1. **Kernel proc**

内核修改

检测都是通过 proc 进行的，因此我们需要从 proc 入手

要修改的数据：

/fs/proc/base.c、/fs/proc/array.c

要修改对以下文件的写入：

status, stat

修改点：

base line:285

```
else{
        if(strstr(svmname, “trace”)){
                return sprint(buffer, “%s”, “sys_epoll_wait”);
        }
        return sprint(buffer, “%s”, symname);
}
```

array line: 134

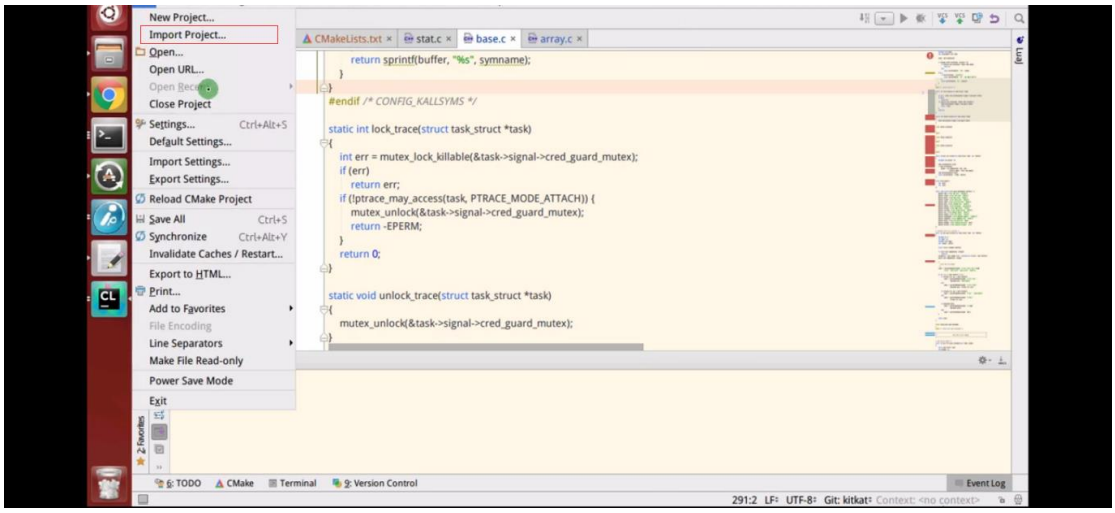
```
static const char * const task_state_array[] = {
        “R (running)”,        /*     0*/
        “S (sleeping)”,       /*     1*/
        “D (disk sleep)”,     /*     2*/
        “S (sleeping)”,       /*     4*/
        “S (sleeping)”,       /*     8*/
        “Z (zombie)”,         /*    16*/
        “X (dead)”,           /*    32*/
        “x (dead)”,           /*    64*/
```

array line 187

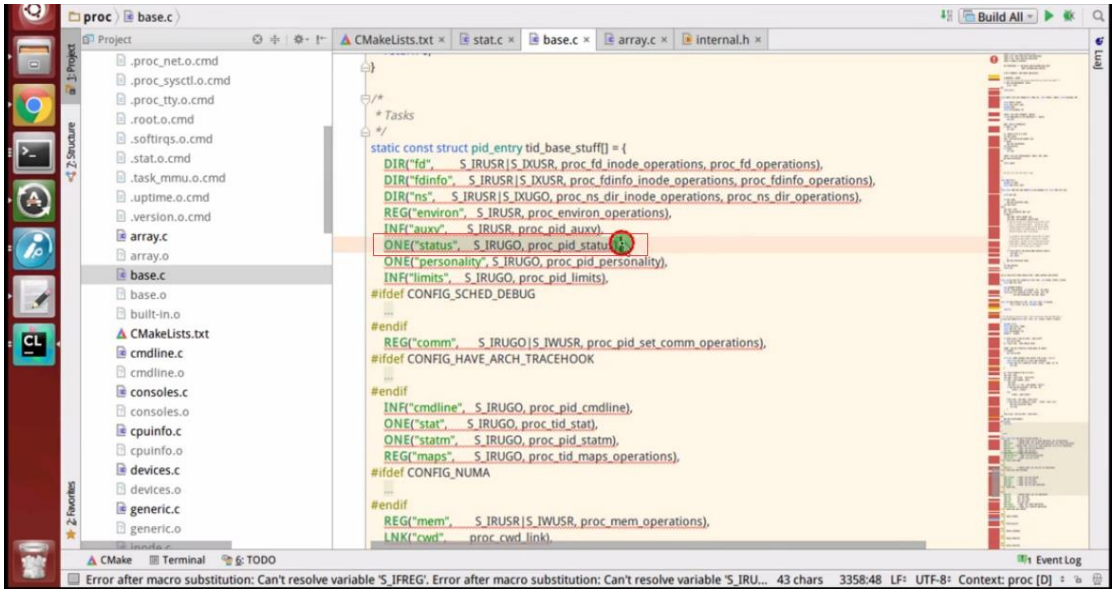
```
“Gid:\t%d\t%d\t%d\t%d\n”,
                get_task_state(p),
                task_tgid_nr_ns(p, ns),
                pid_nr_ns(pid, ns),
                ppid, /*tpid*/0,
                cred->uid, cred->euid, cred->suid, cred->fsuid
                cred->gid, cred->egid, cred->sgid, cred->fsgid);
```

