
Subject: [PATCH 4/4 (resent) net-2.6.25][UNIX] Make the unix sysctl tables per-namespace

Posted by [Pavel Emelianov](#) on Fri, 30 Nov 2007 16:37:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

I'm awfully sorry, but I noticed, that I sent the wrong patch right after I pressed the "Send" button. The first version contains a fatal error - the return code in register function should be inverted (the if (net->unix_ctl != NULL) one). Otherwise the registered table will be freed :(

Sorry. This is the correct patch.

This is the core.

- * add the ctl_table_header on the struct net;
- * make the unix_sysctl_register and _unregister clone the table;
- * moves calls to them into per-net init and exit callbacks;
- * move the .data pointer in the proper place.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
index b0cf075..f97b2a4 100644

--- a/include/net/net_namespace.h
+++ b/include/net/net_namespace.h

@@ -11,6 +11,8 @@

struct proc_dir_entry;

struct net_device;

struct sock;

+struct ctl_table_header;

+

struct net {

atomic_t count; /* To decided when the network

* namespace should be freed.

@@ -41,6 +43,7 @@ struct net {

/* unix sockets */

int sysctl_unix_max_dgram_qlen;

+ struct ctl_table_header *unix_ctl;

};

#ifdef CONFIG_NET

diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c

index 73620d6..b8a2189 100644

--- a/net/unix/af_unix.c

```

+++ b/net/unix/af_unix.c
@@ -2139,10 +2139,14 @@ static int unix_net_init(struct net *net)
    int error = -ENOMEM;

    net->sysctl_unix_max_dgram_qlen = 10;
+ if (unix_sysctl_register(net))
+ goto out;

    #ifdef CONFIG_PROC_FS
- if (!proc_net_fops_create(net, "unix", 0, &unix_seq_fops))
+ if (!proc_net_fops_create(net, "unix", 0, &unix_seq_fops)) {
+ unix_sysctl_unregister(net);
    goto out;
+ }
    #endif
    error = 0;
    out:
@@ -2151,6 +2155,7 @@ out:

    static void unix_net_exit(struct net *net)
    {
+ unix_sysctl_unregister(net);
    proc_net_remove(net, "unix");
    }

@@ -2175,7 +2180,6 @@ static int __init af_unix_init(void)

    sock_register(&unix_family_ops);
    register_pernet_subsys(&unix_net_ops);
- unix_sysctl_register(&init_net);
    out:
    return rc;
}
@@ -2183,7 +2187,6 @@ out:
    static void __exit af_unix_exit(void)
    {
    sock_unregister(PF_UNIX);
- unix_sysctl_unregister(&init_net);
    proto_unregister(&unix_proto);
    unregister_pernet_subsys(&unix_net_ops);
    }
diff --git a/net/unix/sysctl_net_unix.c b/net/unix/sysctl_net_unix.c
index 43dd356..1c57f8f 100644
--- a/net/unix/sysctl_net_unix.c
+++ b/net/unix/sysctl_net_unix.c
@@ -32,16 +32,33 @@ static struct ctl_path unix_path[] = {
    {},
};

```

```

-static struct ctl_table_header * unix_sysctl_header;
-
int unix_sysctl_register(struct net *net)
{
- unix_sysctl_header = register_sysctl_paths(unix_path, unix_table);
- return unix_sysctl_header == NULL ? -ENOMEM : 0;
+ struct ctl_table *table;
+
+ table = kmemdup(unix_table, sizeof(unix_table), GFP_KERNEL);
+ if (table == NULL)
+ goto err_alloc;
+
+ table[0].data = &net->sysctl_unix_max_dgram_qlen;
+ net->unix_ctl = register_net_sysctl_table(net, unix_path, table);
+ if (net->unix_ctl == NULL)
+ goto err_reg;
+
+ return 0;
+
+err_reg:
+ kfree(table);
+err_alloc:
+ return -ENOMEM;
}

void unix_sysctl_unregister(struct net *net)
{
- unregister_sysctl_table(unix_sysctl_header);
+ struct ctl_table *table;
+
+ table = net->unix_ctl->ctl_table_arg;
+ unregister_sysctl_table(net->unix_ctl);
+ kfree(table);
}

--
1.5.3.4

```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
