

## 脱壳基础（调试修改内存）

加壳脱壳第一天.rar

这个程序的demo我给弄丢了。

应该就是

一个按钮，监听事件是实现加法

再写一个减法函数。但不调用

我们做的就是希望在内存中找到加法函数指令，修改为减去的。

以在脑中建立，内存可修改的概念。

### 1.运行程序



### 2.运行android\_servers

```
管理员: C:\Windows\System32\cmd.exe - adb shell
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

D:\>adb shell
root@generic:/ # su
root@generic:/ # cd ./data/local/tmp
root@generic:/data/local/tmp # ls -l
-rwxrwxrwx root    root      523480 2015-04-13 06:35 android_server
-rw-rw-rw- root    root      1241062 2016-07-07 04:59 com.example.laiwenjie.arm
-rw-rw-rw- root    root         0 2016-07-07 05:55 com.example.laiwenjie.arm
drwxrwxr-x root    root         0 2016-07-07 05:55 lldb
-rw-rw-rw- root    root     9399052 2016-06-28 10:50 lldb-server
-rw-rw-rw- root    root        1428 2016-06-28 10:50 start_lldb_server.sh
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...
```

### 3.转发端口

adb forward tcp:23946 tcp:23946

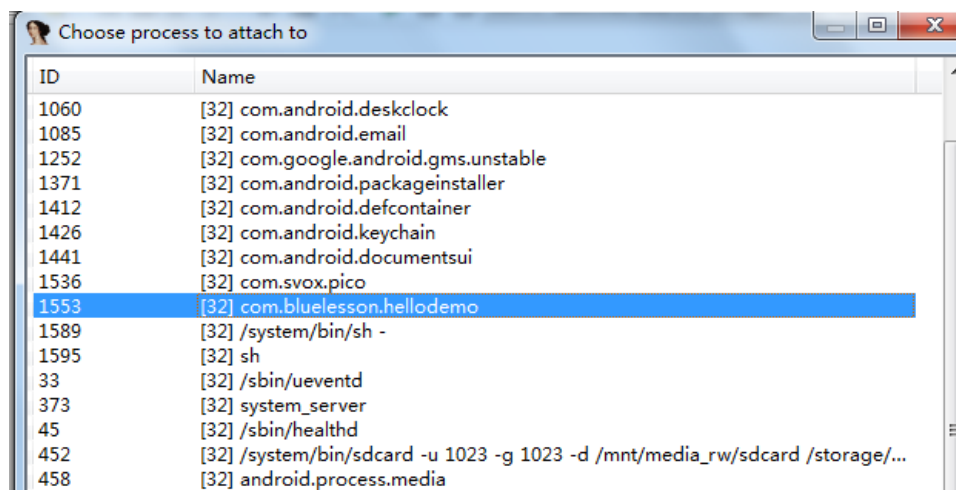
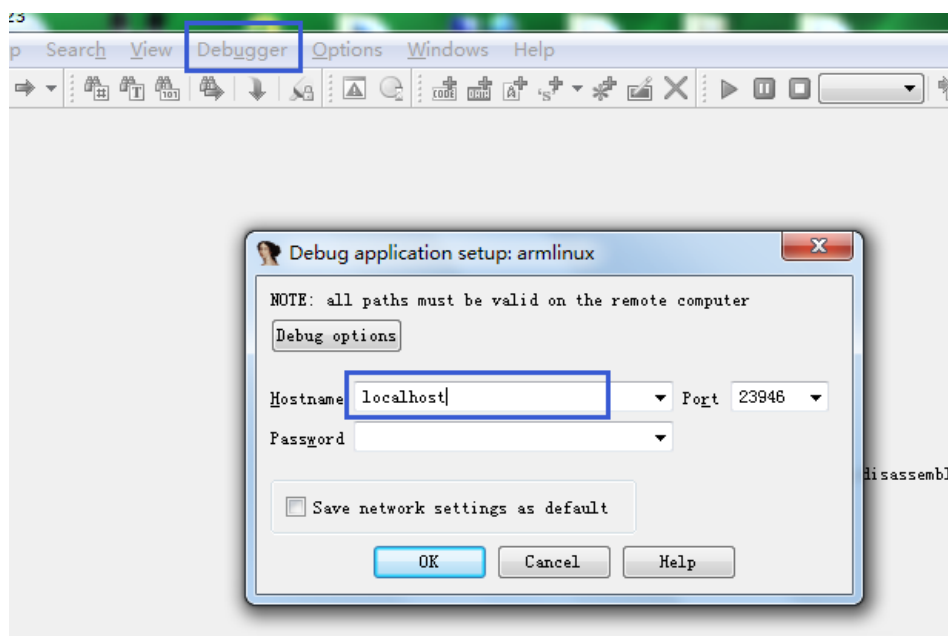
```
管理员: C:\Windows\System32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

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

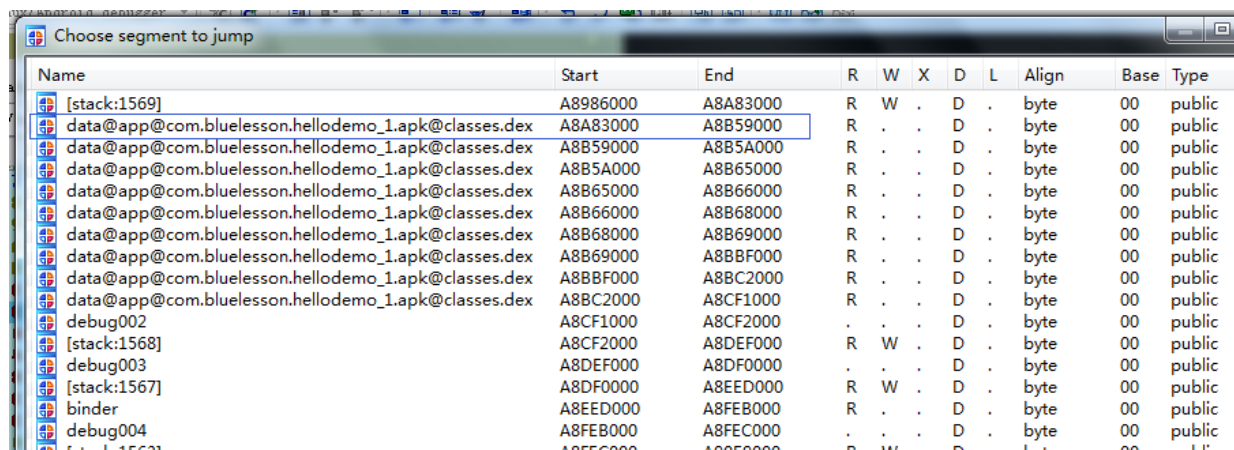
D:\>
```

