Subject: [PATCH net-2.6.26 2/4][NETNS][ICMP]: Move ICMP sysctls on struct net.
Posted by Pavel Emelianov on Wed, 26 Mar 2008 08:53:03 GMT
View Forum Message <> Reply to Message

Initialization is moved to icmp_sk_init, all the places, that
refer to them use init_net for now.

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

---
 include/net/icmp.h         |    7 -----
 include/net/netns/ipv4.h   |    7 +++++
 net/ipv4/icmp.c            |   60 ++++++++++++++++++++++++++----------------
 net/ipv4/sysctl_net_ipv4.c |   12 ++++++----
 4 files changed, 44 insertions(+), 42 deletions(-)

diff --git a/include/net/icmp.h b/include/net/icmp.h
index faba64d..dddb839 100644
--- a/include/net/icmp.h
+++ b/include/net/icmp.h
@@ -65,11 +65,4 @@ static inline struct raw_sock *raw_sk(const struct sock *sk)
  return (struct raw_sock *)sk;
 }

-extern int sysctl_icmp_echo_ignore_all;
-extern int sysctl_icmp_echo_ignore_broadcasts;
-extern int sysctl_icmp_ignore_bogus_error_responses;
-extern int sysctl_icmp_errors_use_inbound_ifaddr;
-extern int sysctl_icmp_ratelimit;
-extern int sysctl_icmp_ratemask;
-
 #endif /* _ICMP_H */
diff --git a/include/net/netns/ipv4.h b/include/net/netns/ipv4.h
index 504fde1..e3de0ff 100644
--- a/include/net/netns/ipv4.h
+++ b/include/net/netns/ipv4.h
@@ -35,5 +35,12 @@ struct netns_ipv4 {
  struct xt_table  *iptable_raw;
  struct xt_table  *arptable_filter;
 #endif
+
+ int sysctl_icmp_echo_ignore_all;
+ int sysctl_icmp_echo_ignore_broadcasts;
+ int sysctl_icmp_ignore_bogus_error_responses;
+ int sysctl_icmp_ratelimit;
+ int sysctl_icmp_ratemask;
+ int sysctl_icmp_errors_use_inbound_ifaddr;
 };

```
 #endif
diff --git a/net/ipv4/icmp.c b/net/ipv4/icmp.c
index f38f093..958a384 100644
--- a/net/ipv4/icmp.c
+++ b/net/ipv4/icmp.c
@@ -188,29 +188,6 @@ struct icmp_err icmp_err_convert[] = {
  },
 };

-/* Control parameters for ECHO replies. */
-int sysctl_icmp_echo_ignore_all __read_mostly;
-int sysctl_icmp_echo_ignore_broadcasts __read_mostly = 1;
-
-/* Control parameter - ignore bogus broadcast responses? */
-int sysctl_icmp_ignore_bogus_error_responses __read_mostly = 1;
-
-/*
- *  Configurable global rate limit.
- *
- * ratelimit defines tokens/packet consumed for dst->rate_token bucket
- * ratemask defines which icmp types are ratelimited by setting
- *  it's bit position.
- *
- * default:
- * dest unreachable (3), source quench (4),
- * time exceeded (11), parameter problem (12)
- */
-
-int sysctl_icmp_ratelimit __read_mostly = 1 * HZ;
-int sysctl_icmp_ratemask __read_mostly = 0x1818;
-int sysctl_icmp_errors_use_inbound_ifaddr __read_mostly;
-
 /*
  * ICMP control array. This specifies what to do with each ICMP.
  */
@@ -310,8 +287,8 @@ static inline int icmpv4_xrlim_allow(struct rtable *rt, int type, int code)
   goto out;

  /* Limit if icmp type is enabled in ratemask. */
- if ((1 << type) & sysctl_icmp_ratemask)
-  rc = xrlim_allow(dst, sysctl_icmp_ratelimit);
+ if ((1 << type) & init_net.ipv4.sysctl_icmp_ratemask)
+  rc = xrlim_allow(dst, init_net.ipv4.sysctl_icmp_ratelimit);
 out:
  return rc;
 }
@@ -523,7 +500,8 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)
  if (!(rt->rt_flags & RTCF_LOCAL)) {
```

```
   struct net_device *dev = NULL;

-  if (rt->fl.iif && sysctl_icmp_errors_use_inbound_ifaddr)
+  if (rt->fl.iif &&
+    init_net.ipv4.sysctl_icmp_errors_use_inbound_ifaddr)
     dev = dev_get_by_index(net, rt->fl.iif);

   if (dev) {
@@ -745,7 +723,7 @@ static void icmp_unreach(struct sk_buff *skb)
    * get the other vendor to fix their kit.
    */

- if (!sysctl_icmp_ignore_bogus_error_responses &&
+ if (!init_net.ipv4.sysctl_icmp_ignore_bogus_error_responses &&
       inet_addr_type(net, iph->daddr) == RTN_BROADCAST) {
   if (net_ratelimit())
     printk(KERN_WARNING "%u.%u.%u.%u sent an invalid ICMP "
@@ -840,7 +818,7 @@ out_err:

 static void icmp_echo(struct sk_buff *skb)
 {
- if (!sysctl_icmp_echo_ignore_all) {
