Subject: [PATCH 2/2 2.6.25] netns: separate af_packet netns data
Posted by den on Tue, 11 Dec 2007 11:54:45 GMT
View Forum Message <> Reply to Message

netns: move af_packet data to the separate header

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 include/net/net_namespace.h | 6 ++----
 include/net/netns/packet.h | 15 +++++++++++++++
 net/packet/af_packet.c | 28 ++++++++++++++++++----------
 3 files changed, 31 insertions(+), 18 deletions(-)
 create mode 100644 include/net/netns/packet.h

diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
index d943fd4..18da0af 100644
--- a/include/net/net_namespace.h
+++ b/include/net/net_namespace.h
@@ -9,6 +9,7 @@
 #include <linux/list.h>

 #include <net/netns/unix.h>
+#include <net/netns/packet.h>

 struct proc_dir_entry;
 struct net_device;
@@ -43,10 +44,7 @@ struct net {
  struct ctl_table_header *sysctl_core_hdr;
  int   sysctl_somaxconn;

- /* List of all packet sockets. */
- rwlock_t packet_sklist_lock;
- struct hlist_head packet_sklist;
-
+ struct netns_packet packet;
  struct netns_unix unx;
 };

diff --git a/include/net/netns/packet.h b/include/net/netns/packet.h
new file mode 100644
index 0000000..637daf6
--- /dev/null
+++ b/include/net/netns/packet.h
@@ -0,0 +1,15 @@
+/*
+ * Packet network namespace
+ */
+#ifndef __NETNS_PACKET_H__

```
+#define __NETNS_PACKET_H__
+
+#include <linux/list.h>
+#include <linux/spinlock.h>
+
+struct netns_packet {
+ rwlock_t  sklist_lock;
+ struct hlist_head sklist;
+};
+
+#endif /* __NETNS_PACKET_H__ */
diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c
index ace29f1..485af56 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -803,9 +803,9 @@ static int packet_release(struct socket *sock)
  net = sk->sk_net;
  po = pkt_sk(sk);

- write_lock_bh(&net->packet_sklist_lock);
+ write_lock_bh(&net->packet.sklist_lock);
  sk_del_node_init(sk);
- write_unlock_bh(&net->packet_sklist_lock);
+ write_unlock_bh(&net->packet.sklist_lock);

  /*
   * Unhook packet receive handler.
@@ -1015,9 +1015,9 @@ static int packet_create(struct net *net, struct socket *sock, int
protocol)
   po->running = 1;
  }

- write_lock_bh(&net->packet_sklist_lock);
- sk_add_node(sk, &net->packet_sklist);
- write_unlock_bh(&net->packet_sklist_lock);
+ write_lock_bh(&net->packet.sklist_lock);
+ sk_add_node(sk, &net->packet.sklist);
+ write_unlock_bh(&net->packet.sklist_lock);
  return(0);
 out:
  return err;
@@ -1452,8 +1452,8 @@ static int packet_notifier(struct notifier_block *this, unsigned long msg,
void
  struct net_device *dev = data;
  struct net *net = dev->nd_net;

- read_lock(&net->packet_sklist_lock);
- sk_for_each(sk, node, &net->packet_sklist) {
```

```
+ read_lock(&net->packet.sklist_lock);
+ sk_for_each(sk, node, &net->packet.sklist) {
   struct packet_sock *po = pkt_sk(sk);

   switch (msg) {
@@ -1492,7 +1492,7 @@ static int packet_notifier(struct notifier_block *this, unsigned long msg,
void
    break;
   }
  }
- read_unlock(&net->packet_sklist_lock);
+ read_unlock(&net->packet.sklist_lock);
  return NOTIFY_DONE;
 }

@@ -1862,7 +1862,7 @@ static inline struct sock *packet_seq_idx(struct net *net, loff_t off)
  struct sock *s;
  struct hlist_node *node;

- sk_for_each(s, node, &net->packet_sklist) {
+ sk_for_each(s, node, &net->packet.sklist) {
   if (!off--)
    return s;
  }
@@ -1872,7 +1872,7 @@ static inline struct sock *packet_seq_idx(struct net *net, loff_t off)
 static void *packet_seq_start(struct seq_file *seq, loff_t *pos)
 {
  struct net *net = seq_file_net(seq);
- read_lock(&net->packet_sklist_lock);
+ read_lock(&net->packet.sklist_lock);
  return *pos ? packet_seq_idx(net, *pos - 1) : SEQ_START_TOKEN;
 }

@@ -1881,14 +1881,14 @@ static void *packet_seq_next(struct seq_file *seq, void *v, loff_t
*pos)
  struct net *net = seq->private;
  ++*pos;
  return  (v == SEQ_START_TOKEN)
-  ? sk_head(&net->packet_sklist)
+  ? sk_head(&net->packet.sklist)
   : sk_next((struct sock*)v) ;
 }

 static void packet_seq_stop(struct seq_file *seq, void *v)
 {
  struct net *net = seq->private;
- read_unlock(&net->packet_sklist_lock);
+ read_unlock(&net->packet.sklist_lock);
```

```
 }

 static int packet_seq_show(struct seq_file *seq, void *v)
@@ -1940,8 +1940,8 @@ static const struct file_operations packet_seq_fops = {

 static int packet_net_init(struct net *net)
 {
- rwlock_init(&net->packet_sklist_lock);
- INIT_HLIST_HEAD(&net->packet_sklist);
+ rwlock_init(&net->packet.sklist_lock);
+ INIT_HLIST_HEAD(&net->packet.sklist);

  if (!proc_net_fops_create(net, "packet", 0, &packet_seq_fops))
   return -ENOMEM;
--
1.5.3.rc5
```

Subject: Re: [PATCH 2/2 2.6.25] netns: separate af_packet netns data
Posted by davem on Tue, 11 Dec 2007 12:20:10 GMT
View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@openvz.org>
Date: Tue, 11 Dec 2007 14:55:52 +0300

> netns: move af_packet data to the separate header
>
> Signed-off-by: Denis V. Lunev <den@openvz.org>

Also applied, thanks a lot.
