
Subject: Re: [patch -mm 08/17] nsproxy: add hashtable
Posted by [Herbert Poetzl](#) on Tue, 12 Dec 2006 23:22:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, Dec 11, 2006 at 04:01:15PM -0600, Serge E. Hallyn wrote:

> Quoting Eric W. Biederman (ebiederm@xmission.com):

> > "Serge E. Hallyn" <serue@us.ibm.com> writes:

> >

> > > Quoting Eric W. Biederman (ebiederm@xmission.com):

> > >

> > > Yeah, that occurred to me, but it doesn't seem like we can possibly make

> > > sufficient guarantees to the client to make this worthwhile.

> > >

> > > I'd love to be wrong about that, but if nothing else we can't prove to

> > > the client that they're running on an unhacked host. So the host admin

> > > will always have to be trusted.

> >

> > To some extent that is true. Although all security models we have

> > currently fall down if you hack the kernel, or run your kernel

> > in a hacked virtual environment. It would be nice if under normal

> > conditions you could mount an encrypted filesystem only in a container

> > and not have concerns of those files escaping.

>

> Hmm, well perhaps I'm being overly pessimistic - IBM research did have a

> demo based on TPM of remote attestation, which may be usable for

> ensuring that you're connecting to a service on your virtual machine on

> a certain (unhacked) kernel on particular hardware, in which case what

> you're talking about may be possible - given a stringent initial

> environment (i.e. not the 'gimme \$20/month for a hosted partition in

> arizona' environment).

interesting, how would you ensure from inside
such an environment, that nobody tampered with
the kernel you are running on?

> Given that, perhaps having a virtual machine with access to encrypted

> storage - safe from the host machine admins - may not be unattainable

> after all. And given that, it would be worth designing the ns_enter()

> system call so that a parent cannot enter some child namespace.

we currently call this Context Privacy, and it
is partially implemented, but of course, it
does only work if the kernel is known good

> > Which would probably be a matter of having a separate uid_ns and not

> > allowing process outside of your container to have any permissions in

> > that filesystem.

>

> Yup. Or even just a separate uid_ns and an ecryptfs partition, so
> that the host can back up the encrypted data incrementally (per file,
> i.e. not just the whole dmccrypted loop file).

it's simple to avoid access to certain 'tagged'
devices and/or filesystems, it's hard to handle
kernel modifications or even simple things like
reading the kernel memory ...

best,
Herbert

> -serge

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
