Subject: [PATCH] proc: fix ->open'less usage due to ->proc_fops flip
Posted by Alexey Dobriyan on Sat, 29 Dec 2007 12:45:26 GMT
View Forum Message <> Reply to Message

NOTE, NOTE, NOTE:

please, drop proc-remove-useless-checks-in-proc_register.patch
before applying this one!

[PATCH] proc: fix ->open'less usage due to ->proc_fops flip

Typical PDE creation code looks like:

```
 pde = create_proc_entry("foo", 0, NULL);
 if (pde)
  pde->proc_fops = &foo_proc_fops;
```

Notice that PDE is first created, only then ->proc_fops is set up to
final value. This is a problem because right after creation
a) PDE is fully visible in /proc , and
b) ->proc_fops are proc_file_operations which do not have ->open callback. So, it's
   possible to ->read without ->open (see one class of oopses below).

The fix is new API called proc_create() which makes sure ->proc_fops are
set up before gluing PDE to main tree. Typical new code looks like:

```
 pde = proc_create("foo", 0, NULL, &foo_proc_fops);
 if (!pde)
  return -ENOMEM;
```

Fix most networking users for a start.

In the long run, create_proc_entry() for regular files will go.

BUG: unable to handle kernel NULL pointer dereference at virtual address 00000024
printing eip: c1188c1b *pdpt = 000000002929e001 *pde = 0000000000000000
Oops: 0002 [#1] PREEMPT SMP DEBUG_PAGEALLOC
last sysfs file: /sys/block/sda/sda1/dev
Modules linked in: foo af_packet ipv6 cpufreq_ondemand loop serio_raw psmouse k8temp
hwmon sr_mod cdrom

Pid: 24679, comm: cat Not tainted (2.6.24-rc3-mm1 #2)
EIP: 0060:[<c1188c1b>] EFLAGS: 00210002 CPU: 0
EIP is at mutex_lock_nested+0x75/0x25d
EAX: 000006fe EBX: fffffffb ECX: 00001000 EDX: e9340570
ESI: 00000020 EDI: 00200246 EBP: e9340570 ESP: e8ea1ef8

```
 DS: 007b ES: 007b FS: 00d8 GS: 0033 SS: 0068
Process cat (pid: 24679, ti=E8EA1000 task=E9340570 task.ti=E8EA1000)
Stack: 00000000 c106f7ce e8ee05b4 00000000 00000001 458003d0 f6fb6f20 fffffffb
       00000000 c106f7aa 00001000 c106f7ce 08ae9000 f6db53f0 00000020 00200246
       00000000 00000002 00000000 00200246 00200246 e8ee05a0 fffffffb e8ee0550
Call Trace:
 [<c106f7ce>] seq_read+0x24/0x28a
 [<c106f7aa>] seq_read+0x0/0x28a
 [<c106f7ce>] seq_read+0x24/0x28a
 [<c106f7aa>] seq_read+0x0/0x28a
 [<c10818b8>] proc_reg_read+0x60/0x73
 [<c1081858>] proc_reg_read+0x0/0x73
 [<c105a34f>] vfs_read+0x6c/0x8b
 [<c105a6f3>] sys_read+0x3c/0x63
 [<c10025f2>] sysenter_past_esp+0x5f/0xa5
 [<c10697a7>] destroy_inode+0x24/0x33
 =======================
INFO: lockdep is turned off.
Code: 75 21 68 e1 1a 19 c1 68 87 00 00 00 68 b8 e8 1f c1 68 25 73 1f c1 e8 84 06 e9 ff e8 52 b8
e7 ff 83 c4 10 9c 5f fa e8 28 89 ea ff <f0> fe 4e 04 79 0a f3 90 80 7e 04 00 7e f8 eb f0 39 76 34
74 33
EIP: [<c1188c1b>] mutex_lock_nested+0x75/0x25d SS:ESP 0068:e8ea1ef8
```

