
Subject: [PATCH net-2.6.26 4/4][NETNS][ICMP]: Use per-net sysctls in ipv4/icmp.c.
Posted by [Pavel Emelianov](#) on Wed, 26 Mar 2008 08:57:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

This mostly re-uses the net, used in icmp netnsization patches from Denis.

After this ICMP sysctls are completely virtualized.

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

net/ipv4/icmp.c | 25 ++++++++-----
1 files changed, 16 insertions(+), 9 deletions(-)

diff --git a/net/ipv4/icmp.c b/net/ipv4/icmp.c
index 958a384..7049b3f 100644

--- a/net/ipv4/icmp.c

+++ b/net/ipv4/icmp.c

@@ -270,7 +270,8 @@ int xrlim_allow(struct dst_entry *dst, int timeout)
 return rc;
}

-static inline int icmpv4_xrlim_allow(struct rtable *rt, int type, int code)

+static inline int icmpv4_xrlim_allow(struct net *net, struct rtable *rt,

+ int type, int code)

{
 struct dst_entry *dst = &rt->u.dst;
 int rc = 1;

@@ -287,8 +288,8 @@ 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) & init_net.ipv4.sysctl_icmp_ratemask)

- rc = xrlim_allow(dst, init_net.ipv4.sysctl_icmp_ratelimit);

+ if ((1 << type) & net->ipv4.sysctl_icmp_ratemask)

+ rc = xrlim_allow(dst, net->ipv4.sysctl_icmp_ratelimit);

out:
 return rc;

}
@@ -390,7 +391,7 @@ static void icmp_reply(struct icmp_bxm *icmp_param, struct sk_buff
*skb)

if (ip_route_output_key(net, &rt, &fl))
 goto out_unlock;

}
- if (icmpv4_xrlim_allow(rt, icmp_param->data.icmph.type,
+ if (icmpv4_xrlim_allow(net, rt, icmp_param->data.icmph.type,
 icmp_param->data.icmph.code))
 icmp_push_reply(icmp_param, &ipc, rt);

```

ip_rt_put(rt);
@@ -501,7 +502,7 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)
    struct net_device *dev = NULL;

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

    if (dev) {
@@ -617,7 +618,7 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)
    }

route_done:
- if (!icmpv4_xrlim_allow(rt, type, code))
+ if (!icmpv4_xrlim_allow(net, rt, type, code))
    goto ende;

/* RFC says return as much as we can without exceeding 576 bytes. */
@@ -723,7 +724,7 @@ static void icmp_unreach(struct sk_buff *skb)
    * get the other vendor to fix their kit.
    */

- if (!init_net.ipv4.sysctl_icmp_ignore_bogus_error_responses &&
+ if (!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 "
@@ -818,7 +819,10 @@ out_err:

static void icmp_echo(struct sk_buff *skb)
{
- if (!init_net.ipv4.sysctl_icmp_echo_ignore_all) {
+ struct net *net;
+
+ net = skb->dst->dev->nd_net;
+ if (!net->ipv4.sysctl_icmp_echo_ignore_all) {
    struct icmp_bxm icmp_param;

    icmp_param.data.icmph = *icmp_hdr(skb);
@@ -1021,6 +1025,9 @@ int icmp_rcv(struct sk_buff *skb)
    */

    if (rt->rt_flags & (RTCF_BROADCAST | RTCF_MULTICAST)) {
+ struct net *net;
+
+ net = rt->u.dst.dev->nd_net;
    /*

```

```

* RFC 1122: 3.2.2.6 An ICMP_ECHO to broadcast MAY be
*   silently ignored (we let user decide with a sysctl).
@@ -1029,7 +1036,7 @@ int icmp_rcv(struct sk_buff *skb)
*/
if ((icmph->type == ICMP_ECHO ||
    icmph->type == ICMP_TIMESTAMP) &&
-   init_net.ipv4.sysctl_icmp_echo_ignore_broadcasts) {
+   net->ipv4.sysctl_icmp_echo_ignore_broadcasts) {
    goto error;
}
if (icmph->type != ICMP_ECHO &&
--
1.5.3.4

```