### 4.IDA附加调试

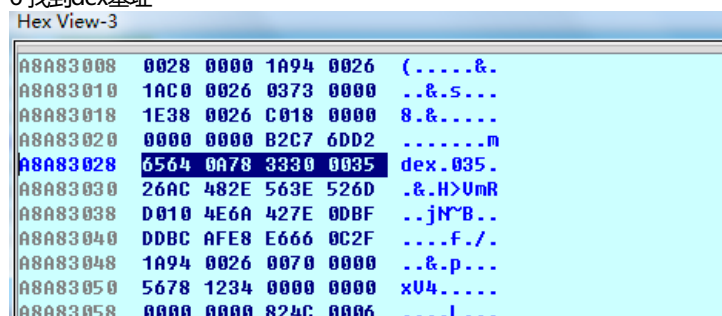
注意程序位数：32位 如果不对应 获取不到进程模块



5. ctrl+s 获取模块信息，获得基址



6 找到dex基址



用另一个IDA找到函数偏移地址

```
f PopupWindowCompat$Api23PopupWindowImpl_getW
f PopupWindowCompat$Api23PopupWindowImpl_setOx
f PopupWindowCompat$Api23PopupWindowImpl_setWi
f TextViewCompat$Api23TextViewCompatImpl_init_@V
f TextViewCompat$Api23TextViewCompatImpl_setTextAp
f ActionBarActivity_init_@V
f AppCompatActivityDelegateImplV23_init_@VLLL
f AppCompatActivityDelegateImplV23_mapNightMode@II
f AppCompatActivityDelegateImplV23_wrapWindowCallback@L
f MainActivity_init_@V
f MainActivity_init_@VLL
f MainActivity_access$super@LLLL
f MainActivity_Add@III
f MainActivity_Sub@III
f MainActivity_onClick@VL
f MainActivity_onCreate@VL
f DrawableCompat$LollipopDrawableImpl_init_@V
f DrawableCompat$LollipopDrawableImpl_applyTheme@
f DrawableCompat$LollipopDrawableImpl_canApplyThe
```

```
CODE:0013F86C      move-result          v0
CODE:0013F86E      locret:              # CODE XREF: MainActivity
CODE:0013F86E      .line 21
CODE:0013F86E      return               v0
CODE:0013F870      # -----
CODE:0013F870      loc_13F870:          # CODE XREF: MainActivity
CODE:0013F870      sub-int             v0, a, b
CODE:0013F874      goto                locret
CODE:0013F874      Method End
CODE:0013F874      # -----
CODE:0013F876      .byte 0
CODE:0013F877      .byte 0
CODE:0013F878      # =====
CODE:0013F878      # Method 16373 (0x3FF5)
CODE:0013F878      word_13F878:        .short 8
CODE:0013F878      # DATA XREF: CODE:0008A
CODE:0013F878      # Number of registers :
```

