Subject: [PATCH net-2.6.26 3/4][NETNS][ICMP]: Make ctl tables for ICMP sysctls per-net.
Posted by Pavel Emelianov on Wed, 26 Mar 2008 08:55:52 GMT
View Forum Message <> Reply to Message

Add some flesh to ipv4_sysctl_init_net and ipv4_sysctl_exit_net,
i.e. copy the table, alter .data pointers and register it per-net.

Other ipv4_table's sysctls are now global, but this is going to
change once sysctl permissions patches migrate from -mm tree to
mainline in 2.6.26 merge window :)

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

---
 include/net/netns/ipv4.h | 1 +
 net/ipv4/sysctl_net_ipv4.c | 138 ++++++++++++++++++++++++++++++++----------
 2 files changed, 91 insertions(+), 48 deletions(-)

diff --git a/include/net/netns/ipv4.h b/include/net/netns/ipv4.h
index e3de0ff..af685f7 100644
--- a/include/net/netns/ipv4.h
+++ b/include/net/netns/ipv4.h
@@ -17,6 +17,7 @@ struct netns_ipv4 {
 #ifdef CONFIG_SYSCTL
  struct ctl_table_header *forw_hdr;
  struct ctl_table_header *frags_hdr;
+ struct ctl_table_header *ipv4_hdr;
 #endif
  struct ipv4_devconf *devconf_all;
  struct ipv4_devconf *devconf_dflt;
diff --git a/net/ipv4/sysctl_net_ipv4.c b/net/ipv4/sysctl_net_ipv4.c
index e9585c0..c437f80 100644
--- a/net/ipv4/sysctl_net_ipv4.c
+++ b/net/ipv4/sysctl_net_ipv4.c
@@ -404,38 +404,6 @@ static struct ctl_table ipv4_table[] = {
   .strategy = &ipv4_sysctl_local_port_range,
  },
  {
- .ctl_name = NET_IPV4_ICMP_ECHO_IGNORE_ALL,
- .procname = "icmp_echo_ignore_all",
- .data = &init_net.ipv4.sysctl_icmp_echo_ignore_all,
- .maxlen = sizeof(int),
- .mode = 0644,
- .proc_handler = &proc_dointvec
- },
- {
- .ctl_name = NET_IPV4_ICMP_ECHO_IGNORE_BROADCASTS,

```
-  .procname = "icmp_echo_ignore_broadcasts",
-  .data  = &init_net.ipv4.sysctl_icmp_echo_ignore_broadcasts,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
- {
-  .ctl_name = NET_IPV4_ICMP_IGNORE_BOGUS_ERROR_RESPONSES,
-  .procname = "icmp_ignore_bogus_error_responses",
-  .data  = &init_net.ipv4.sysctl_icmp_ignore_bogus_error_responses,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
- {
-  .ctl_name = NET_IPV4_ICMP_ERRORS_USE_INBOUND_IFADDR,
-  .procname = "icmp_errors_use_inbound_ifaddr",
-  .data  = &init_net.ipv4.sysctl_icmp_errors_use_inbound_ifaddr,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
- {
   .ctl_name = NET_IPV4_ROUTE,
   .procname = "route",
   .maxlen  = 0,
@@ -586,22 +554,6 @@ static struct ctl_table ipv4_table[] = {
   .proc_handler = &proc_dointvec
  },
  {
-  .ctl_name = NET_IPV4_ICMP_RATELIMIT,
-  .procname = "icmp_ratelimit",
-  .data  = &init_net.ipv4.sysctl_icmp_ratelimit,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
- {
-  .ctl_name = NET_IPV4_ICMP_RATEMASK,
-  .procname = "icmp_ratemask",
-  .data  = &init_net.ipv4.sysctl_icmp_ratemask,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
- {
   .ctl_name = NET_TCP_TW_REUSE,
   .procname = "tcp_tw_reuse",
```

```
   .data  = &sysctl_tcp_tw_reuse,
@@ -804,6 +756,58 @@ static struct ctl_table ipv4_table[] = {
  { .ctl_name = 0 }
 };

+static struct ctl_table ipv4_net_table[] = {
+ {
+  .ctl_name = NET_IPV4_ICMP_ECHO_IGNORE_ALL,
+  .procname = "icmp_echo_ignore_all",
+  .data  = &init_net.ipv4.sysctl_icmp_echo_ignore_all,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ {
+  .ctl_name = NET_IPV4_ICMP_ECHO_IGNORE_BROADCASTS,
+  .procname = "icmp_echo_ignore_broadcasts",
+  .data  = &init_net.ipv4.sysctl_icmp_echo_ignore_broadcasts,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ {
+  .ctl_name = NET_IPV4_ICMP_IGNORE_BOGUS_ERROR_RESPONSES,
+  .procname = "icmp_ignore_bogus_error_responses",
+  .data  = &init_net.ipv4.sysctl_icmp_ignore_bogus_error_responses,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ {
+  .ctl_name = NET_IPV4_ICMP_ERRORS_USE_INBOUND_IFADDR,
+  .procname = "icmp_errors_use_inbound_ifaddr",
+  .data  = &init_net.ipv4.sysctl_icmp_errors_use_inbound_ifaddr,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ {
+  .ctl_name = NET_IPV4_ICMP_RATELIMIT,
+  .procname = "icmp_ratelimit",
+  .data  = &init_net.ipv4.sysctl_icmp_ratelimit,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ {
+  .ctl_name = NET_IPV4_ICMP_RATEMASK,
```

```
+  .procname = "icmp_ratemask",
+  .data  = &init_net.ipv4.sysctl_icmp_ratemask,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ { }
+};
+
 struct ctl_path net_ipv4_ctl_path[] = {
  { .procname = "net", .ctl_name = CTL_NET, },
  { .procname = "ipv4", .ctl_name = NET_IPV4, },
@@ -813,11 +817,49 @@ EXPORT_SYMBOL_GPL(net_ipv4_ctl_path);

 static __net_init int ipv4_sysctl_init_net(struct net *net)
 {
+ struct ctl_table *table;
+
+ table = ipv4_net_table;
+ if (net != &init_net) {
+  table = kmemdup(table, sizeof(ipv4_net_table), GFP_KERNEL);
+  if (table == NULL)
+   goto err_alloc;
+
+  table[0].data =
+   &net->ipv4.sysctl_icmp_echo_ignore_all;
+  table[1].data =
+   &net->ipv4.sysctl_icmp_echo_ignore_broadcasts;
+  table[2].data =
+   &net->ipv4.sysctl_icmp_ignore_bogus_error_responses;
+  table[3].data =
+   &net->ipv4.sysctl_icmp_errors_use_inbound_ifaddr;
+  table[4].data =
+   &net->ipv4.sysctl_icmp_ratelimit;
+  table[5].data =
+   &net->ipv4.sysctl_icmp_ratemask;
+ }
+
+ net->ipv4.ipv4_hdr = register_net_sysctl_table(net,
+   net_ipv4_ctl_path, table);
+ if (net->ipv4.ipv4_hdr == NULL)
+  goto err_reg;
+
  return 0;
+
+err_reg:
+ if (net != &init_net)
+  kfree(table);
```

```
+err_alloc:
+ return -ENOMEM;
 }

 static __net_exit void ipv4_sysctl_exit_net(struct net *net)
 {
+ struct ctl_table *table;
+
+ table = net->ipv4.ipv4_hdr->ctl_table_arg;
+ unregister_net_sysctl_table(net->ipv4.ipv4_hdr);
+ kfree(table);
 }

 static __net_initdata struct pernet_operations ipv4_sysctl_ops = {
--
1.5.3.4
```