+ if (!init_net.ipv4.sysctl_icmp_echo_ignore_all) {
   struct icmp_bxm icmp_param;

   icmp_param.data.icmph    = *icmp_hdr(skb);
@@ -1051,7 +1029,7 @@ int icmp_rcv(struct sk_buff *skb)
     */
   if ((icmph->type == ICMP_ECHO ||
        icmph->type == ICMP_TIMESTAMP) &&
-      sysctl_icmp_echo_ignore_broadcasts) {
+      init_net.ipv4.sysctl_icmp_echo_ignore_broadcasts) {
     goto error;
   }
   if (icmph->type != ICMP_ECHO &&
@@ -1195,6 +1173,30 @@ int __net_init icmp_sk_init(struct net *net)
     */
   sk->sk_prot->unhash(sk);
 }
+
+ /* Control parameters for ECHO replies. */
+ net->ipv4.sysctl_icmp_echo_ignore_all = 0;
+ net->ipv4.sysctl_icmp_echo_ignore_broadcasts = 1;
+
+ /* Control parameter - ignore bogus broadcast responses? */
+ net->ipv4.sysctl_icmp_ignore_bogus_error_responses = 1;
+
+ /*
```

```
+  *  Configurable global rate limit.
+  *
+  * ratelimit defines tokens/packet consumed for dst->rate_token
+  * bucket ratemask defines which icmp types are ratelimited by
+  * setting it's bit position.
+  *
+  * default:
+  * dest unreachable (3), source quench (4),
+  * time exceeded (11), parameter problem (12)
+  */
+
+ net->ipv4.sysctl_icmp_ratelimit = 1 * HZ;
+ net->ipv4.sysctl_icmp_ratemask = 0x1818;
+ net->ipv4.sysctl_icmp_errors_use_inbound_ifaddr = 0;
+
   return 0;

  fail:
diff --git a/net/ipv4/sysctl_net_ipv4.c b/net/ipv4/sysctl_net_ipv4.c
index c2fca30..e9585c0 100644
--- a/net/ipv4/sysctl_net_ipv4.c
+++ b/net/ipv4/sysctl_net_ipv4.c
@@ -406,7 +406,7 @@ static struct ctl_table ipv4_table[] = {
   {
    .ctl_name = NET_IPV4_ICMP_ECHO_IGNORE_ALL,
    .procname = "icmp_echo_ignore_all",
-   .data  = &sysctl_icmp_echo_ignore_all,
+   .data  = &init_net.ipv4.sysctl_icmp_echo_ignore_all,
    .maxlen  = sizeof(int),
    .mode  = 0644,
    .proc_handler = &proc_dointvec
@@ -414,7 +414,7 @@ static struct ctl_table ipv4_table[] = {
   {
    .ctl_name = NET_IPV4_ICMP_ECHO_IGNORE_BROADCASTS,
    .procname = "icmp_echo_ignore_broadcasts",
-   .data  = &sysctl_icmp_echo_ignore_broadcasts,
+   .data  = &init_net.ipv4.sysctl_icmp_echo_ignore_broadcasts,
    .maxlen  = sizeof(int),
    .mode  = 0644,
    .proc_handler = &proc_dointvec
@@ -422,7 +422,7 @@ static struct ctl_table ipv4_table[] = {
   {
    .ctl_name = NET_IPV4_ICMP_IGNORE_BOGUS_ERROR_RESPONSES,
    .procname = "icmp_ignore_bogus_error_responses",
-   .data  = &sysctl_icmp_ignore_bogus_error_responses,
+   .data  = &init_net.ipv4.sysctl_icmp_ignore_bogus_error_responses,
    .maxlen  = sizeof(int),
    .mode  = 0644,
```

```
   .proc_handler = &proc_dointvec
@@ -430,7 +430,7 @@ static struct ctl_table ipv4_table[] = {
  {
   .ctl_name = NET_IPV4_ICMP_ERRORS_USE_INBOUND_IFADDR,
   .procname = "icmp_errors_use_inbound_ifaddr",
-  .data  = &sysctl_icmp_errors_use_inbound_ifaddr,
+  .data  = &init_net.ipv4.sysctl_icmp_errors_use_inbound_ifaddr,
   .maxlen  = sizeof(int),
   .mode  = 0644,
   .proc_handler = &proc_dointvec
@@ -588,7 +588,7 @@ static struct ctl_table ipv4_table[] = {
  {
   .ctl_name = NET_IPV4_ICMP_RATELIMIT,
   .procname = "icmp_ratelimit",
-  .data  = &sysctl_icmp_ratelimit,
+  .data  = &init_net.ipv4.sysctl_icmp_ratelimit,
   .maxlen  = sizeof(int),
   .mode  = 0644,
   .proc_handler = &proc_dointvec
@@ -596,7 +596,7 @@ static struct ctl_table ipv4_table[] = {
  {
   .ctl_name = NET_IPV4_ICMP_RATEMASK,
   .procname = "icmp_ratemask",
-  .data  = &sysctl_icmp_ratemask,
+  .data  = &init_net.ipv4.sysctl_icmp_ratemask,
   .maxlen  = sizeof(int),
   .mode  = 0644,
   .proc_handler = &proc_dointvec
--
1.5.3.4
```

---

Subject: Re: [PATCH net-2.6.26 2/4][NETNS][ICMP]: Move ICMP sysctls on struct net.
Posted by davem on Wed, 26 Mar 2008 08:55:48 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Wed, 26 Mar 2008 11:53:03 +0300

> Initialization is moved to icmp_sk_init, all the places, that
> refer to them use init_net for now.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

Applied.

---