Subject: [PATCH] struct file leakage
Posted by dev on Mon, 10 Jul 2006 09:05:35 GMT

View Forum Message <> Reply to Message

Hello!

Andrew, this is a patch from Alexey Kuznetsov for 2.6.16.
I believe 2.6.17 still has this leak.

 -- -

2.6.16 leaks like hell. While testing, I found massive leakage
(reproduced in openvz) in:

*filp
*size-4096

And 1 object leaks in
*size-32
*size-64
*size-128

It is the fix for the first one. filp leaks in the bowels
of namei.c.

Seems, size-4096 is file table leaking in expand_fdtables.

I have no idea what are the rest and why they show only
accompaniing another leaks. Some debugging structs?

Signed-Off-By: Alexey Kuznetsov <kuznet@ms2.inr.ac.ru>
CC: Kirill Korotaev <dev@openvz.org>

```
--- linux-2.6.16-w/fs/namei.c 2006-07-10 11:43:11.000000000 +0400
+++ linux-2.6.16/fs/namei.c 2006-07-10 11:53:36.000000000 +0400
@@ -1774,8 +1774,15 @@ do_link:
  if (error)
   goto exit_dput;
  error = __do_follow_link(&path, nd);
- if (error)
+ if (error) {
+  /* Does someone understand code flow here? Or it is only
+   * me so stupid? Anathema to whoever designed this non-sense
+   * with "intent.open".
+   */
+  if (!IS_ERR(nd->intent.open.file))
+   release_open_intent(nd);
   return error;
+ }
  nd->flags &= ~LOOKUP_PARENT;
  if (nd->last_type == LAST_BIND)
   goto ok;
```

Subject: Re: [PATCH] struct file leakage
Posted by Andrew Morton on Mon, 10 Jul 2006 10:05:26 GMT
View Forum Message <> Reply to Message

On Mon, 10 Jul 2006 13:05:35 +0400
Kirill Korotaev <dev@sw.ru> wrote:

> Hello!
>
> Andrew, this is a patch from Alexey Kuznetsov for 2.6.16.
> I believe 2.6.17 still has this leak.
>
> -- -
>
> 2.6.16 leaks like hell. While testing, I found massive leakage
> (reproduced in openvz) in:
>
> *filp
> *size-4096
>
> And 1 object leaks in
> *size-32
> *size-64
> *size-128
>
>
> It is the fix for the first one. filp leaks in the bowels
> of namei.c.
>
> Seems, size-4096 is file table leaking in expand_fdtables.

I suspect that's been there for a long time.

> I have no idea what are the rest and why they show only
> accompaniing another leaks. Some debugging structs?

I don't understand this. Are you implying that there are other bugs.

> Signed-Off-By: Alexey Kuznetsov <kuznet@ms2.inr.ac.ru>
> CC: Kirill Korotaev <dev@openvz.org>

```
>

> --- linux-2.6.16-w/fs/namei.c 2006-07-10 11:43:11.000000000 +0400
> +++ linux-2.6.16/fs/namei.c 2006-07-10 11:53:36.000000000 +0400
> @@ -1774,8 +1774,15 @@ do_link:
>   if (error)
>    goto exit_dput;
>   error = __do_follow_link(&path, nd);
> - if (error)
> + if (error) {
> +  /* Does someone understand code flow here? Or it is only
> +   * me so stupid? Anathema to whoever designed this non-sense
> +   * with "intent.open".
> +   */
> +  if (!IS_ERR(nd->intent.open.file))
> +   release_open_intent(nd);
>    return error;
> + }
>   nd->flags &= ~LOOKUP_PARENT;
>   if (nd->last_type == LAST_BIND)
>    goto ok;
>
```

It's good to have some more Alexeycomments in the tree.

I wonder if we're also needing a path_release() here. And if not, whether it is still safe to run release_open_intent() against this nameidata?

Hopefully Trond can recall what's going on in there...

Subject: Re: [PATCH] struct file leakage

Posted by Alexey Kuznetsov on Mon, 10 Jul 2006 10:16:14 GMT

View Forum Message <> Reply to Message

Hello!

> I don't understand this. Are you implying that there are other bugs.

