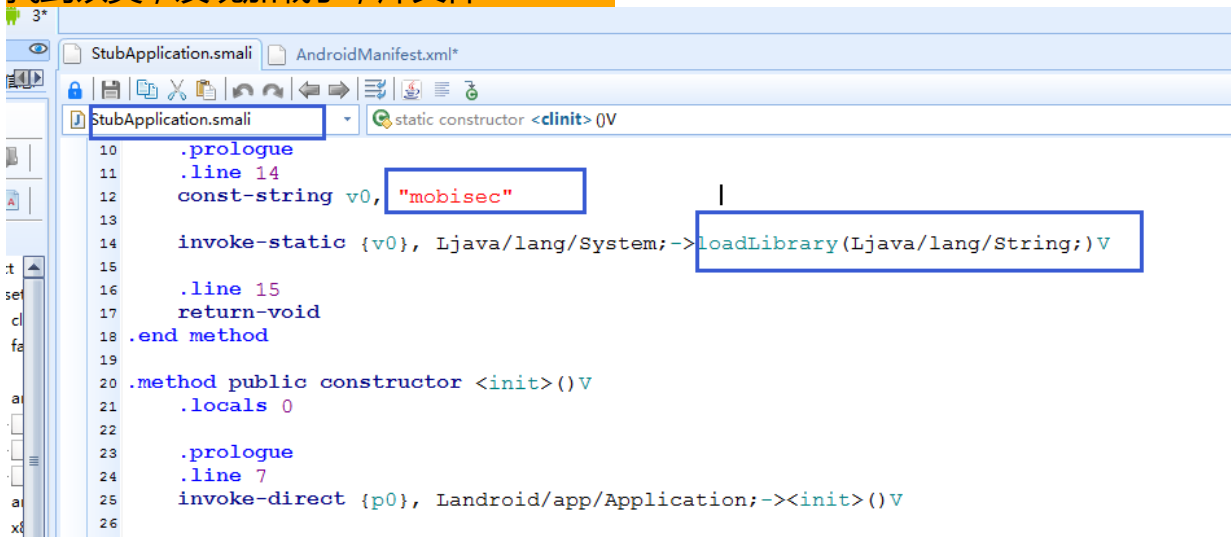
通过命名，判断ali加固



先于activity启动的类



找到该类，发现加载了，库文件mobisec

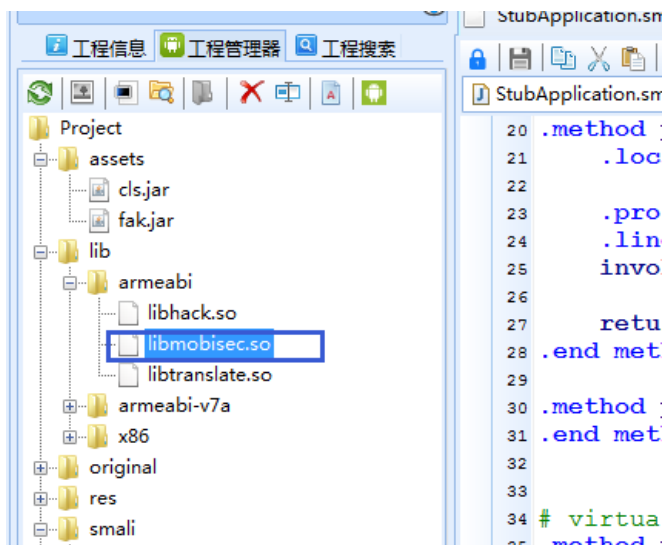


两个函数，一般用于加固

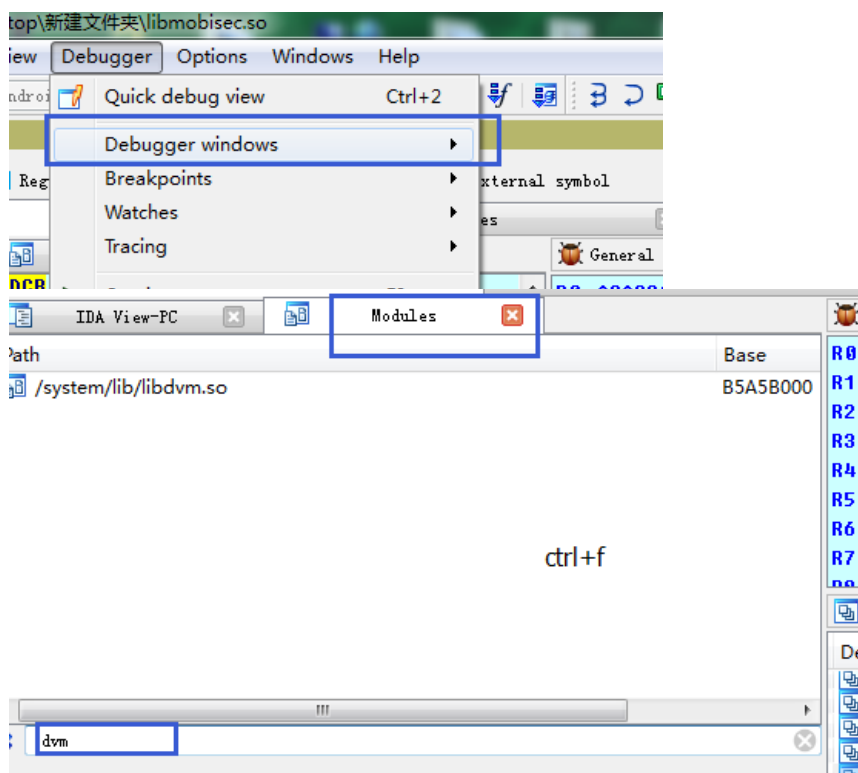
```
# virtual methods
.method protected native attachBaseContext(Landroid/content/Context;)V
.end method

.method public native onCreate()V
.end method
```