修改完成后，编译内核，刷入系统

2. **Demo 演示**



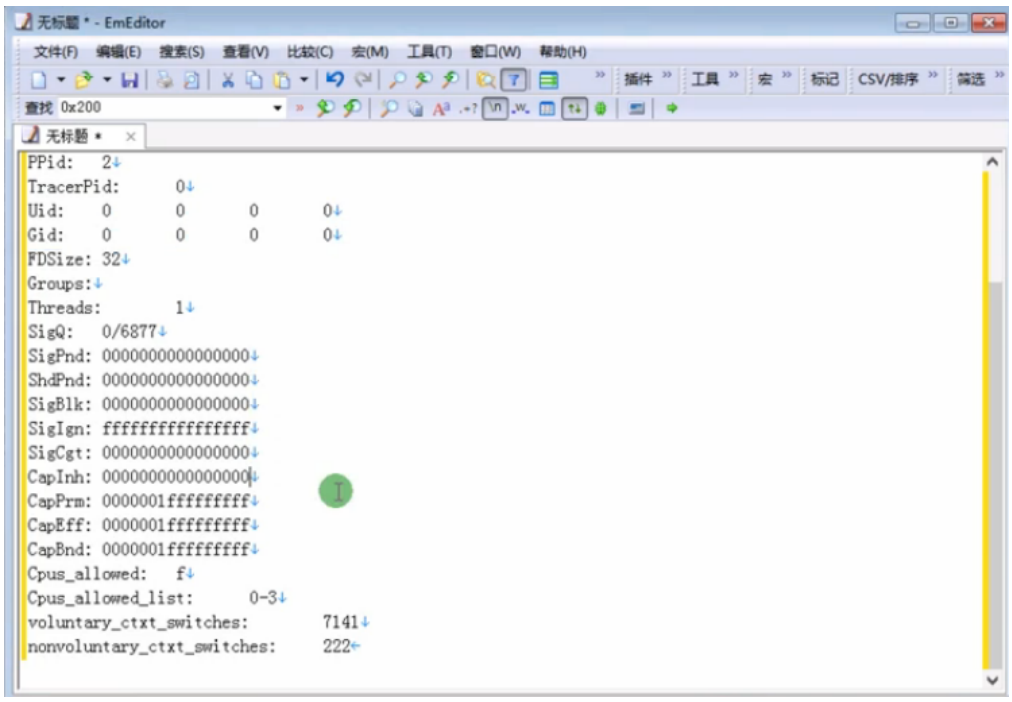
导入内核源代码



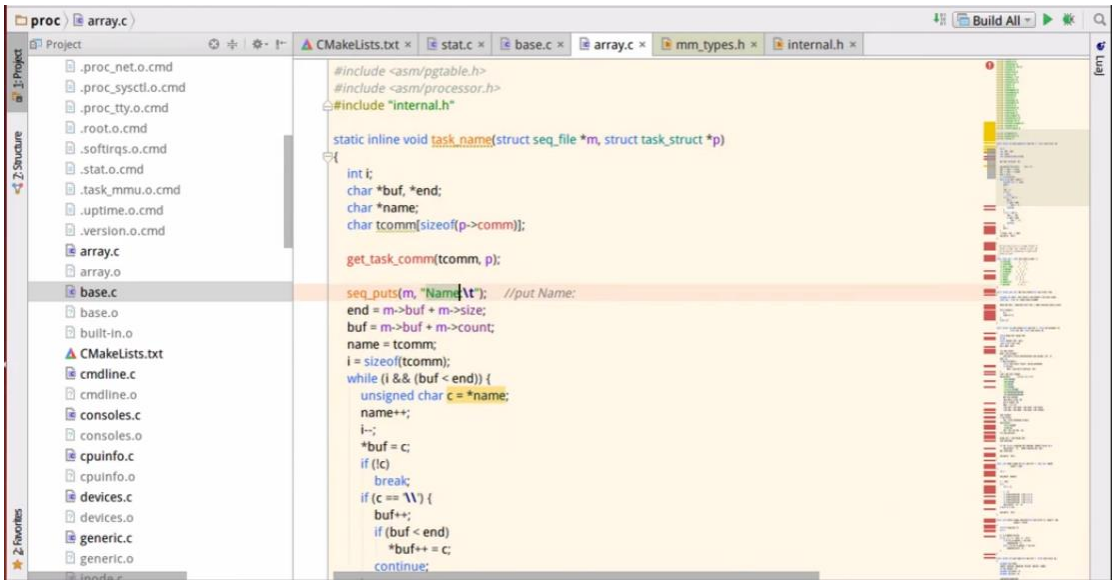
定位到/fs/proc/base.c 文件中找到 **status** 相关的方法

status 文件中的信息如下图

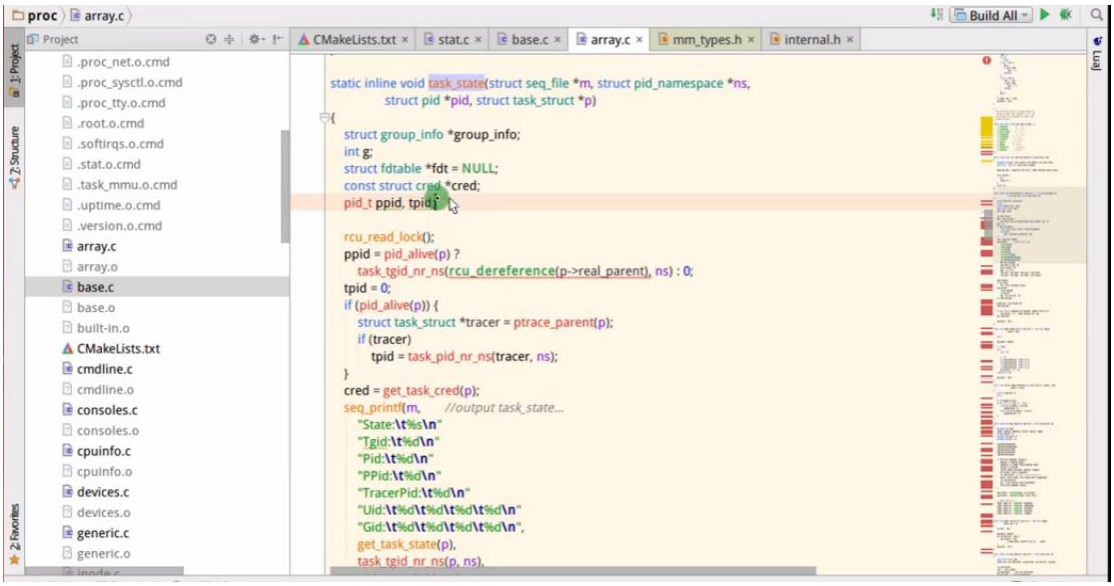
```
Name:   kworker/u8:4
State:  S (sleeping)
Tgid:   5710
```

