Subject: Re: [PATCH 7/8] user ns: handle file sigio
Posted by ebiederm on Wed, 24 Jan 2007 17:23:31 GMT
View Forum Message <> Reply to Message

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

> From: Serge E. Hallyn <serue@us.ibm.com>
> Subject: [PATCH 7/8] user ns: handle file sigio
>
> A process in one user namespace could set a fowner and sigio on a file in a
> shared vfsmount, ending up killing a task in another user namespace.
>
> Prevent this by adding a user namespace pointer to the fown_struct, and
> enforcing that a process causing a signal to be sent be in the same
> user namespace as the file owner.
>

> @@ -455,6 +460,9 @@ static const long band_table[NSIGPOLL] =
> static inline int sigio_perm(struct task_struct *p,
>   struct fown_struct *fown, int sig)
> {
> + if (fown->user_ns != init_task.nsproxy->user_ns &&
> +   fown->user_ns != p->nsproxy->user_ns)
> + return 0;

Why is the initial user namespace being treated specially here?
Especially when you start considering nested containers special treatment
like this is semantically a real problem, to maintain.

If we need to I can see doing something special if the process setting
fown has CAP_KILL and bypassing the security checks that way, but
hard coding rules like that when it doesn't appear we have any
experience to indicate we need the extra functionality looks
premature.

>   return (((fown->euid == 0) ||
>   (fown->euid == p->suid) || (fown->euid == p->uid) ||
>   (fown->uid == p->suid) || (fown->uid == p->uid)) &&

Eric

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