Subject: [PATCH] ip{,6}_queue: convert to seq_file interface
Posted by Alexey Dobriyan on Fri, 19 Oct 2007 15:22:21 GMT
View Forum Message <> Reply to Message

I plan to kill ->get_info which means killing proc_net_create().

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

net/ipv4/netfilter/ip_queue.c | 36 ++++++++++++++++++++----------
net/ipv6/netfilter/ip6_queue.c | 36 ++++++++++++++++++++----------
2 files changed, 38 insertions(+), 34 deletions(-)

```
--- a/net/ipv4/netfilter/ip_queue.c
+++ b/net/ipv4/netfilter/ip_queue.c
@@ -22,6 +22,7 @@
 #include <linux/spinlock.h>
 #include <linux/sysctl.h>
 #include <linux/proc_fs.h>
+#include <linux/seq_file.h>
 #include <linux/security.h>
 #include <linux/mutex.h>
 #include <net/net_namespace.h>
@@ -607,15 +608,11 @@ static ctl_table ipq_root_table[] = {
  { .ctl_name = 0 }
 };

-#ifdef CONFIG_PROC_FS
-static int
-ipq_get_info(char *buffer, char **start, off_t offset, int length)
+static int ip_queue_show(struct seq_file *m, void *v)
 {
- int len;
-
  read_lock_bh(&queue_lock);

- len = sprintf(buffer,
+ seq_printf(m,
         "Peer PID          : %d\n"
         "Copy mode         : %hu\n"
         "Copy range        : %u\n"
@@ -632,16 +629,20 @@ ipq_get_info(char *buffer, char **start, off_t offset, int length)
         queue_user_dropped);

  read_unlock_bh(&queue_lock);
- return 0;
+}
```

```
- *start = buffer + offset;
- len -= offset;
- if (len > length)
-  len = length;
- else if (len < 0)
-  len = 0;
- return len;
+static int ip_queue_open(struct inode *inode, struct file *file)
+{
+ return single_open(file, ip_queue_show, NULL);
 }
-#endif /* CONFIG_PROC_FS */
+
+static const struct file_operations ip_queue_proc_fops = {
+ .open  = ip_queue_open,
+ .read  = seq_read,
+ .llseek  = seq_lseek,
+ .release = single_release,
+};

 static struct nf_queue_handler nfqh = {
  .name = "ip_queue",
@@ -661,10 +662,11 @@ static int __init ip_queue_init(void)
   goto cleanup_netlink_notifier;
  }

- proc = proc_net_create(&init_net, IPQ_PROC_FS_NAME, 0, ipq_get_info);
- if (proc)
+ proc = create_proc_entry(IPQ_PROC_FS_NAME, 0, init_net.proc_net);
+ if (proc) {
   proc->owner = THIS_MODULE;
- else {
+  proc->proc_fops = &ip_queue_proc_fops;
+ } else {
   printk(KERN_ERR "ip_queue: failed to create proc entry\n");
   goto cleanup_ipqnl;
  }
--- a/net/ipv6/netfilter/ip6_queue.c
+++ b/net/ipv6/netfilter/ip6_queue.c
@@ -23,6 +23,7 @@
 #include <linux/spinlock.h>
 #include <linux/sysctl.h>
 #include <linux/proc_fs.h>
+#include <linux/seq_file.h>
 #include <linux/mutex.h>
 #include <net/net_namespace.h>
 #include <net/sock.h>
@@ -596,15 +597,11 @@ static ctl_table ipq_root_table[] = {
```

```
 { .ctl_name = 0 }
};

-#ifdef CONFIG_PROC_FS
-static int
-ipq_get_info(char *buffer, char **start, off_t offset, int length)
+static int ip6_queue_show(struct seq_file *m, void *v)
 {
- int len;
-
  read_lock_bh(&queue_lock);

- len = sprintf(buffer,
+ seq_printf(m,
         "Peer PID          : %d\n"
         "Copy mode         : %hu\n"
         "Copy range        : %u\n"
@@ -621,16 +618,20 @@ ipq_get_info(char *buffer, char **start, off_t offset, int length)
         queue_user_dropped);

  read_unlock_bh(&queue_lock);
+ return 0;
+}

- *start = buffer + offset;
- len -= offset;
- if (len > length)
-  len = length;
- else if (len < 0)
-  len = 0;
- return len;
+static int ip6_queue_open(struct inode *inode, struct file *file)
+{
+ return single_open(file, ip6_queue_show, NULL);
 }
-#endif /* CONFIG_PROC_FS */
+
+static const struct file_operations ip6_queue_proc_fops = {
+ .open  = ip6_queue_open,
+ .read  = seq_read,
+ .llseek  = seq_lseek,
+ .release = single_release,
+};

 static struct nf_queue_handler nfqh = {
   .name = "ip6_queue",
@@ -650,10 +651,11 @@ static int __init ip6_queue_init(void)
    goto cleanup_netlink_notifier;
```

```
  }

- proc = proc_net_create(&init_net, IPQ_PROC_FS_NAME, 0, ipq_get_info);
- if (proc)
+ proc = create_proc_entry(IPQ_PROC_FS_NAME, 0, init_net.proc_net);
+ if (proc) {
   proc->owner = THIS_MODULE;
- else {
+  proc->proc_fops = &ip6_queue_proc_fops;
+ } else {
   printk(KERN_ERR "ip6_queue: failed to create proc entry\n");
   goto cleanup_ipqnl;
  }
```