Subject: Re: [PATCH 5/7]: Determine pts_ns from a pty's inode.
Posted by serue on Wed, 26 Mar 2008 15:43:44 GMT

View Forum Message <> Reply to Message

Quoting sukadev@us.ibm.com (sukadev@us.ibm.com):
> Serge E. Hallyn [serue@us.ibm.com] wrote:
> | Quoting sukadev@us.ibm.com (sukadev@us.ibm.com):
> | > Serge E. Hallyn [serue@us.ibm.com] wrote:
> | > | > | I suppose you could just create /dev/pts/ptmx and /dev/pts/tty.
> | > | > | Recommend that in containers /dev/ptmx and /dev/tty be symlinks
> | > | > | into /dev/pts. Applications don't need to change. If
> | > | > | ptmx_open() sees that inode->i_sb is a devptsfs, it gets the
> | > | > | namespace from the sb. If not, then it was a device in /dev
> | > | > | and it gets the nmespace from current.
> | > | >
> | > | > But we would still depend on user-space remounting /dev/pts after
> | > | > the clone right ? Until they do that we would access the parent
> | > | > container's /dev/pts/ptmx ?
> | > |
> | > | Yes. Which is the right thing to do imo.
> | >
> | > Hmm, that sounds reasonable, although slightly inconsistent with pid-ns,
> | > where pid starts at 1 regardless of whether /proc is remounted.
> |
> | Very different cases. The pid is the task's pid in the new pidns.
> | The task ALSO has a different pid in the parent pidns.
> |
> | The pts only has an identity in one ptsns.
> |
> | > But even so, if user fails to establish the symlink, clones the pts ns
> | > and tries to create a pty, we would end up with different pts nses again ?
> |
> | Yes. So what?
>
> We would end up allocating a pts index from child-pts-ns (i.e index 0)
> and attempt to open /dev/pts/0 which could be an existing pty in the
> parent pts ns ?

An SELinux policy tagging child devpts entries with vps1_u:vps1_r:vps1_pts_t
and not allowing vps1_t access to host_pts_t entries would forbid it if
you wanted. But failing that, the kernel doesn't break, so I don't
it's a problem.

> | > i.e
> | > /dev/ptmx is still a char dev in root fs
> | > clone(pts_ns)
> | >   (In child, (before remount /dev/pts))
> | >   open("/dev/ptmx")

> | >   open("/dev/pts/0")
> | >
> | > Since ptmx is not in devpts, we use current_pts_ns() or child-pts-ns
> | > Since /dev/pts is not remounted in child, we get the parent pts-ns from
> | >
> | > If we can somehow detect the incorrect configuration and fail either
> | > open, we should be ok :-)
> |
> | I completely disagree with this sentiment. The kernel doesn't need
> | to detect an "incorrect configuration" if it isn't dangerous. One
> | man's "incorrect configuration" is another man's useful trick.
>
> Myabe configuration is the wrong word, but unless I am missing something
> above, spanning two pts-nses is an error condition ?

For userspace, but it doesn't crash the kernel. Userspace didn't set
things up right, so it gets the wrong thing. If I do a dup2 into fd 3
and then try to read from fd 4, I get the wrong data. Is that the
kernel's fault?

-serge

____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers