Subject: [patch 3/3][NETNS][IPV6] flowlabels - make proc per namespace
Posted by Daniel Lezcano on Wed, 26 Mar 2008 12:28:35 GMT
View Forum Message <> Reply to Message

Make /proc/net/ip6_flowlabel show only flow labels belonging to the
current network namespace.

Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>
Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
---
 net/ipv6/ip6_flowlabel.c | 39 ++++++++++++++++++++++++++----------
 1 file changed, 23 insertions(+), 16 deletions(-)

Index: net-2.6.26/net/ipv6/ip6_flowlabel.c
===
--- net-2.6.26.orig/net/ipv6/ip6_flowlabel.c
+++ net-2.6.26/net/ipv6/ip6_flowlabel.c
@@ -611,6 +611,7 @@ done:
 #ifdef CONFIG_PROC_FS

 struct ip6fl_iter_state {
+ struct seq_net_private p;
  int bucket;
 };

@@ -620,12 +621,15 @@ static struct ip6_flowlabel *ip6fl_get_f
 {
  struct ip6_flowlabel *fl = NULL;
  struct ip6fl_iter_state *state = ip6fl_seq_private(seq);
+ struct net *net = seq_file_net(seq);

  for (state->bucket = 0; state->bucket <= FL_HASH_MASK; ++state->bucket) {
- if (fl_ht[state->bucket]) {
-   fl = fl_ht[state->bucket];
+ fl = fl_ht[state->bucket];
+
+ while (fl && fl->fl_net != net)
+   fl = fl->next;
+ if (fl)
    break;
- }
  }
  return fl;
 }
@@ -633,12 +637,18 @@ static struct ip6_flowlabel *ip6fl_get_f
 static struct ip6_flowlabel *ip6fl_get_next(struct seq_file *seq, struct ip6_flowlabel *fl)
 {
  struct ip6fl_iter_state *state = ip6fl_seq_private(seq);

```
+ struct net *net = seq_file_net(seq);

  fl = fl->next;
+try_again:
+ while (fl && fl->fl_net != net)
+  fl = fl->next;
+
  while (!fl) {
-  if (++state->bucket <= FL_HASH_MASK)
+  if (++state->bucket <= FL_HASH_MASK) {
     fl = fl_ht[state->bucket];
-  else
+   goto try_again;
+  } else
     break;
  }
  return fl;
@@ -708,8 +718,8 @@ static const struct seq_operations ip6fl

 static int ip6fl_seq_open(struct inode *inode, struct file *file)
 {
- return seq_open_private(file, &ip6fl_seq_ops,
-   sizeof(struct ip6fl_iter_state));
+ return seq_open_net(inode, file, &ip6fl_seq_ops,
+       sizeof(struct ip6fl_iter_state));
 }

 static const struct file_operations ip6fl_seq_fops = {
@@ -717,12 +727,13 @@ static const struct file_operations ip6f
   .open  = ip6fl_seq_open,
   .read  = seq_read,
   .llseek  = seq_lseek,
- .release = seq_release_private,
+ .release = seq_release_net,
 };

 static int ip6_flowlabel_proc_init(struct net *net)
 {
- if (!proc_net_fops_create(net, "ip6_flowlabel", S_IRUGO, &ip6fl_seq_fops))
+ if (!proc_net_fops_create(net, "ip6_flowlabel",
+       S_IRUGO, &ip6fl_seq_fops))
    return -ENOMEM;
  return 0;
 }
@@ -745,25 +756,21 @@ static inline void ip6_flowlabel_proc_fi
 static inline void ip6_flowlabel_net_exit(struct net *net)
 {
  ip6_fl_purge(net);
```

```
+ ip6_flowlabel_proc_fini(net);
 }

 static struct pernet_operations ip6_flowlabel_net_ops = {
+ .init = ip6_flowlabel_proc_init,
  .exit = ip6_flowlabel_net_exit,
 };

 int ip6_flowlabel_init(void)
 {
- int err;
-
- err = register_pernet_subsys(&ip6_flowlabel_net_ops);
- if (err)
-  return err;
- return ip6_flowlabel_proc_init(&init_net);
+ return register_pernet_subsys(&ip6_flowlabel_net_ops);
 }

 void ip6_flowlabel_cleanup(void)
 {
  del_timer(&ip6_fl_gc_timer);
  unregister_pernet_subsys(&ip6_flowlabel_net_ops);
- ip6_flowlabel_proc_fini(&init_net);
 }

--
```