
Subject: [PATCH] ve_env checks for netlink
Posted by [dim](#) on Sat, 10 Dec 2005 16:43:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Patch from Dmitry (dim@):

- this patch adds owner_env to primary key for selecting from nl_table in order to avoid collisions due to virt pids.
Bug #55602

--

Thanks,
Dmitry.

```
--- ./net/netlink/af_netlink.c.venetlink 2005-12-05 12:01:57.000000000 +0300
+++ ./net/netlink/af_netlink.c 2005-12-10 19:42:31.000000000 +0300
@@ -154,7 +154,10 @@ static __inline__ struct sock *netlink_l
```

```
    read_lock(&nl_table_lock);
    sk_for_each(sk, node, &nl_table[protocol]) {
-   if (nlk_sk(sk)->pid == pid) {
+   /* VEs should find sockets, created by kernel */
+   if ((nlk_sk(sk)->pid == pid) &&
+       (!pid || ve_accessible_strict(VE_OWNER_SK(sk),
+   get_exec_env()))){
        sock_hold(sk);
        goto found;
    }
@@ -175,7 +178,9 @@ static int netlink_insert(struct sock *s
```

```
    netlink_table_grab();
    sk_for_each(osk, node, &nl_table[sk->sk_protocol]) {
-   if (nlk_sk(osk)->pid == pid)
+   if ((nlk_sk(osk)->pid == pid) &&
+       ve_accessible_strict(VE_OWNER_SK(osk),
+   get_exec_env()))
        break;
    }
    if (!node) {
@@ -286,13 +291,15 @@ static int netlink_autobind(struct socke
    struct sock *sk = sock->sk;
    struct sock *osk;
    struct hlist_node *node;
-   s32 pid = current->pid;
+   s32 pid = virt_pid(current);
    int err;
```

```
retry:
    netlink_table_grab();
```

```

sk_for_each(osk, node, &nl_table[sk->sk_protocol]) {
- if (nlk_sk(osk)->pid == pid) {
+ if ((nlk_sk(osk)->pid == pid) &&
+ ve_accessible_strict(VE_OWNER_SK(osk),
+ get_exec_env())){
/* Bind collision, search negative pid values. */
if (pid > 0)
pid = -4096;
@@ -661,6 +668,9 @@ void netlink_set_err(struct sock *ssk, u
if (nlk->pid == pid || !(nlk->groups & group))
continue;

+ if (!ve_accessible_strict(get_exec_env(), VE_OWNER_SK(sk)))
+ continue;
+
sk->sk_err = code;
sk->sk_error_report(sk);
}
@@ -1276,6 +1286,7 @@ static int __init netlink_proto_init(voi
}
sock_register(&netlink_family_ops);
#ifdef CONFIG_PROC_FS
+ /* FIXME: virtualize before give access from VEs */
proc_net_fops_create("netlink", 0, &netlink_seq_fops);
#endif
/* The netlink device handler may be needed early. */

```

File Attachments

1) [diff-ve-netlink-20051210](#), downloaded 714 times