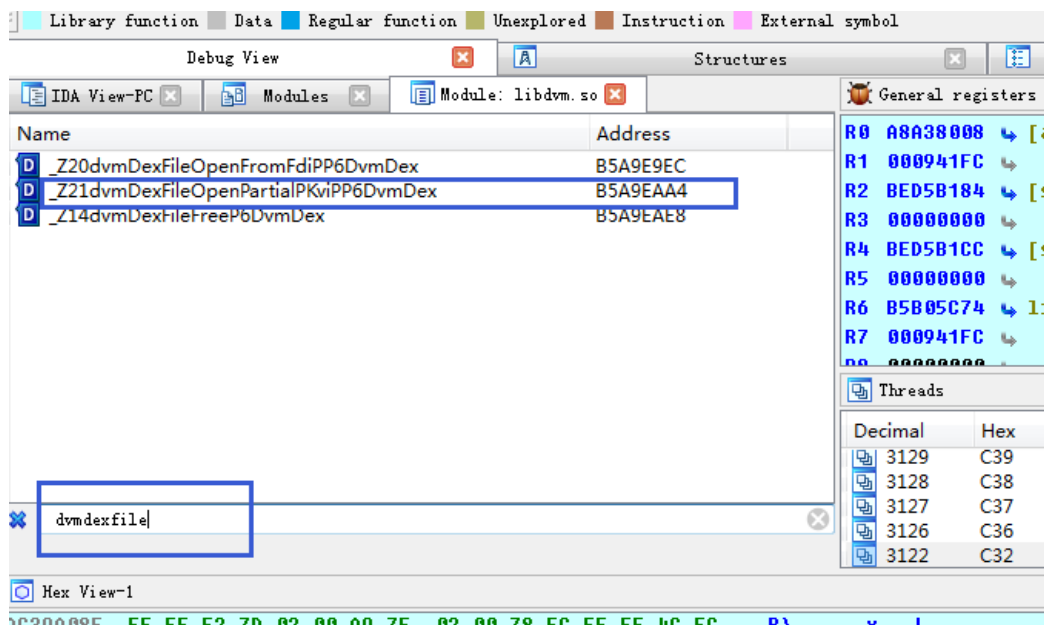
找到该so文件，IDA动态调试



打开模块窗口，Ctrl+F查找要下断的函数



ctrl+f



//安装要脱壳的软件

//命令行调试启动

```
D:\>adb shell
root@generic:/ # cd ./data/local/tmp
root@generic:/data/local/tmp # ./android_server
IDA Android 32-bit remote debug server(ST) v1.19. Hex-Rays (c) 2004-2015
Listening on port #23946...
=====
[1] Accepting connection from 127.0.0.1...
```

```
D:\>adb forward tcp:23946 tcp:23946
```

```
D:\>adb shell am start -D -n com.ali.tg.testapp/.MainActivity
Starting: Intent { cmp=com.ali.tg.testapp/.MainActivity }
```

```
D:\>jdb -connect com.sun.jdi.SocketAttach:hostname=localhost,port=8700
```

设置未捕获的java.lang.Throwable
设置延迟的未捕获的java.lang.Throwable
正在初始化jdb...

//断下来后查看下寄存器值，r0,r1,在内存中

//确定是dex后dump

General registers

R0	A8A38008	[anon:libc_malloc]:A8A38008
R1	000941FC	

Library function Data Regular function Unexplored Instruction External symbol

Debug View Modules Module: libdvm.so

libdvm.so:B5A9EAA1 DCB 0xC2 ;
libdvm.so:B5A9EAA2 DCB 4
libdvm.so:B5A9EAA3 DCB 0
libdvm.so:B5A9EAA4 CODE16
libdvm.so:B5A9EAA5 Z21dvmDexFileOpenPartialPKviPP6DvmDex DCB 0xB5 ;
libdvm.so:B5A9EAA6 DCB 0x14
libdvm.so:B5A9EAA7 DCB 0x46 ; F
libdvm.so:B5A9EAA8 DCB 0
libdvm.so:B5A9EAA9 DCB 0x22 ; ""
libdvm.so:B5A9EAAA DCB 0x41 ; A
libdvm.so:B5A9EAAB DCB 0xF0
libdvm.so:B5A9EAAc DCB 0xB9
libdvm.so:B5A9EAAE DCB 0xF8 ;
libdvm.so:B5A9EAAE DCB 5
libdvm.so:B5A9EAAF DCB 0x46 ; F
UNKNOWN B5A9EAA4: libdvm.so:Z21dvmDexFileOpen (Synchronized with PC)

General registers

R0	A8A38008	[anon:libc_malloc]:A8A38008
R1	000941FC	
R2	BED5B184	[stack]:BED5B184
R3	00000000	
R4	BED5B1CC	[stack]:BED5B1CC
R5	00000000	
R6	B5B05C74	libdvm.so:gDvmInlineOpsTable+868
R7	000941FC	
R8	00000000	

Threads

Decimal	Hex	State
3129	C38	Ready
3128	C38	Ready
3127	C37	Ready
3126	C36	Ready
3122	C32	Ready

Hex View-1

Address	Hex	Comment
37FF8	00 00 00 00 F0 4F 09 00	
38008	64 65 78 0A 30 33 35 00	dex.035.1P4...y
38018	EB 32 41 72 87 09 CE A3	.2Ar...mQ.A.U..
38028	FC 41 09 00 70 00 00 00	.A..p...xU4....
38038	00 00 00 00 20 41 09 00	...A.....p...
38048	02 04 00 00 30 64 00 00	...0d...*....8t..
38058	96 06 00 00 30 B2 00 00	...0.....