Yes. I still see leakage of another objects, most likely fdtables.
Probably, it is an internal bleeding of openvz or it was already fixed
in mainstreem. I still do not know.

Alexey

Subject: Re: [PATCH] struct file leakage

Posted by ebiederm on Mon, 10 Jul 2006 11:56:07 GMT
View Forum Message <> Reply to Message

Kirill Korotaev <dev@sw.ru> writes:

> Hello!
>
> Andrew, this is a patch from Alexey Kuznetsov for 2.6.16.
> I believe 2.6.17 still has this leak.
>
> -- -
>
> 2.6.16 leaks like hell. While testing, I found massive leakage
> (reproduced in openvz) in:
>
> *filp
> *size-4096
>
> And 1 object leaks in
> *size-32
> *size-64
> *size-128
>
>
> It is the fix for the first one. filp leaks in the bowels
> of namei.c.
>
> Seems, size-4096 is file table leaking in expand_fdtables.
>
> I have no idea what are the rest and why they show only
> accompaniing another leaks. Some debugging structs?

Or something the intent or the filp holds a reference to?

Looks like this has been broken since 834f2a4a1554dc5b2598038b3fe8703defcbe467
about 9 months ago.

The patch looks sane.

Trond did you just miss this case?

```
> Signed-Off-By: Alexey Kuznetsov <kuznet@ms2.inr.ac.ru>
> CC: Kirill Korotaev <dev@openvz.org>
>
> --- linux-2.6.16-w/fs/namei.c 2006-07-10 11:43:11.000000000 +0400
> +++ linux-2.6.16/fs/namei.c 2006-07-10 11:53:36.000000000 +0400
> @@ -1774,8 +1774,15 @@ do_link:
>   if (error)
>    goto exit_dput;
>   error = __do_follow_link(&path, nd);
> - if (error)
> + if (error) {
> +  /* Does someone understand code flow here? Or it is only
> +   * me so stupid? Anathema to whoever designed this non-sense
> +   * with "intent.open".
> +   */
> +  if (!IS_ERR(nd->intent.open.file))
> +   release_open_intent(nd);
>    return error;
> + }
>   nd->flags &= ~LOOKUP_PARENT;
>   if (nd->last_type == LAST_BIND)
>    goto ok;
```

Eric

Subject: Re: [PATCH] struct file leakage
Posted by Trond Myklebust on Tue, 11 Jul 2006 12:04:06 GMT

View Forum Message <> Reply to Message

On Mon, 2006-07-10 at 03:05 -0700, Andrew Morton wrote:
> On Mon, 10 Jul 2006 13:05:35 +0400
> Kirill Korotaev <dev@sw.ru> wrote:
>
> > Hello!
> >
> > Andrew, this is a patch from Alexey Kuznetsov for 2.6.16.
> > I believe 2.6.17 still has this leak.
> >
> > --- -
> >
> > 2.6.16 leaks like hell. While testing, I found massive leakage
> > (reproduced in openvz) in:
> >
> > *filp
> > *size-4096
> >
> > And 1 object leaks in
> > *size-32
> > *size-64
> > *size-128
> >
> >
> > It is the fix for the first one. filp leaks in the bowels
> > of namei.c.

> >
> > Seems, size-4096 is file table leaking in expand_fdtables.
>
> I suspect that's been there for a long time.
>
> > I have no idea what are the rest and why they show only
> > accompaniing another leaks. Some debugging structs?
>
> I don't understand this. Are you implying that there are other bugs.
>
> > Signed-Off-By: Alexey Kuznetsov <kuznet@ms2.inr.ac.ru>
> > CC: Kirill Korotaev <dev@openvz.org>
> >
>

```
> > --- linux-2.6.16-w/fs/namei.c 2006-07-10 11:43:11.000000000 +0400
> > +++ linux-2.6.16/fs/namei.c 2006-07-10 11:53:36.000000000 +0400
> > @@ -1774,8 +1774,15 @@ do_link:
> >   if (error)
> >    goto exit_dput;
> >   error = __do_follow_link(&path, nd);
> > - if (error)
> > + if (error) {
> > +  /* Does someone understand code flow here? Or it is only
> > +   * me so stupid? Anathema to whoever designed this non-sense
> > +   * with "intent.open".
> > +   */
> > +  if (!IS_ERR(nd->intent.open.file))
> > +   release_open_intent(nd);
> >    return error;
> > + }
> >   nd->flags &= ~LOOKUP_PARENT;
> >   if (nd->last_type == LAST_BIND)
> >    goto ok;
> >
```

