Subject: Kernelspace L2TP (using openl2tpd) inside VE - PPPOX socket fail
Posted by Night Nord on Fri, 05 Sep 2008 16:40:08 GMT
View Forum Message <> Reply to Message

I previously posted this inside ppp-related thread into Russian forum, but it looks like, that it was wrong place. So now I repost (and restruct) all information into new thread.

Before all:
I have 2.6.26 openvz kernel (from git, but also tried with 2.6.24)
Linux 2.6.26 #10 Tue Sep 2 22:32:53 MSD 2008 i686 Intel(R) Celeron(R) CPU 2.80GHz
vzctl --version: 3.0.22
Kernel and vzctl were patched with patches from bugzilla -
http://bugzilla.openvz.org/show_bug.cgi?id=268
Also security check into net/socket.c was patched to pass through PF_PPPOX requests;

I'm trying to get openl2tpd, which use CONFIG_PPPOL2TP kernel module, working. While openl2tpd itself starting successfully, and even establish a tunnel - session creation failing during PPPOX socket creation with '-EAFNOSUPPORT' error:
556 socket(PF_PPPOX, SOCK_DGRAM, 1) = -1 EAFNOSUPPORT (Address family not supported by protocol)
This problem appears only in VE environment.

During some tests, error was located into drivers/net/pppox.c:

```
static int pppox_create(struct net *net, struct socket *sock, int protocol)
{
	int rc = -EPROTOTYPE;

	if (net != &init_net) /* << Error here */
		return -EAFNOSUPPORT;
```

(where init_net - boot-created structure from include/net/net_namespace.h:

```
/* Init's network namespace */
extern struct net init_net;
```

So, actually init_net - this is HN-init's network namespace and, of course, it isn't equal VE'e net)

I have tried to simple disable this check, but this fails with 'EFAULT'. I wonder if there is any workaround... I have idea about saving init_net into ve_struct objects during VE creation, and replacing all such checks to VE-compatible checks, but this is a big work and my knowledge of C and kernel insufficient to predict results (I even can't code for VE creation).
Is there any difference between HN'n net and VE's net for socket operations, or can we just jump into ve0_enviroment while pppox creating and then jump back (or this will mess up everything as creating ve-requested socket into ve0)?

P.S. Sorry for bad English