Subject: Re: [PATCH] Wake up mandatory locks waiter on chmod (v2)
Posted by Trond Myklebust on Tue, 18 Sep 2007 16:54:56 GMT
View Forum Message <> Reply to Message

On Tue, 2007-09-18 at 12:52 -0400, J. Bruce Fields wrote:
> On Tue, Sep 18, 2007 at 12:14:55PM -0400, Trond Myklebust wrote:
> > Note also that strictly speaking, we're not even compliant with the
> > System V behaviour on read() and write(). See:
> >
> >   http://www.unix.org.ua/orelly/networking_2ndEd/nfs/ch11_01.htm
> > and
> >   http://docs.sun.com/app/docs/doc/801-6736/6i13fom0a?l=en&a=view&q=mandatory+lock
> >
> > According to these docs, we should be wrapping each and every read() and
> > write() syscall with a mandatory lock. The fact that we're not, and yet
> > still not seeing any complaints just goes to show how few people are
> > actually using and relying on this...
>
> So currently there's nothing to prevent this:
>
>   - write passes locks_mandatory_area() checks
> - get mandatory lock
> - read old data
>   - write updates file data
> - read new data
>
> You can see the data change even while you hold a mandatory lock that
> should exclude writes.
>
> Similarly you might think that an application could prevent anyone from
> seeing the intermediate state of a file while it performs a series of
> writes under an exclusive mandatory lock, but actually there's nothing
> to stop a read in progress from racing with acquisition of the lock.
>
> Unless I'm missing something, that makes our mandatory lock
> implementation pretty pointless. I wish we could either fix it or just
> ditch it, but I suppose either option would be unpopular.

It gets even better when you throw mmap() into the mix :-)

Trond