>
> It's good to have some more Alexeycomments in the tree.
>
> I wonder if we're also needing a path_release() here. And if not, whether
> it is still safe to run release_open_intent() against this nameidata?
>
> Hopefully Trond can recall what's going on in there...

The patch looks correct, except that I believe we can skip the IS_ERR()
test there: if we're following links then we presumably have not tried
to open any files yet, so the call to release_open_intent(nd) can be
made unconditional.

Cheers,

Trond

Subject: Re: [PATCH] struct file leakage
Posted by Andrew Morton on Tue, 11 Jul 2006 23:30:08 GMT
View Forum Message <> Reply to Message

Trond Myklebust <trond.myklebust@fys.uio.no> wrote:

```
>
> > > - if (error)
> > > + if (error) {
> > > +  /* Does someone understand code flow here? Or it is only
> > > +   * me so stupid? Anathema to whoever designed this non-sense
> > > +   * with "intent.open".
> > > +   */
> > > +  if (!IS_ERR(nd->intent.open.file))
> > > +   release_open_intent(nd);
> > >    return error;
> > > + }
> > >   nd->flags &= ~LOOKUP_PARENT;
> > >   if (nd->last_type == LAST_BIND)
> > >    goto ok;
> > >
> >
> > It's good to have some more Alexeycomments in the tree.
> >
> > I wonder if we're also needing a path_release() here.  And if not, whether
> > it is still safe to run release_open_intent() against this nameidata?
> >
> > Hopefully Trond can recall what's going on in there...
>
> The patch looks correct, except that I believe we can skip the IS_ERR()
> test there: if we're following links then we presumably have not tried
> to open any files yet, so the call to release_open_intent(nd) can be
> made unconditional.
```

Sorry, but phrases like "looks correct" and "I believe" don't inspire
confidence. (Although what you say looks correct ;)) Are you sure?

And do we also need a path_release(nd) in there?

Subject: Re: [PATCH] struct file leakage
Posted by Trond Myklebust on Wed, 12 Jul 2006 00:26:02 GMT
View Forum Message <> Reply to Message

On Tue, 2006-07-11 at 16:32 -0700, Andrew Morton wrote:

> Trond Myklebust <trond.myklebust@fys.uio.no> wrote:
> >
> > > > - if (error)
> > > > + if (error) {
> > > > + /* Does someone understand code flow here? Or it is only
> > > > +   * me so stupid? Anathema to whoever designed this non-sense
> > > > +   * with "intent.open".
> > > > +   */
> > > > + if (!IS_ERR(nd->intent.open.file))
> > > > +   release_open_intent(nd);
> > > >   return error;
> > > > + }
> > > >   nd->flags &= ~LOOKUP_PARENT;
> > > >   if (nd->last_type == LAST_BIND)
> > > >   goto ok;
> > > >
> > >
> > > It's good to have some more Alexeycomments in the tree.
> > >
> > > I wonder if we're also needing a path_release() here. And if not, whether
> > > it is still safe to run release_open_intent() against this nameidata?
> > >
> > > Hopefully Trond can recall what's going on in there...
> >
> > The patch looks correct, except that I believe we can skip the IS_ERR()
> > test there: if we're following links then we presumably have not tried
> > to open any files yet, so the call to release_open_intent(nd) can be
> > made unconditional.
>
> Sorry, but phrases like "looks correct" and "I believe" don't inspire
> confidence. (Although what you say looks correct ;)) Are you sure?

We do need the call to release_open_intent(), since otherwise we will
leak a struct file. The question is whether we can optimise away the
IS_ERR() test. In my opinion, we can.

> And do we also need a path_release(nd) in there?

No. do_follow_link() should release the path for us on error. Replacing
with a 'goto exit' would therefore be a mistake.

Cheers,
  Trond