
Subject: [PATCH] mounstats NULL pointer dereference
Posted by [Vasily Tarasov](#) on Sat, 18 Nov 2006 12:43:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

OpenVZ developers team has encountered the following problem in 2.6.19-rc6 kernel. After some seconds of running script

```
while [[ 1 ]]
do
  find /proc -name mountstats | xargs cat
done
```

this Oops appears:

```
BUG: unable to handle kernel NULL pointer dereference at virtual address
00000010
printing eip:
c01a6b70
*pde = 00000000
Oops: 0000 [#1]
SMP
Modules linked in: xt_length ipt_ttl xt_tcpmss ipt_TCPMSS iptable_mangle
iptable_filter xt_multiport xt_limit ipt_tos ipt_REJECT ip_tables x_tables
parport_pc lp parport sunrpc af_packet thermal processor fan button battery
asus_acpi ac ohci_hcd ehci_hcd usbcore i2c_nforce2 i2c_core tg3 floppy
pata_amd
ide_cd cdrom sata_nv libata
CPU: 1
EIP: 0060:[<c01a6b70>] Not tainted VLI
EFLAGS: 00010246 (2.6.19-rc6 #2)
EIP is at mountstats_open+0x70/0xf0
eax: 00000000 ebx: e6247030 ecx: e62470f8 edx: 00000000
esi: 00000000 edi: c01a6b00 ebp: c33b83c0 esp: f4105eb4
ds: 007b es: 007b ss: 0068
Process cat (pid: 6044, ti=f4105000 task=f4104a70 task.ti=f4105000)
Stack: c33b83c0 c04ee940 f46a4a80 c33b83c0 e4df31b4 c01a6b00 f4105000 c0169231
e4df31b4 c33b83c0 c33b83c0 f4105f20 00000003 f4105000 c0169445 f2503cf0
f7f8c4c0 00008000 c33b83c0 00000000 00008000 c0169350 f4105f20 00008000
Call Trace:
[<c01a6b00>] mountstats_open+0x0/0xf0
[<c0169231>] __dentry_open+0x181/0x250
[<c0169445>] nameidata_to_filp+0x35/0x50
[<c0169350>] do_filp_open+0x50/0x60
[<c01873d6>] seq_read+0xc6/0x300
[<c0169511>] get_unused_fd+0x31/0xc0
[<c01696d3>] do_sys_open+0x63/0x110
[<c01697a7>] sys_open+0x27/0x30
[<c01030bd>] sysenter_past_esp+0x56/0x79
```

=====

Code: 45 74 8b 54 24 20 89 44 24 08 8b 42 f0 31 d2 e8 47 cb f8 ff 85 c0 89 c3
74 51 8d 80 a0 04 00 00 e8 46 06 2c 00 8b 83 48 04 00 00 <8b> 78 10 85 ff 74
03
f0 ff 07 b0 01 86 83 a0 04 00 00 f0 ff 4b
EIP: [<c01a6b70>] mountstats_open+0x70/0xf0 SS:ESP 0068:f4105eb4

The problem is that task->nsproxy can be equal NULL for some time during task exit. This patch fixes the BUG.

Signed-off-by: Vasily Tarasov <vtaras@openvz.org>

--

--- ./fs/proc/base.c.mountstatsfix 2006-11-18 02:43:48.000000000 +0300
+++ ./fs/proc/base.c 2006-11-18 03:11:41.000000000 +0300
@@ -442,7 +442,8 @@

```
    if (task) {  
        task_lock(task);  
- namespace = task->nsproxy->namespace;  
+ if (task->nsproxy)  
+ namespace = task->nsproxy->namespace;  
    if (namespace)  
        get_namespace(namespace);  
    task_unlock(task);
```

--- ./include/linux/nsproxy.h.mountstatsfix 2006-11-18 02:43:51.000000000 +0300
+++ ./include/linux/nsproxy.h 2006-11-18 03:15:25.000000000 +0300
@@ -45,8 +45,10 @@

```
{  
    struct nsproxy *ns = p->nsproxy;  
    if (ns) {  
- put_nsproxy(ns);  
+ task_lock(p);  
    p->nsproxy = NULL;  
+ task_unlock(p);  
+ put_nsproxy(ns);  
    }  
}  
#endif
```