Stack view

BED5B148	00000000
BED5B14C	00000000
BED5B150	00000000
BED5B154	B6F53FE
BED5B158	B5A5A0E
BED5B15C	AC3C22B

static main(void)

{

auto fp, begin, end, dexbyte;

fp = fopen("d:\\dump1.dex", "wb");

begin = r0;

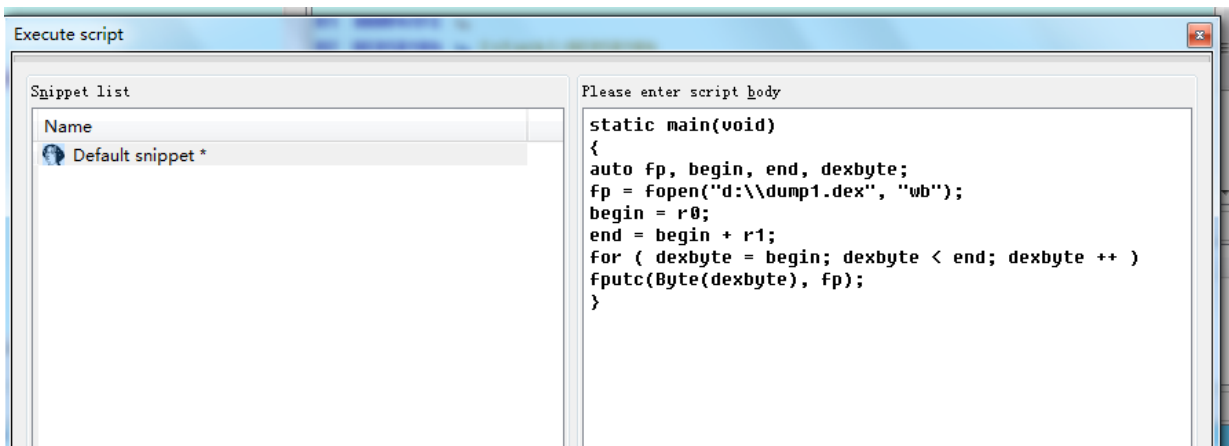
end = begin + r1;

for (dexbyte = begin; dexbyte < end; dexbyte ++)

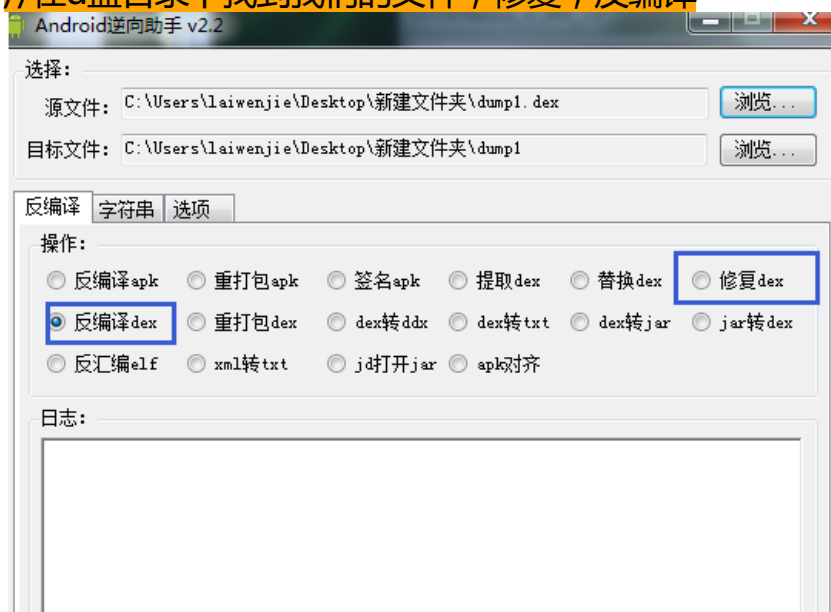
fputc(Byte(dexbyte), fp);