Signed-off-by: Alexey Dobriyan <adobriyan@sw.ru>

```
 fs/proc/generic.c       |   40 ++++++++++++++++++++++++++++++++++++++----
 fs/proc/proc_net.c      |    7 +------
 fs/proc/root.c          |    1 +
 include/linux/proc_fs.h |    6 +++++-
 4 files changed, 43 insertions(+), 11 deletions(-)

--- a/fs/proc/generic.c
+++ b/fs/proc/generic.c
@@ -558,7 +558,7 @@ static int proc_register(struct proc_dir_entry * dir, struct proc_dir_entry *
dp
  return 0;
 }

-static struct proc_dir_entry *proc_create(struct proc_dir_entry **parent,
+static struct proc_dir_entry *__proc_create(struct proc_dir_entry **parent,
        const char *name,
        mode_t mode,
        nlink_t nlink)
@@ -601,7 +601,7 @@ struct proc_dir_entry *proc_symlink(const char *name,
 {
  struct proc_dir_entry *ent;
```

```
- ent = proc_create(&parent,name,
+ ent = __proc_create(&parent,name,
      (S_IFLNK | S_IRUGO | S_IWUGO | S_IXUGO),1);

  if (ent) {
@@ -626,7 +626,7 @@ struct proc_dir_entry *proc_mkdir_mode(const char *name, mode_t
mode,
 {
  struct proc_dir_entry *ent;

- ent = proc_create(&parent, name, S_IFDIR | mode, 2);
+ ent = __proc_create(&parent, name, S_IFDIR | mode, 2);
  if (ent) {
   if (proc_register(parent, ent) < 0) {
    kfree(ent);
@@ -660,7 +660,7 @@ struct proc_dir_entry *create_proc_entry(const char *name, mode_t
mode,
   nlink = 1;
  }

- ent = proc_create(&parent,name,mode,nlink);
+ ent = __proc_create(&parent,name,mode,nlink);
  if (ent) {
   if (proc_register(parent, ent) < 0) {
    kfree(ent);
@@ -670,6 +670,38 @@ struct proc_dir_entry *create_proc_entry(const char *name, mode_t
mode,
  return ent;
 }

+struct proc_dir_entry *proc_create(const char *name, mode_t mode,
+       struct proc_dir_entry *parent,
+       const struct file_operations *proc_fops)
+{
+ struct proc_dir_entry *pde;
+ nlink_t nlink;
+
+ if (S_ISDIR(mode)) {
+  if ((mode & S_IALLUGO) == 0)
+   mode |= S_IRUGO | S_IXUGO;
+  nlink = 2;
+ } else {
+  if ((mode & S_IFMT) == 0)
+   mode |= S_IFREG;
+  if ((mode & S_IALLUGO) == 0)
+   mode |= S_IRUGO;
+  nlink = 1;
+ }
```

```
+
+ pde = __proc_create(&parent, name, mode, nlink);
+ if (!pde)
+  goto out;
+ pde->proc_fops = proc_fops;
+ if (proc_register(parent, pde) < 0)
+  goto out_free;
+ return pde;
+out_free:
+ kfree(pde);
+out:
+ return NULL;
+}
+
 void free_proc_entry(struct proc_dir_entry *de)
 {
  unsigned int ino = de->low_ino;
--- a/fs/proc/proc_net.c
+++ b/fs/proc/proc_net.c
@@ -29,12 +29,7 @@
 struct proc_dir_entry *proc_net_fops_create(struct net *net,
  const char *name, mode_t mode, const struct file_operations *fops)
 {
- struct proc_dir_entry *res;
-
- res = create_proc_entry(name, mode, net->proc_net);
- if (res)
-  res->proc_fops = fops;
- return res;
+ return proc_create(name, mode, net->proc_net, fops);
 }
 EXPORT_SYMBOL_GPL(proc_net_fops_create);

--- a/fs/proc/root.c
+++ b/fs/proc/root.c
@@ -232,6 +232,7 @@ void pid_ns_release_proc(struct pid_namespace *ns)
 EXPORT_SYMBOL(proc_symlink);
 EXPORT_SYMBOL(proc_mkdir);
 EXPORT_SYMBOL(create_proc_entry);
+EXPORT_SYMBOL(proc_create);
 EXPORT_SYMBOL(remove_proc_entry);
 EXPORT_SYMBOL(proc_root);
 EXPORT_SYMBOL(proc_root_fs);
--- a/include/linux/proc_fs.h
+++ b/include/linux/proc_fs.h
@@ -124,6 +124,7 @@ void de_put(struct proc_dir_entry *de);

 extern struct proc_dir_entry *create_proc_entry(const char *name, mode_t mode,
```

```
       struct proc_dir_entry *parent);
+struct proc_dir_entry *proc_create(const char *name, mode_t mode, struct proc_dir_entry
*parent, const struct file_operations *proc_fops);
 extern void remove_proc_entry(const char *name, struct proc_dir_entry *parent);

 extern struct vfsmount *proc_mnt;
@@ -216,7 +217,10 @@ static inline void proc_flush_task(struct task_struct *task)

 static inline struct proc_dir_entry *create_proc_entry(const char *name,
  mode_t mode, struct proc_dir_entry *parent) { return NULL; }
-
+static inline struct proc_dir_entry *proc_create(const char *name, mode_t mode, struct
proc_dir_entry *parent, const struct file_operations *proc_fops)
+{
+ return NULL;
+}
 #define remove_proc_entry(name, parent) do {} while (0)

 static inline struct proc_dir_entry *proc_symlink(const char *name,
```