Subject: Re: [patch -mm 08/17] nsproxy: add hashtable
Posted by ebiederm on Mon, 11 Dec 2006 20:34:40 GMT
View Forum Message <> Reply to Message

"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.

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.

>> 2) When we only partially enter a namespace it is very easy for additional
>>   properties to enter that namespace. For example we enter the pid
>>   namespace and the mount namespace, but keep our current working directory
>>   in the previous namespace. Then a process in the restricted namespace
>>   can get out by cd into /proc/<?>/cwd.
>
> Yup, entering existing namespaces should be all-or-nothing.

A truly all-or-nothing has the problem that there is no external
input into the container, and a very controlled external input
to the existing container is what this is about.

>> If someones permissions to various objects does not depend on the namespace
>> they are in quite possibly this is a non-issue. If we actually depend on
>> the isolation to keep things secure enter is a setup for a first rate escape.
>
> I don't believe the isolation can be effective between two namespaces
> where one is an ancestor of another. It can be so long as one isn't
> the ancestor of another, but then we're not allowing either to enter
> the other namespace. So it's not a problem.

Reasonable.

> The bind_ns() proposed by Cedric is stricter, only allowing nsid 0 to
> switch namespaces. So it may be overly restrictive, and does introduce
> a new global namespace, but it is safe.

I will look a little more. There are a lot patches out there that need review.   What disturbs a little is that with ptrace we have an existing mechanism that can do everything we want enter or bind_ns to be able to do.

I actually have code that will let me fork a process in a new namespace today with out needing bind_ns. What is more I don't even have to be root to use it.

I would very much prefer to see us optimizing our debugging and control interfaces so they are efficient then see us implement something completely new that is problem domain specific.

Eric