Subject: [PATCH net-2.6.26 4/6][SOCK][NETNS]: Register sockstat(6) files in each net.
Posted by Pavel Emelianov on Mon, 31 Mar 2008 12:17:41 GMT
View Forum Message <> Reply to Message

Currently they live in init_net only, but now almost all the info
they can provide is available per-net.

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

---
 net/ipv4/proc.c | 27 ++++++++++++++++++++-----
 net/ipv6/proc.c | 29 ++++++++++++++++++++++++++-----
 2 files changed, 46 insertions(+), 10 deletions(-)

diff --git a/net/ipv4/proc.c b/net/ipv4/proc.c
index 8156c26..24ae23b 100644
--- a/net/ipv4/proc.c
+++ b/net/ipv4/proc.c
@@ -426,25 +426,42 @@ static const struct file_operations netstat_seq_fops = {
  .release = single_release,
 };

+static __net_init int ip_proc_init_net(struct net *net)
+{
+ if (!proc_net_fops_create(net, "sockstat", S_IRUGO, &sockstat_seq_fops))
+ return -ENOMEM;
+ return 0;
+}
+
+static __net_exit void ip_proc_exit_net(struct net *net)
+{
+ proc_net_remove(net, "sockstat");
+}
+
+static __net_initdata struct pernet_operations ip_proc_ops = {
+ .init = ip_proc_init_net,
+ .exit = ip_proc_exit_net,
+};
+
 int __init ip_misc_proc_init(void)
 {
  int rc = 0;

+ if (register_pernet_subsys(&ip_proc_ops))
+ goto out_pernet;
+
  if (!proc_net_fops_create(&init_net, "netstat", S_IRUGO, &netstat_seq_fops))

```
   goto out_netstat;

  if (!proc_net_fops_create(&init_net, "snmp", S_IRUGO, &snmp_seq_fops))
   goto out_snmp;
-
- if (!proc_net_fops_create(&init_net, "sockstat", S_IRUGO, &sockstat_seq_fops))
-  goto out_sockstat;
 out:
  return rc;
-out_sockstat:
- proc_net_remove(&init_net, "snmp");
 out_snmp:
  proc_net_remove(&init_net, "netstat");
 out_netstat:
+ unregister_pernet_subsys(&ip_proc_ops);
+out_pernet:
  rc = -ENOMEM;
  goto out;
 }
diff --git a/net/ipv6/proc.c b/net/ipv6/proc.c
index 4b9d5a9..5623660 100644
--- a/net/ipv6/proc.c
+++ b/net/ipv6/proc.c
@@ -243,27 +243,45 @@ int snmp6_unregister_dev(struct inet6_dev *idev)
  return 0;
 }

+static int ipv6_proc_init_net(struct net *net)
+{
+ if (!proc_net_fops_create(net, "sockstat6", S_IRUGO,
+    &sockstat6_seq_fops))
+  return -ENOMEM;
+ return 0;
+}
+
+static void ipv6_proc_exit_net(struct net *net)
+{
+ proc_net_remove(net, "sockstat6");
+}
+
+static struct pernet_operations ipv6_proc_ops = {
+ .init = ipv6_proc_init_net,
+ .exit = ipv6_proc_exit_net,
+};
+
 int __init ipv6_misc_proc_init(void)
 {
  int rc = 0;
```

```
+ if (register_pernet_subsys(&ipv6_proc_ops))
+  goto proc_net_fail;
+
  if (!proc_net_fops_create(&init_net, "snmp6", S_IRUGO, &snmp6_seq_fops))
   goto proc_snmp6_fail;

  proc_net_devsnmp6 = proc_mkdir("dev_snmp6", init_net.proc_net);
  if (!proc_net_devsnmp6)
   goto proc_dev_snmp6_fail;
-
- if (!proc_net_fops_create(&init_net, "sockstat6", S_IRUGO, &sockstat6_seq_fops))
-  goto proc_sockstat6_fail;
 out:
  return rc;

-proc_sockstat6_fail:
- proc_net_remove(&init_net, "dev_snmp6");
 proc_dev_snmp6_fail:
  proc_net_remove(&init_net, "snmp6");
 proc_snmp6_fail:
+ unregister_pernet_subsys(&ipv6_proc_ops);
+proc_net_fail:
  rc = -ENOMEM;
  goto out;
 }
@@ -273,5 +291,6 @@ void ipv6_misc_proc_exit(void)
  proc_net_remove(&init_net, "sockstat6");
  proc_net_remove(&init_net, "dev_snmp6");
  proc_net_remove(&init_net, "snmp6");
+ unregister_pernet_subsys(&ipv6_proc_ops);
 }

--
1.5.3.4
```