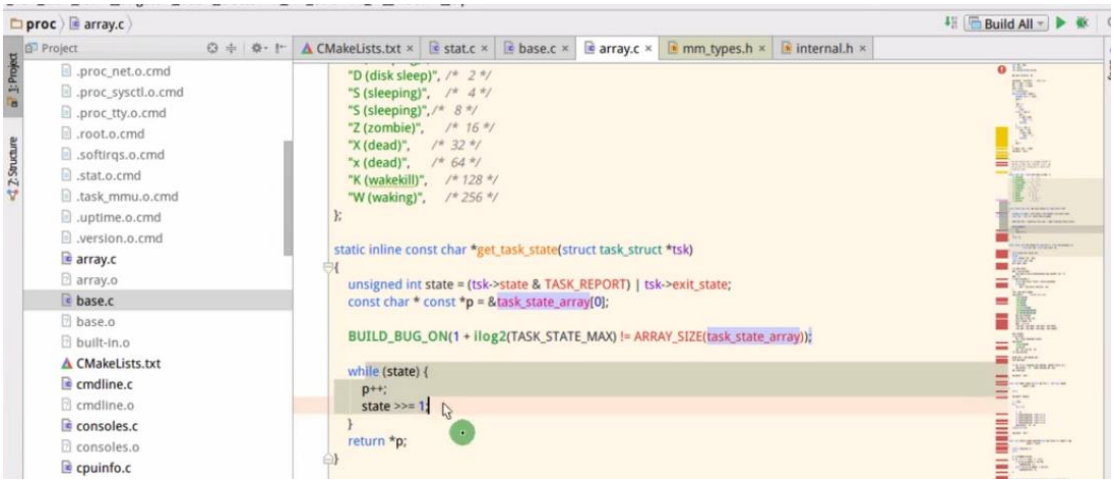


task_name 方法

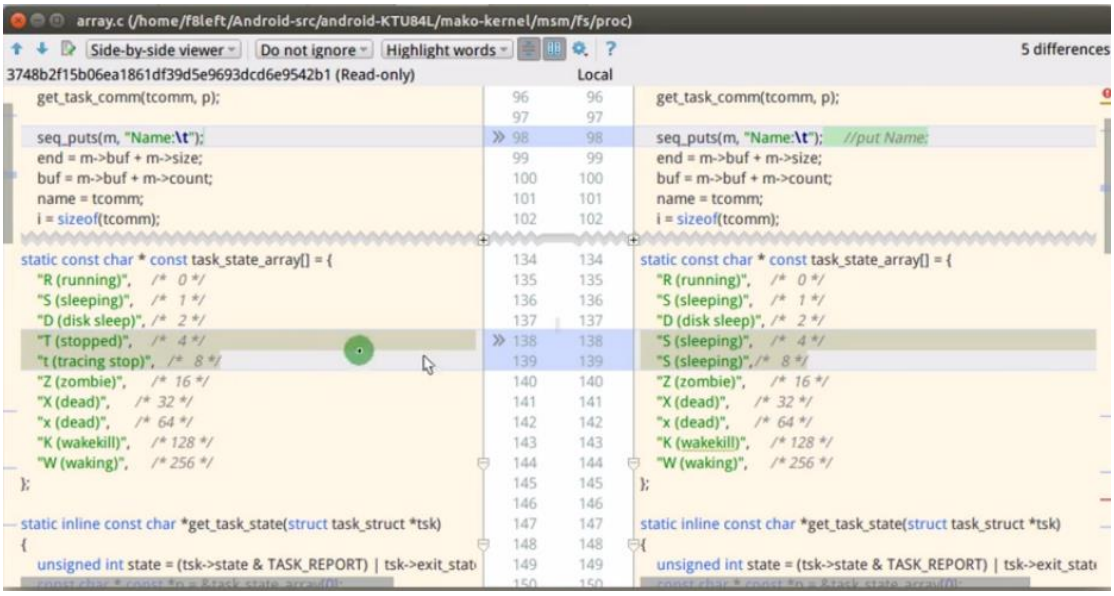


task_state 方法

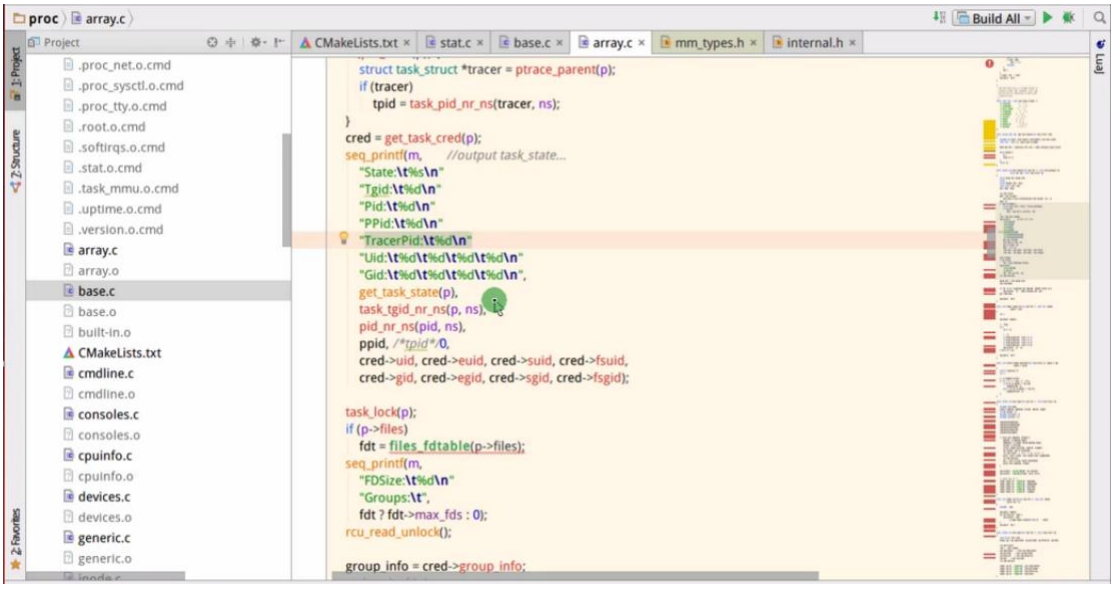




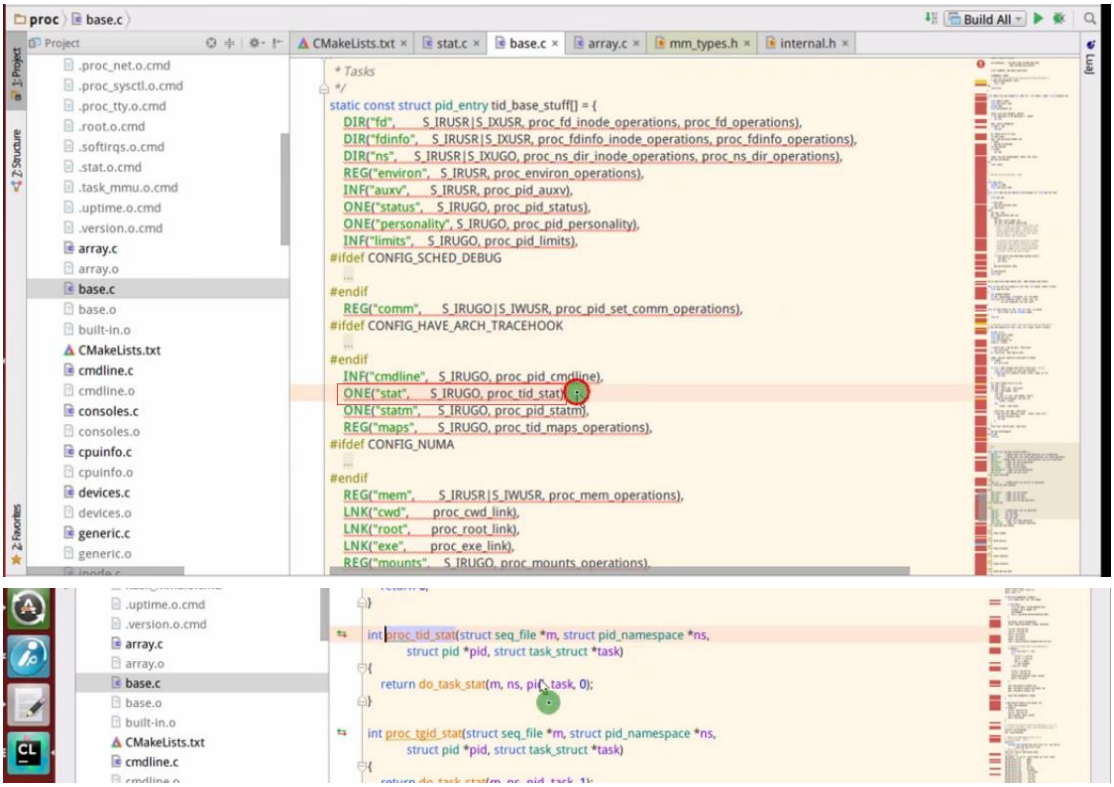
想把 status 数据中的状态信息进行修改的话就要修改 task_status_array 内容（**修改点一**），这样便绕过了 State



