Subject: [PATCH 2/3] netns netfilter: netns propagation for /proc/net/*_tables_names
Posted by Alexey Dobriyan on Fri, 25 Jan 2008 16:44:29 GMT
View Forum Message <> Reply to Message

Propagate netns together with AF down to ->start/->next/->stop
iterators. Choose table based on netns and AF for showing.

Signed-off-by: Alexey Dobriyan <adobriyan@sw.ru>
---

 net/netfilter/x_tables.c | 31 ++++++++++++++++++++-----------
 1 file changed, 19 insertions(+), 12 deletions(-)

--- a/net/netfilter/x_tables.c
+++ b/net/netfilter/x_tables.c
@@ -726,27 +726,33 @@ void *xt_unregister_table(struct xt_table *table)
 EXPORT_SYMBOL_GPL(xt_unregister_table);

 #ifdef CONFIG_PROC_FS
+struct xt_names_priv {
+ struct seq_net_private p;
+ int af;
+};
 static void *xt_table_seq_start(struct seq_file *seq, loff_t *pos)
 {
- struct proc_dir_entry *pde = (struct proc_dir_entry *)seq->private;
- u_int16_t af = (unsigned long)pde->data;
+ struct xt_names_priv *priv = seq->private;
+ struct net *net = priv->p.net;
+ int af = priv->af;

  mutex_lock(&xt[af].mutex);
- return seq_list_start(&init_net.xt.tables[af], *pos);
+ return seq_list_start(&net->xt.tables[af], *pos);
 }

 static void *xt_table_seq_next(struct seq_file *seq, void *v, loff_t *pos)
 {
- struct proc_dir_entry *pde = (struct proc_dir_entry *)seq->private;
- u_int16_t af = (unsigned long)pde->data;
+ struct xt_names_priv *priv = seq->private;
+ struct net *net = priv->p.net;
+ int af = priv->af;

- return seq_list_next(v, &init_net.xt.tables[af], pos);
+ return seq_list_next(v, &net->xt.tables[af], pos);
 }

```
 static void xt_table_seq_stop(struct seq_file *seq, void *v)
 {
- struct proc_dir_entry *pde = seq->private;
- u_int16_t af = (unsigned long)pde->data;
+ struct xt_names_priv *priv = seq->private;
+ int af = priv->af;

  mutex_unlock(&xt[af].mutex);
 }
@@ -771,12 +777,13 @@ static const struct seq_operations xt_table_seq_ops = {
 static int xt_table_open(struct inode *inode, struct file *file)
 {
  int ret;
+ struct xt_names_priv *priv;

- ret = seq_open(file, &xt_table_seq_ops);
+ ret = seq_open_net(inode, file, &xt_table_seq_ops,
+       sizeof(struct xt_names_priv));
  if (!ret) {
-  struct seq_file *seq = file->private_data;
-
-  seq->private = PDE(inode);
+  priv = ((struct seq_file *)file->private_data)->private;
+  priv->af = (unsigned long)PDE(inode)->data;
  }
  return ret;
 }
```