
Subject: [PATCH net-2.6.25 1/3]sysctl: make the sys.net.core sysctls per-namespace

Posted by [Pavel Emelianov](#) on Fri, 07 Dec 2007 13:07:19 GMT

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

Making them per-namespace is required for the following two reasons:

First, some ctl values have a per-namespace meaning.
Second, making them writable from the sub-namespace is an isolation hole.

So I introduce the pernet operations to create these tables. For init_net I use the existing statically declared tables, for sub-namespace they are duplicated and the write bits are removed from the mode.

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

diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h

index f97b2a4..d593611 100644

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

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

@@ -37,6 +37,9 @@ struct net {

struct sock *rtnl; /* rtnetlink socket */

+ /* core sysctls */

+ struct ctl_table_header *sysctl_core_hdr;

+

/* List of all packet sockets. */

rwlock_t packet_sklist_lock;

struct hlist_head packet_sklist;

diff --git a/net/core/sysctl_net_core.c b/net/core/sysctl_net_core.c

index e322713..57a7ead 100644

--- a/net/core/sysctl_net_core.c

+++ b/net/core/sysctl_net_core.c

@@ -151,18 +151,58 @@ static struct ctl_table net_core_table[] = {
{ .ctl_name = 0 }

};

-static __initdata struct ctl_path net_core_path[] = {

+static __net_initdata struct ctl_path net_core_path[] = {

{ .procname = "net", .ctl_name = CTL_NET, },

{ .procname = "core", .ctl_name = NET_CORE, },

{ },

```

};

-static __init int sysctl_core_init(void)
+static __net_init int sysctl_core_net_init(struct net *net)
{
- struct ctl_table_header *hdr;
+ struct ctl_table *tbl, *tmp;
+
+ tbl = net_core_table;
+ if (net != &init_net) {
+   tbl = kmemdup(tbl, sizeof(net_core_table), GFP_KERNEL);
+   if (tbl == NULL)
+     goto err_dup;
+
+   for (tmp = tbl; tmp->procname; tmp++)
+     tmp->mode &= ~0222;
+ }
+
+ net->sysctl_core_hdr = register_net_sysctl_table(net,
+   net_core_path, tbl);
+ if (net->sysctl_core_hdr == NULL)
+   goto err_reg;

- hdr = register_sysctl_paths(net_core_path, net_core_table);
- return hdr == NULL ? -ENOMEM : 0;
+ return 0;
+
+err_reg:
+ if (tbl != net_core_table)
+   kfree(tbl);
+err_dup:
+ return -ENOMEM;
+}
+
+static __net_exit void sysctl_core_net_exit(struct net *net)
+{
+ struct ctl_table *tbl;
+
+ tbl = net->sysctl_core_hdr->ctl_table_arg;
+ unregister_net_sysctl_table(net->sysctl_core_hdr);
+ BUG_ON(tbl == net_core_table);
+ kfree(tbl);
+}
+
+static __net_initdata struct pernet_operations sysctl_core_ops = {
+ .init = sysctl_core_net_init,
+ .exit = sysctl_core_net_exit,
+};

```

```
+
+static __init int sysctl_core_init(void)
+{
+ return register_pernet_subsys(&sysctl_core_ops);
+}
```

```
__initcall(sysctl_core_init);
```

```
--
```

1.5.3.4