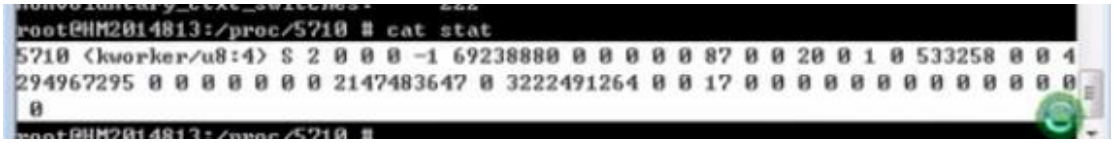
对 TracePid 进行绕过：TracePid 通常都是对父进程 pid 进行检测，这里将 ppid 改为 0，这样不管是否为调试状态，TracePid 都无法检测出



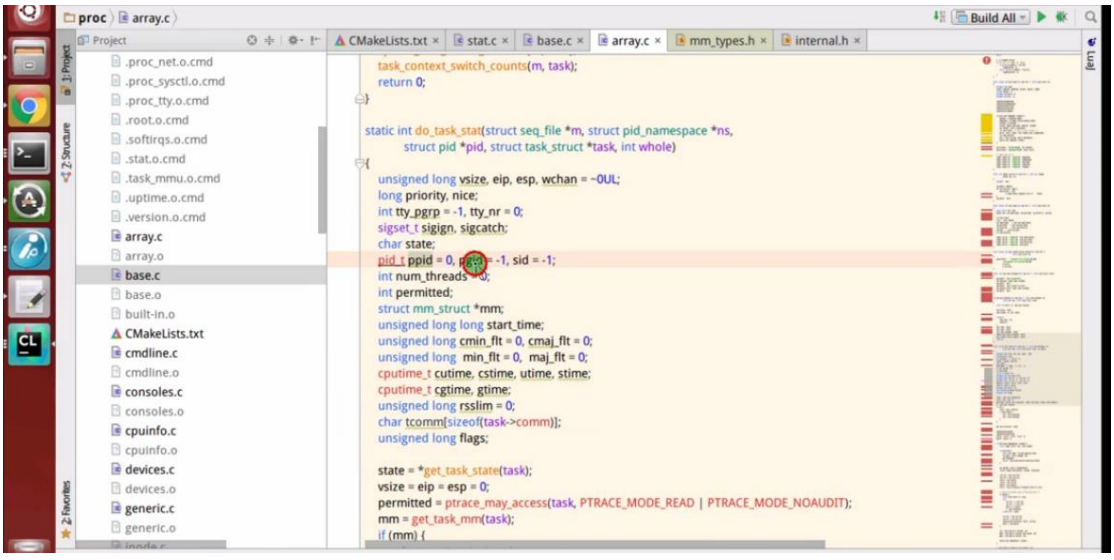
定位到 **Stat** 相关方法



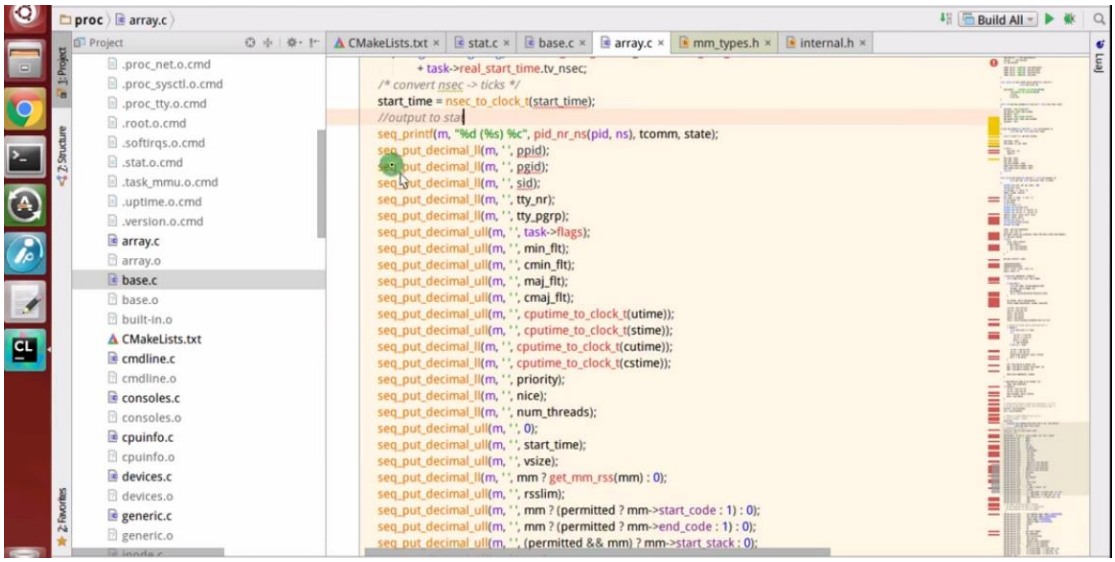
Stat 文件中存放着用户属性的简短描述符



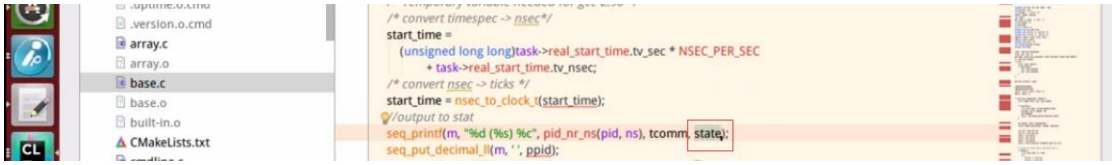
跳转到 do_task_start 方法



定位到写数据的相关代码

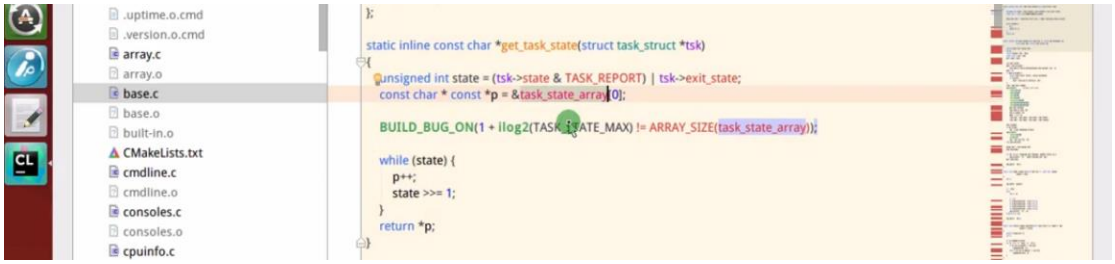


找到 state 的位置

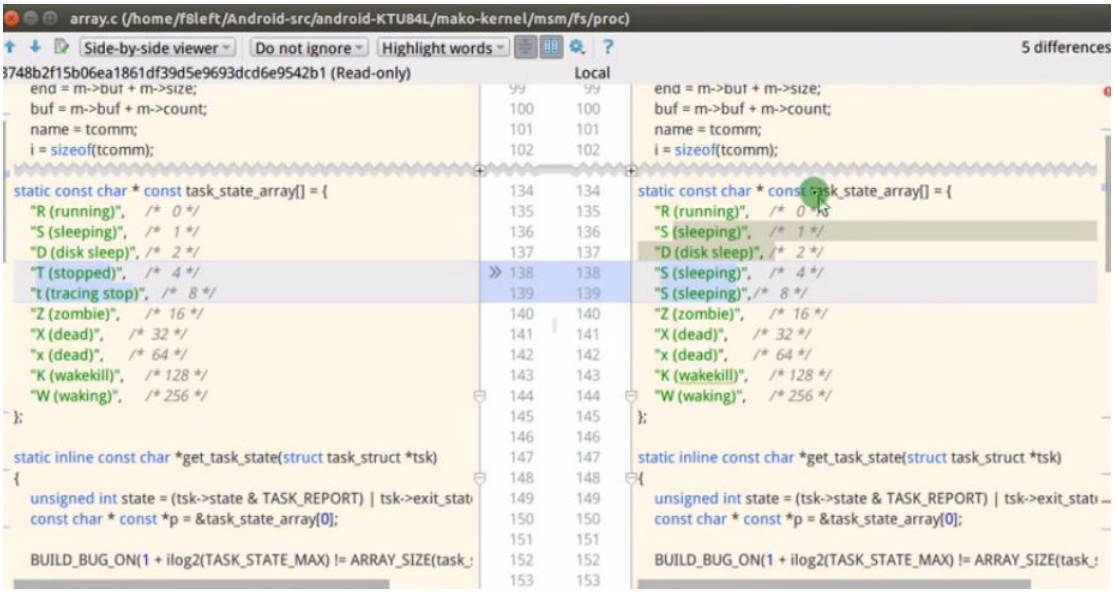


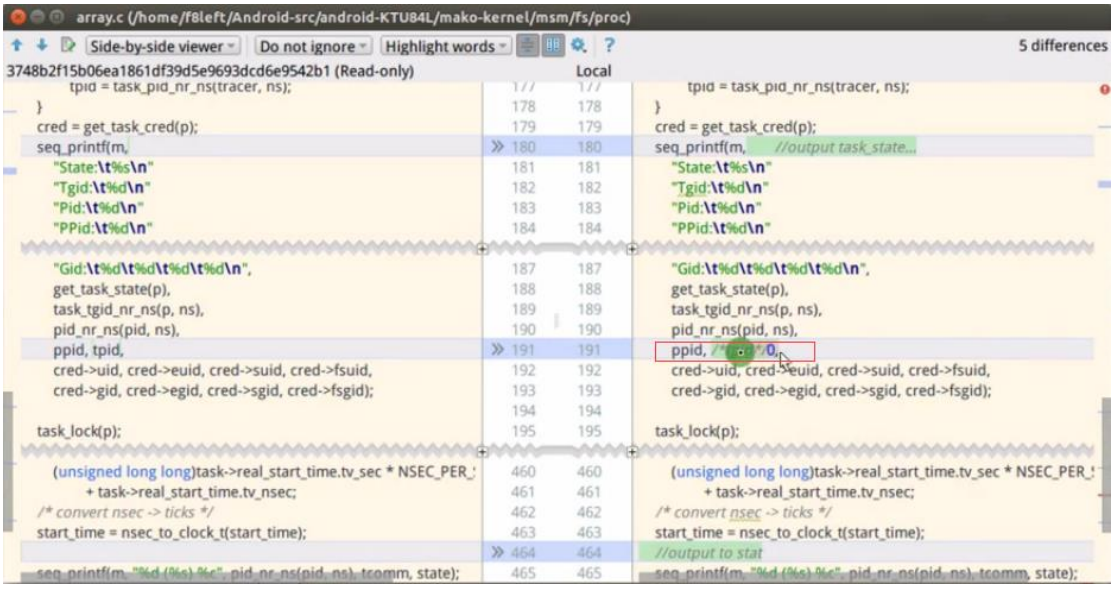
