Subject: Re: [RFC][PATCH 2/7] VPIDs: pid/vpid conversions
Posted by ebiederm on Wed, 08 Feb 2006 20:29:30 GMT
View Forum Message <> Reply to Message

Kirill Korotaev <dev@sw.ru> writes:

> This is one of the major patches,
> it adds vpid-to-pid conversions by placing macros
> introduced in diff-vpid-macro patch.
>
> Note that in CONFIG_VIRTUAL_PIDS=n case these macros expand to default code.

Do you know how incomplete this patch is?
```
 drivers/char/tty_io.c |    7 +++++--
 fs/binfmt_elf.c       |   16 ++++++++--------
 fs/exec.c             |    4 ++--
 fs/fcntl.c            |    3 ++-
 fs/locks.c            |    4 ++--
 fs/proc/array.c       |   18 ++++++++++--------
 fs/proc/base.c        |    6 +++---
 include/net/scm.h     |    2 +-
 ipc/msg.c             |    6 +++---
 ipc/sem.c             |    8 ++++----
 ipc/shm.c             |    6 +++---
 kernel/capability.c   |    8 ++++++--
 kernel/exit.c         |   28 ++++++++++++++++++++++------
 kernel/fork.c         |    2 +-
 kernel/sched.c        |    2 +-
 kernel/signal.c       |   23 +++++++++++++++--------
 kernel/sys.c          |   37 ++++++++++++++++++++++++++-----------
 kernel/timer.c        |    6 +++---
 net/core/scm.c        |    2 +-
 net/unix/af_unix.c    |    8 ++++----
 20 files changed, 121 insertions(+), 75 deletions(-)
```

You missed drivers/char/drm, and in your shipping OpenVZ patch.
You missed get_xpid() on alpha.
You missed nfs.

All it seems you have found is the low hanging fruit where pids are used.
Without compile errors to help I don't know how you are ever going to find
everything, especially with the kernel constantly changing.

Honestly this approach looks like a maintenance nightmare, you didn't
even correctly handle all of the interfaces you posted in you patch.

I suspect the tagging of the VPIDS and the WARN_ON's help so you have
a chance of catching things if someone uses a code path you haven't

caught. But I don't see how you can possibly get full kernel
coverage.

Is there a plan to catch all of the in-kernel use of pids that I am
being to dense to see?

Eric

> Kirill

> --- ./kernel/capability.c.vpid_core 2006-02-02 14:15:35.152784704 +0300
> +++ ./kernel/capability.c 2006-02-02 14:33:58.808003608 +0300
> @@ -67,7 +67,7 @@ asmlinkage long sys_capget(cap_user_head
> 	spin_lock(&task_capability_lock);
> 	read_lock(&tasklist_lock);
>
> -	if (pid && pid != current->pid) {
> +	if (pid && pid != virt_pid(current)) {
> 		target = find_task_by_pid(pid);
> 		if (!target) {
> 			ret = -ESRCH;
> @@ -100,6 +100,10 @@ static inline int cap_set_pg(int pgrp, k
> 	int ret = -EPERM;
> 	int found = 0;
>
> +	pgrp = vpid_to_pid(pgrp);
> +	if (pgrp < 0)
> +		return ret;
> +
> 	do_each_task_pid(pgrp, PIDTYPE_PGID, g) {
> 		target = g;
> 		while_each_thread(g, target) {
> @@ -199,7 +203,7 @@ asmlinkage long sys_capset(cap_user_head
> 	spin_lock(&task_capability_lock);
> 	read_lock(&tasklist_lock);
>
> -	if (pid > 0 && pid != current->pid) {
> +	if (pid > 0 && pid != virt_pid(current)) {
> 		target = find_task_by_pid(pid);
> 		if (!target) {
> 			ret = -ESRCH;

You missed cap_set_all.