Subject: [PATCH v3] Allow access to /proc/$PID/fd after setuid()
Posted by adobriyan on Fri, 09 Feb 2007 15:02:57 GMT
View Forum Message <> Reply to Message

/proc/$PID/fd has r-x------ permissions, so if process does setuid(), it
will not be able to access /proc/*/fd/. This breaks fstatat() emulation
in glibc.

open("foo", O_RDONLY|O_DIRECTORY)   = 4
setuid32(65534)   = 0
stat64("/proc/self/fd/4/bar", 0xbfafb298) = -1 EACCES (Permission denied)

Comments and suggestions from Andrew Morton and Oleg Nesterov.

Signed-off-by: Alexey Dobriyan <adobriyan@openvz.org>
---

 fs/proc/base.c | 18 ++++++++++++++++++
 1 file changed, 18 insertions(+)

--- a/fs/proc/base.c
+++ b/fs/proc/base.c
@@ -1414,10 +1414,28 @@ static struct file_operations proc_fd_op
 };

 /*
+ * /proc/pid/fd needs a special permission handler so that a process can still
+ * access /proc/self/fd after it has executed a setuid().
+ */
+static int proc_fd_permission(struct inode *inode, int mask,
+   struct nameidata *nd)
+{
+ int rv;
+
+ rv = generic_permission(inode, mask, NULL);
+ if (rv == 0)
+ return 0;
+ if (task_pid(current) == proc_pid(inode))
+ rv = 0;
+ return rv;
+}
+
+/*
  * proc directories can do almost nothing..
  */
 static struct inode_operations proc_fd_inode_operations = {
  .lookup = proc_lookupfd,
+ .permission = proc_fd_permission,

```
  .setattr = proc_setattr,
};
```