跳转到 state 被赋值的地方

get_task_state 方法



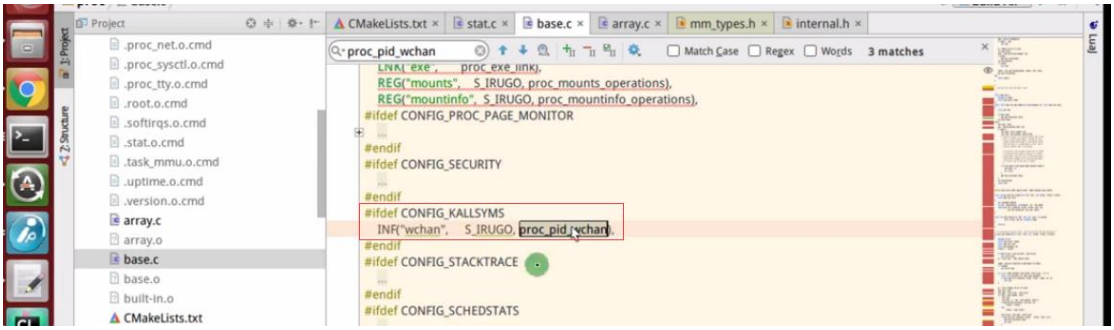
修改相应的 state 数据就会绕过检测

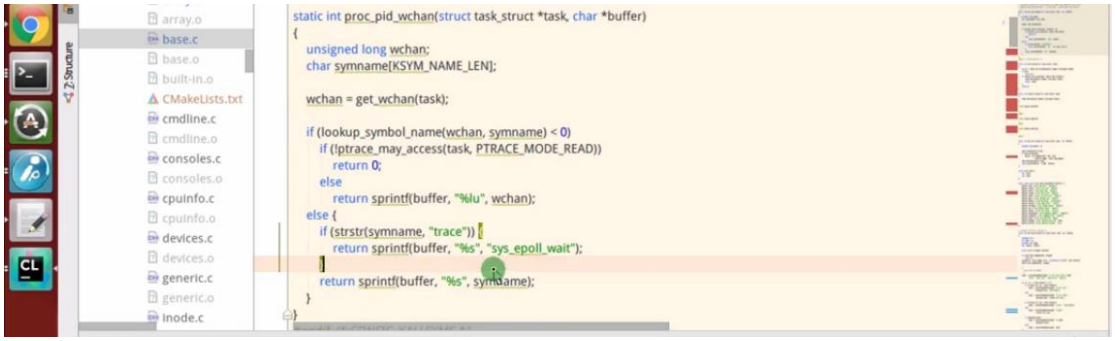


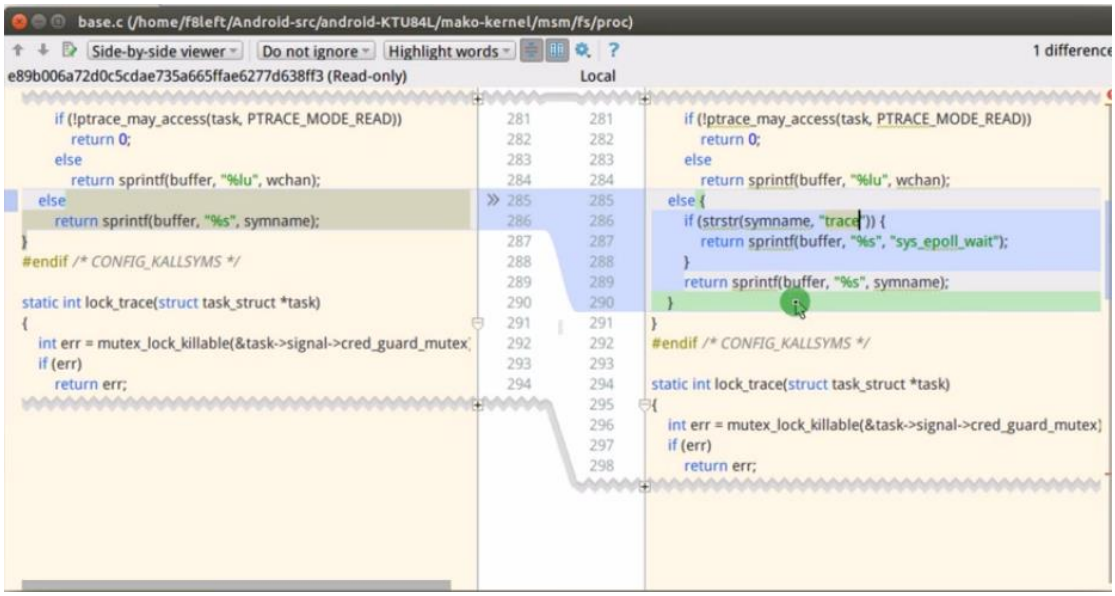


对 wchan 文件的修改

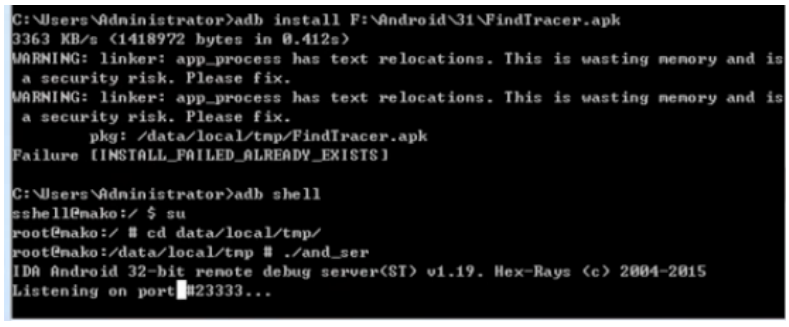
whan 文件如果是在调试过程中会变成 Trace 数据

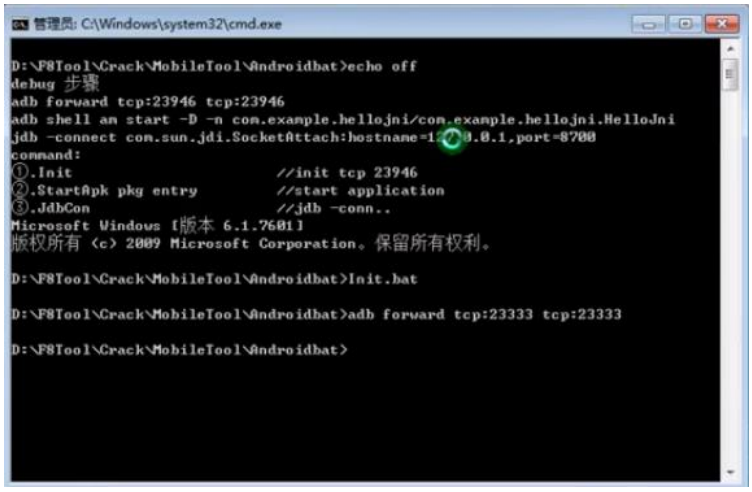


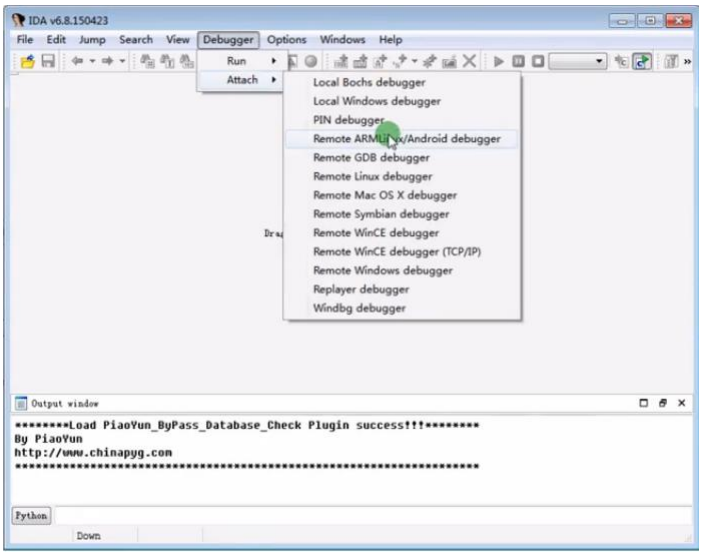

array.o
base.c
base.o
built-in.o
CMakeLists.txt
cmdline.c
cmdline.o
consoles.c
consoles.o
cpuinfo.c
cpuinfo.o
devices.c
devices.o
generic.c
generic.o
inode.c
static int proc_pid_wchan(struct task_struct *task, char *buffer)
{
unsigned long wchan;
char symname[KSYM_NAME_LEN];
wchan = get_wchan(task);
if (lookup_symbol_name(wchan, symname) < 0)
if (!ptrace_may_access(task, PTRACE_MODE_READ))
return 0;
else
return sprintf(buffer, "%lu", wchan);
else {
if (strstr(symname, "trace")) {
return sprintf(buffer, "%s", "sys_epoll_wait");
}
return sprintf(buffer, "%s", symname);
}