减函数对应的指令

```
0013F850  70 20 B3 40 74 00 4D 04 02 03 72 30 A7 3F 10 02 p
0013F860  0C 00 1F 00 0C 08 6E 10 DE 40 00 00 0A 00 0F 00 ..
0013F870  91 00 06 07 28 FD 00 00 08 00 02 00 03 00 00 00 ..
0013F880  AF A7 21 00 3B 00 00 00 62 02 1E 17 38 02 12 00 ..
0013F890  1B 03 2F 36 00 00 12 24 23 44 9A 08 12 05 4D 06 ..
0013F8A0  04 05 12 15 4D 07 04 05 72 30 A7 3F 32 04 0E 00 ..
0013F8B0  12 72 12 63 6E 30 F1 3F 26 03 0A 00 14 02 50 00 .t
0013F8C0  0C 7F 6E 20 F4 3F 26 00 0C 01 1F 01 9B 07 22 02 ..
0013F8D0  15 08 70 10 0D 41 02 00 1B 03 6E 44 00 00 6E 20 ..
```

```
f AppCompatActivityDelegateImplV23_wrapWindowCallback@L
f MainActivity_init_@V
f MainActivity_init_@VLL
f MainActivity_access$super@LLLL
f MainActivity_Add@III
f MainActivity_Sub@III
f MainActivity_onClick@VL
f MainActivity_onCreate@VL
f DrawableCompat$LollipopDrawableImpl_init_@V
f DrawableCompat$LollipopDrawableImpl_applyTheme@
f DrawableCompat$LollipopDrawableImpl_canApplyThe
f DrawableCompat$LollipopDrawableImpl_getColorFilt
f DrawableCompat$LollipopDrawableImpl_inflate@VLLL
f DrawableCompat$LollipopDrawableImpl_setHotspot@
```

```
CODE:0013F806      locret:              # CODE XREF:
CODE:0013F806      .line 17
CODE:0013F806      return               v0
CODE:0013F808      # -----
CODE:0013F808      loc_13F808:          # CODE XREF:
CODE:0013F808      add-int             v0, .
CODE:0013F80C      goto                locret
CODE:0013F80C      Method End
CODE:0013F80C      # -----
CODE:0013F80E      .byte 0, 0
CODE:0013F810      # =====
CODE:0013F810      # Method 16370 (0x3FF2)
0013F808 0013F808: MainActivity_Add@III:loc_13F808 (Synchronized with Hex Vi
```

加函数对应的指令

```
0013F7D8  70 20 B3 40 64 00 4D 04 02 03 12 23 22 04 0E E
0013F7E8  70 20 B3 40 74 00 4D 04 02 03 72 30 A7 3F 10 0E
0013F7F8  0C 00 1F 00 0C 08 6E 10 DE 40 00 00 0A 00 0F 0E
0013F808  90 00 06 07 28 FD 00 00 08 00 03 00 03 00 00 0E
0013F818  A1 A7 21 00 2B 00 00 00 62 00 1E 17 38 00 20 0E
0013F828  1B 01 44 17 00 00 12 32 23 22 9A 08 12 03 40 0E
0013F838  02 03 12 13 22 04 03 08 70 20 B3 40 64 00 40 0E
0013F848  02 03 12 23 22 04 03 08 70 20 B3 40 74 00 40 0E
0013F858  02 03 72 30 A7 3F 10 02 0C 00 1F 00 0C 08 6E 0E
0013F868  DE 40 00 00 0A 00 0F 00 91 00 06 07 28 FD 0E
```

根据函数偏移，在内存中计算函数实际地址

A8A83028+0013F808=A8BC2830

G跳转到内存地址

```
A8BC2828  DE 40 00 00  .@..
A8BC282C  0A 00 0F 00  ....
A8BC2830  90 00 06 07  ....
A8BC2834  28 FD 00 00  (...
A8BC2838  08 00 03 00  ....
A8BC283C  03 00 00 00  ....
A8BC2840  A1 A7 21 00  ..!.
A8BC2844  2B 00 00 00  +...
A8BC2848  62 00 1E 17  b...
A8BC284C  38 00 26 00  8.&.
A8BC2850  1B 01 44 17  ..D.
A8BC2854  00 00 12 32  ...2
A8BC2858  23 22 9A 08  #"..
```