}

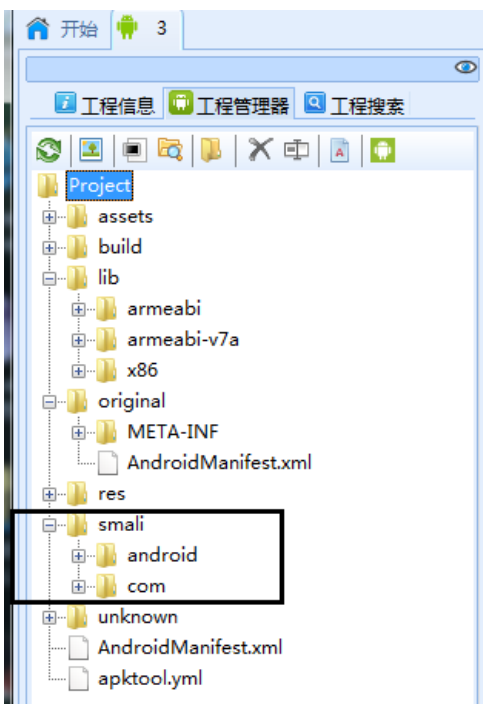
//运行脚本：菜单file->Scripcomman



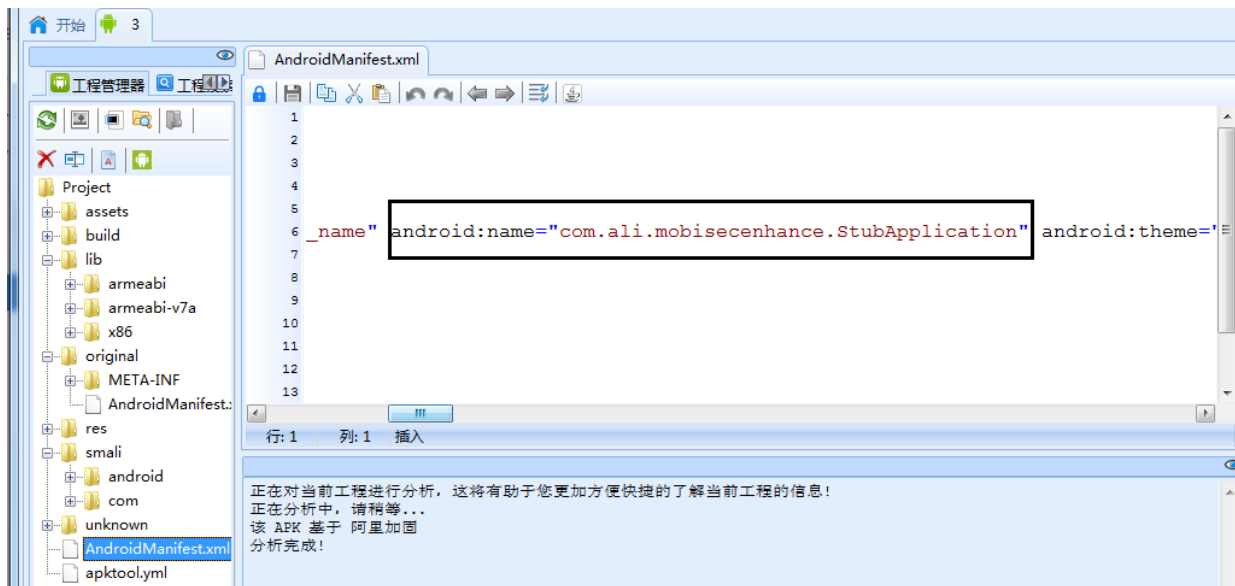
//在d盘目录下找到我们的文件，修复，反编译



在原有的smali目录下，右键，路径。将修复反编译的dex文件拷贝在打开的路径下



删除清单文件的加密启动类



最后保存编译就可以了