Subject: stat on /proc/<PID>/exe when <PID> is zombie inside a VE
Posted by Ivan Dubrov on Sat, 12 Jan 2008 14:44:45 GMT
View Forum Message <> Reply to Message

Hi,

Please ignore my previous mail.

I was investigating why stat on /proc/<PID>/exe fails with EACCES when <PID> is zombie. In short, this situation is quite often then starting services on openSUSE 10.3 VE (startproc does stat on its children for some reason and sometimes this children is already zombie). That results in multiple "startproc: cannot stat /proc/1128/exe: Permission denied" when starting the service.

So, I've traced down the source of this error and found that it occurs in kernel/ptrace.c may_attach() function. If process is zombie, it has empty task->mm, so vps_dumpable is 0 for such process. As a result, if VE is not a super VE, the check fails.

Here is the corresponding piece of code (may_attach.c, around .. line):

```
if (task->mm) {
  dumpable = task->mm->dumpable;
  vps_dumpable = (task->mm->vps_dumpable == 1);
}

if (!dumpable && !capable(CAP_SYS_PTRACE)) // #1
  return -EPERM;
if (!vps_dumpable && !ve_is_super(get_exec_env())) // This check fails
if process is zombie
  return -EPERM;
```

The questions here is it safe to allow the "ptrace" if process is zombie? It seems to me that this should be perfectly safe. Anyway, the actual ptrace_attach will fail on task with empty mm, so this only affects /proc behavior.

On the other hand, maybe this is a startproc issue and not the kernel one? It seems that in regular environment it works only because it is usually executed under "root" account which has CAP_SYS_PTRACE capability and therefore check #1 fails for zombies.

I've attached a patch that fixes startproc. It skips the check if (task->exit_state&EXIT_ZOMBIE) is true.

--

WBR,
Ivan S. Dubrov

Subject: Re: stat on /proc/<PID>/exe when <PID> is zombie inside a VE
Posted by dev on Sat, 12 Jan 2008 15:28:42 GMT

View Forum Message <> Reply to Message

Ouch... Looks like vps_dumpable flag should be moved to task_struct
to fix it properly...

Your fix is incorrect for a number of reasons:
1. it introduces a race: in do_exit() task first looses it's mm, only then
   it's exit_state is set to EXIT_ZOMBIE, so there is still a window when it will
   return EACCESS.
2. It's illogical, since the idea was to protect tasks which
   did VE_ENTER from looking at them in some proc files and ptrace.
   You open this barrier allowing to investigate state of zombie
   such entered process.

So I think we simply have to move vps_dumpable flag to task_struct...
As a temporary workaround you can simply drop this check in your kernel
for some time until we fixed it.

Thanks,
Kirill

Ivan Dubrov wrote:
> Hi,
>
> Please ignore my previous mail.
>
> I was investigating why stat on /proc/<PID>/exe fails with EACCES when
> <PID> is zombie. In short, this situation is quite often then starting
> services on openSUSE 10.3 VE (startproc does stat on its children for
> some reason and sometimes this children is already zombie). That results
> in multiple "startproc: cannot stat /proc/1128/exe: Permission denied"
> when starting the service.
>
> So, I've traced down the source of this error and found that it occurs
> in kernel/ptrace.c may_attach() function. If process is zombie, it has
> empty task->mm, so vps_dumpable is 0 for such process. As a result, if
> VE is not a super VE, the check fails.
>
> Here is the corresponding piece of code (may_attach.c, around .. line):
>
> if (task->mm) {
>   dumpable = task->mm->dumpable;

```
>   vps_dumpable = (task->mm->vps_dumpable == 1);
> }
>
> if (!dumpable && !capable(CAP_SYS_PTRACE)) // #1
>   return -EPERM;
> if (!vps_dumpable && !ve_is_super(get_exec_env())) // This check fails
> if process is zombie
>   return -EPERM;
>
>
> The questions here is it safe to allow the "ptrace" if process is
> zombie? It seems to me that this should be perfectly safe. Anyway, the
> actual ptrace_attach will fail on task with empty mm, so this only
> affects /proc behavior.
>
> On the other hand, maybe this is a startproc issue and not the kernel
> one? It seems that in regular environment it works only because it is
> usually executed under "root" account which has CAP_SYS_PTRACE
> capability and therefore check #1 fails for zombies.
>
> I've attached a patch that fixes startproc. It skips the check if
> (task->exit_state&EXIT_ZOMBIE) is true.
>
>
>
> ------------------------------------------------------------------------
>
> --- linux-2.6.22.orig/kernel/ptrace.c 2008-01-12 20:25:24.000000000 +0600
> +++ linux-2.6.22/kernel/ptrace.c 2008-01-12 20:25:58.000000000 +0600
> @@ -151,7 +151,8 @@
>
>   if (!dumpable && !capable(CAP_SYS_PTRACE))
>    return -EPERM;
> - if (!vps_dumpable && !ve_is_super(get_exec_env()))
> + if (!vps_dumpable && !ve_is_super(get_exec_env())
> +   && !(task->exit_state & EXIT_ZOMBIE))
>    return -EPERM;
>   if (!ve_accessible(VE_TASK_INFO(task)->owner_env, get_exec_env()))
>    return -EPERM;
>
>
> ------------------------------------------------------------------------
>
```