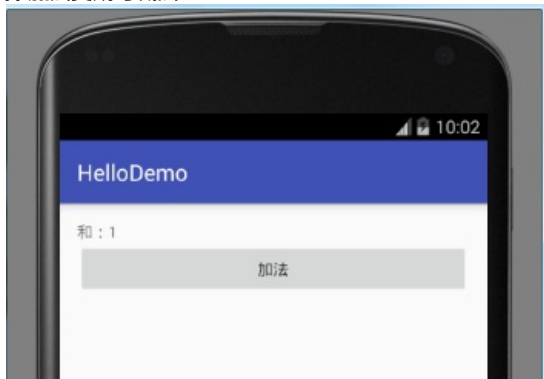
F2将90改为91 再F2保存

```

A8BC282C 0A 00 0F 00 ....
A8BC2830 91 00 06 07 ....
A8BC2834 28 FD 00 00 (...
A8BC2838 08 00 03 00 ....
A8BC283C 03 00 00 00 ....
A8BC2840 A1 A7 21 00 ..!.
A8BC2844 2B 00 00 00 +...
A8BC2848 62 00 1E 17 b...

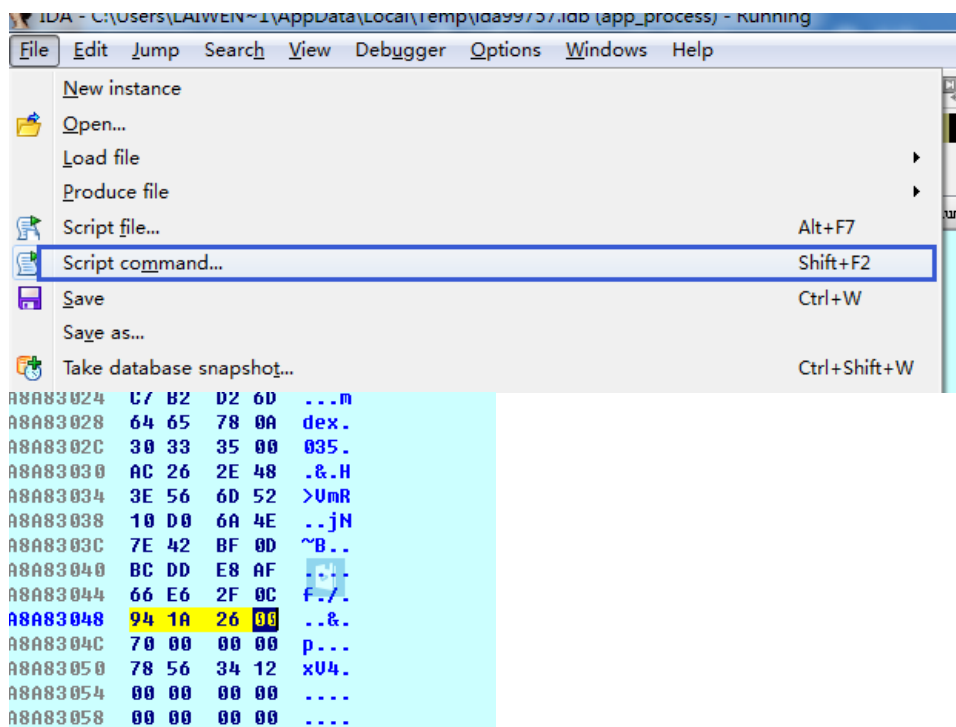
```

//加法变成可减法



成功！

下边dump下来



先取出起始地址，还有文件大小4个字节

起始地址：A8A83028

大小：261A94

static main(void)

```

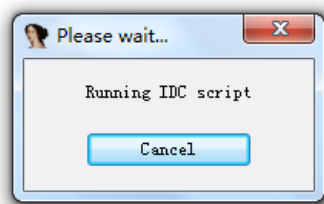
{
    auto fp, begin, end, dexbyte;

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

    begin = 0xA8A83028; //基址
    end = begin + 0x261a94; //大小
    for ( dexbyte = begin; dexbyte < end;    dexbyte ++ )
        fputc(Byte(dexbyte), fp);
}

```

```
}  
  
in = '0xA8A83028'; //基址  
= begin + 0x261a94; //大小  
( dexbyte = begin; dexbyte < end;    dexbyte ++ )  
fputc(Byte(dexbyte), fp);
```



lumn:1

- dump之后可以使用最新版的baksmali反编译dex文件