
Subject: [PATCH] /proc/*/environ: wrong placing of ptrace_may_attach() check
Posted by [Alexey Dobriyan](#) on Mon, 28 May 2007 13:33:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

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

fs/proc/base.c | 11 ++++++----
1 file changed, 8 insertions(+), 3 deletions(-)

```
--- a/fs/proc/base.c
+++ b/fs/proc/base.c
@@ -204,12 +204,17 @@ static int proc_pid_environ(struct task_
    int res = 0;
    struct mm_struct *mm = get_task_mm(task);
    if (mm) {
-     unsigned int len = mm->env_end - mm->env_start;
+     unsigned int len;
+
+     +
+     + res = -ESRCH;
+     + if (!ptrace_may_attach(task))
+     + goto out;
+     +
+     + len = mm->env_end - mm->env_start;
        if (len > PAGE_SIZE)
            len = PAGE_SIZE;
        res = access_process_vm(task, mm->env_start, buffer, len, 0);
-     if (!ptrace_may_attach(task))
-     res = -ESRCH;
+out:
+     mmput(mm);
}
return res;
```

Subject: Re: [PATCH] /proc/*/environ: wrong placing of ptrace_may_attach() check
Posted by [Andrew Morton](#) on Wed, 30 May 2007 00:16:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Mon, 28 May 2007 17:41:57 +0400

Alexey Dobriyan <adobriyan@sw.ru> wrote:

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

Better changelogs, please.

> --- a/fs/proc/base.c

> +++ b/fs/proc/base.c

```

> @@ -204,12 +204,17 @@ static int proc_pid_environ(struct task_
>     int res = 0;
>     struct mm_struct *mm = get_task_mm(task);
>     if (mm) {
> -    unsigned int len = mm->env_end - mm->env_start;
> +    unsigned int len;
> +
> +    res = -ESRCH;
> +    if (!ptrace_may_attach(task))
> +        goto out;
> +
> +    len = mm->env_end - mm->env_start;
>     if (len > PAGE_SIZE)
>         len = PAGE_SIZE;
>     res = access_process_vm(task, mm->env_start, buffer, len, 0);
> -    if (!ptrace_may_attach(task))
> -        res = -ESRCH;
> +out:
>     mmput(mm);
> }
> return res;

```

What's wrong with the existing code? It's a bit dopey-looking and can, I guess, permit a task to cause a pagefault in an mm which it doesn't have permission to read from. But is there some more serious problem being fixed here?

I shouldn't have to ask this stuff.

Subject: Re: [PATCH] /proc/*/environ: wrong placing of ptrace_may_attach() check
Posted by [Alexey Dobriyan](#) on Wed, 30 May 2007 08:31:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, May 29, 2007 at 05:16:23PM -0700, Andrew Morton wrote:

> On Mon, 28 May 2007 17:41:57 +0400
> Alexey Dobriyan <adobriyan@sw.ru> wrote:

```

> > --- a/fs/proc/base.c
> > +++ b/fs/proc/base.c
> > @@ -204,12 +204,17 @@ static int proc_pid_environ(struct task_
> >     int res = 0;
> >     struct mm_struct *mm = get_task_mm(task);
> >     if (mm) {
> > -    unsigned int len = mm->env_end - mm->env_start;
> > +    unsigned int len;
> > +
> > +    res = -ESRCH;

```

```
> > + if (!ptrace_may_attach(task))
> > + goto out;
> > +
> > + len = mm->env_end - mm->env_start;
> >     if (len > PAGE_SIZE)
> >         len = PAGE_SIZE;
> >     res = access_process_vm(task, mm->env_start, buffer, len, 0);
> > - if (!ptrace_may_attach(task))
> > -     res = -ESRCH;
> > +out:
> >     mmput(mm);
> > }
> > return res;
>
> What's wrong with the existing code? It's a bit dopey-looking and can, I
> guess, permit a task to cause a pagefault in an mm which it doesn't have
> permission to read from. But is there some more serious problem being
> fixed here?
```

I think not, because environment will be copied from target task, stay in kernel tmp buffer, but not copied to target buffer due to -ESRCH.
But such code is asking for problems.