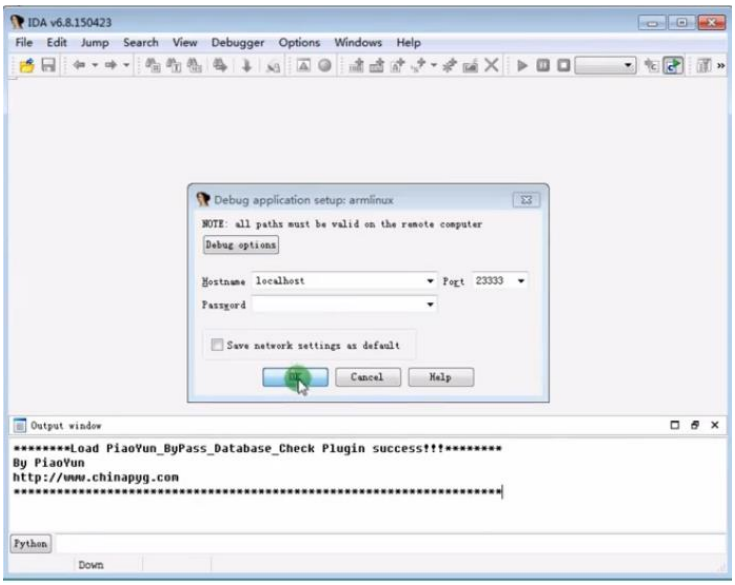

base.c (/home/f8left/Android-src/android-KTU84L/mako-kernel/msm/fs/proc)
Side-by-side viewer
Do not ignore
Highlight words
1 difference
e89b006a72d0c5cdae735a665ffae6277d638ff3 (Read-only)
Local
if (!ptrace_may_access(task, PTRACE_MODE_READ))
return 0;
else
return sprintf(buffer, "%lu", wchan);
else
return sprintf(buffer, "%s", symname);
}
#endif /* CONFIG_KALLSYMS */
static int lock_trace(struct task_struct *task)
{
int err = mutex_lock_killable(&task->signal->cred_guard_mutex
if (err)
return err;
281 281
282 282
283 283
284 284
285 285
286 286
287 287
288 288
289 289
290 290
291 291
292 292
293 293
294 294
295
296
297
298
if (!ptrace_may_access(task, PTRACE_MODE_READ))
return 0;
else
return sprintf(buffer, "%lu", wchan);
else {
if (strstr(symname, "trace")) {
return sprintf(buffer, "%s", "sys_epoll_wait");
}
return sprintf(buffer, "%s", symname);
}
#endif /* CONFIG_KALLSYMS */
static int lock_trace(struct task_struct *task)
{
int err = mutex_lock_killable(&task->signal->cred_guard_mutex)
if (err)
return err;

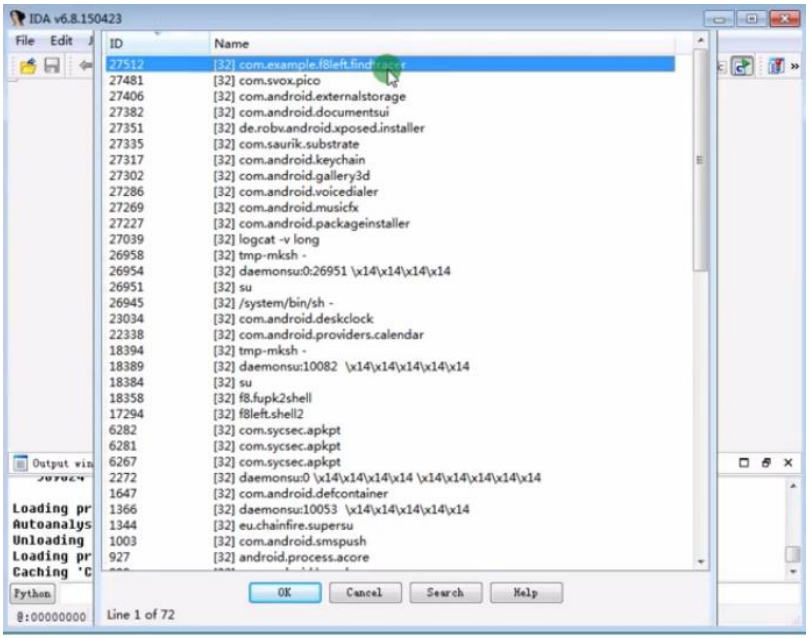
结果验证

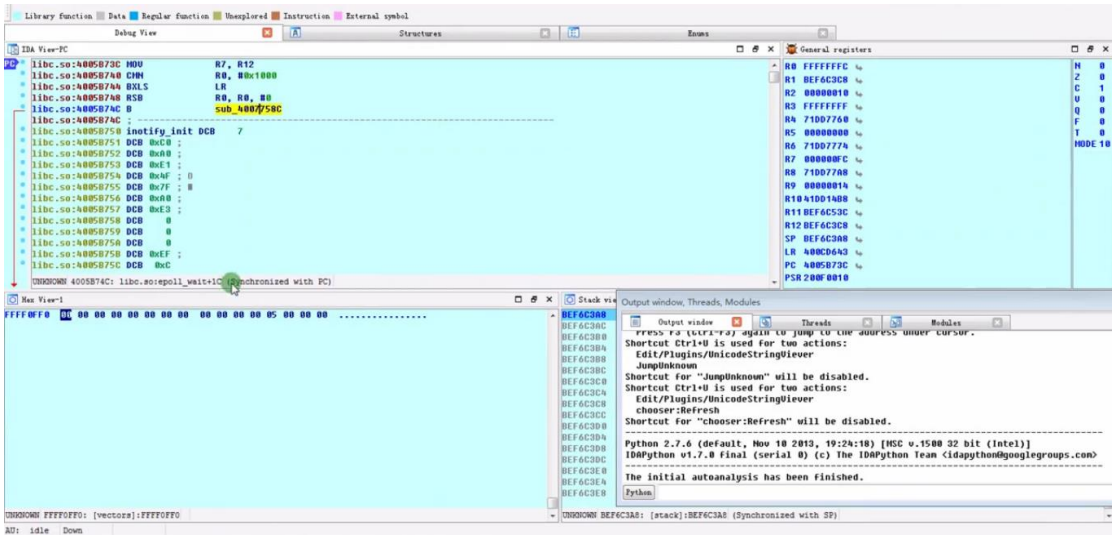

C:\Users\Administrator>adb install F:\Android\31\FindTracer.apk
3363 KB/s (1418972 bytes in 0.412s)
WARNING: linker: app_process has text relocations. This is wasting memory and is
a security risk. Please fix.
WARNING: linker: app_process has text relocations. This is wasting memory and is
a security risk. Please fix.
pkg: /data/local/tmp/FindTracer.apk
Failure [INSTALL_FAILED_ALREADY_EXISTS]
C:\Users\Administrator>adb shell
shell@mako:/ $ su
root@mako:/ # cd data/local/tmp/
root@mako:/data/local/tmp # ./and_ser
IDA Android 32-bit remote debug server(ST) v1.19. Hex-Rays (c) 2004-2015
Listening on port #23333...


管理员: C:\Windows\system32\cmd.exe
D:\F8Tool\Crack\MobileTool\Androidbat>echo off
debug 步骤
adb forward tcp:23946 tcp:23946
adb shell am start -D -n com.example.hellojni/com.example.hellojni.HelloJni
jdb -connect com.sun.jdi.SocketAttach:hostname=127.0.0.1,port=8700
command:
①.Init //init tcp 23946
②.StartApk pkg entry //start application
③.JdbCon //jdb -conn..
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。
D:\F8Tool\Crack\MobileTool\Androidbat>Init.bat
D:\F8Tool\Crack\MobileTool\Androidbat>adb forward tcp:23333 tcp:23333
D:\F8Tool\Crack\MobileTool\Androidbat>


IDA v6.8.150423
File Edit Jump Search View Debugger Options Windows Help
Run
Attach
Local Bochs debugger
Local Windows debugger
PIN debugger
Remote ARMLinux/Android debugger
Remote GDB debugger
Remote Linux debugger
Remote Mac OS X debugger
Remote Symbian debugger
Remote WinCE debugger
Remote WinCE debugger (TCP/IP)
Remote Windows debugger
Replayer debugger
Windbg debugger
Output window
********Load PiaoYun_ByPass_Database_Check Plugin success!!!********
By PiaoYun
http://www.chinapyg.com
Python
Down


IDA v6.8.150423
File Edit Jump Search View Debugger Options Windows Help
Debug application setup: armlinux
NOTE: all paths must be valid on the remote computer
Debug options
Hostname localhost
Port 23333
Password
Save network settings as default
OK
Cancel
Help
Output window
********Load PiaoYun_ByPass_Database_Check Plugin success!!!********
By PiaoYun
http://www.chinapyg.com
Python
Down


IDA v6.8.150423
File Edit
ID Name
27512 [32] com.example.f8left.findtracer
27481 [32] com.svox.pico
27406 [32] com.android.externalstorage
27382 [32] com.android.documentsui
27351 [32] de.robv.android.xposed.installer
27335 [32] com.saurik.substrate
27317 [32] com.android.keychain
27302 [32] com.android.gallery3d
27286 [32] com.android.voicedialer
27269 [32] com.android.musicfx
27227 [32] com.android.packageinstaller
27039 [32] logcat -v long
26958 [32] tmp-mksh -
26954 [32] daemonsu:0:26951 \x14\x14\x14\x14
26951 [32] su
26945 [32] /system/bin/sh -
23034 [32] com.android.deskclock
22338 [32] com.android.providers.calendar
18394 [32] tmp-mksh -
18389 [32] daemonsu:10082 \x14\x14\x14\x14\x14
18384 [32] su
18358 [32] f8.fupk2shell
17294 [32] f8left.shell2
6282 [32] com.sycsec.apkpt
6281 [32] com.sycsec.apkpt
6267 [32] com.sycsec.apkpt
2272 [32] daemonsu:0 \x14\x14\x14\x14 \x14\x14\x14\x14\x14
1647 [32] com.android.defcontainer
1366 [32] daemonsu:10053 \x14\x14\x14\x14\x14
1344 [32] eu.chainfire.supersu
1003 [32] com.android.smspush
927 [32] android.process.acore
OK
Cancel
Search
Help
Line 1 of 72
Loading pr
Autoanalys
Unloading
Loading pr
Caching 'C
Python


Library function Data Regular function Unexplored Instruction External symbol
Debug View
Structures
Enums
IDA View-PC
libc.so:4005B73C MOV R7, R12
libc.so:4005B740 CMN R0, #0x1000
libc.so:4005B744 BXLS LR
libc.so:4005B748 RSB R0, R0, #0
libc.so:4005B74C B sub_4007758C
libc.so:4005B74C ; ---
libc.so:4005B750 inotify_init DCB 7
libc.so:4005B751 DCB 0xC0 ;
libc.so:4005B752 DCB 0xA0 ;
libc.so:4005B753 DCB 0xE1 ;
libc.so:4005B754 DCB 0x4F ; O
libc.so:4005B755 DCB 0x7F ;
libc.so:4005B756 DCB 0x80 ;
libc.so:4005B757 DCB 0xE3 ;
libc.so:4005B758 DCB 0
libc.so:4005B759 DCB 0
libc.so:4005B75A DCB 0
libc.so:4005B75B DCB 0xEF ;
libc.so:4005B75C DCB 0xC
UNKNOWN 4005B74C: libc.so:epoll_wait+1C (Synchronized with PC)
General registers
R0 FFFFFFFC
R1 BEF6C3C8
R2 00000010
R3 FFFFFFFF
R4 71DD7760
R5 00000000
R6 71DD777A
R7 000000FC
R8 71DD77A8
R9 0000001A
R10 41DD1408
R11 BEF6C53C
R12 BEF6C3C8
SP BEF6C3A8
LR 400CD643
PC 4005B73C
PSR 200F0010
N 0
Z 0
C 1
V 0
Q 0
F 0
T 0
MODE 10
Hex View-1
FFFF0FF0 00 00 00 00 00 00 00 00 00 00 00 00 05 00 00 00 ....
UNKNOWN FFFF0FF0: [vectors]:FFFF0FF0
Stack view
BEF6C3A8
BEF6C3AC
BEF6C3B0
BEF6C3B4
BEF6C3B8
BEF6C3BC
BEF6C3C0
BEF6C3C4
BEF6C3C8
BEF6C3CC
BEF6C3D0
BEF6C3D4
BEF6C3D8
BEF6C3DC
BEF6C3E0
BEF6C3E4
BEF6C3E8
UNKNOWN BEF6C3A8: [stack]:BEF6C3A8 (Synchronized with SP)
Output window, Threads, Modules
Output window
Threads
Modules
Shortcut Ctrl+U is used for two actions:
Edit/Plugins/UnicodeStringViewer
JumpUnknown
Shortcut for "JumpUnknown" will be disabled.
Shortcut Ctrl+U is used for two actions:
Edit/Plugins/UnicodeStringViewer
chooser:Refresh
Shortcut for "chooser:Refresh" will be disabled.
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)]
IDAPython v1.7.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
The initial autoanalysis has been finished.
Python
AU: idle Down