
Subject: [PATCH 11/15] Make some netfilter-related proc files use seq_list_xxx helpers

Posted by [xemul](#) on Fri, 18 May 2007 09:55:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

This includes /proc/net/ip_conntrack_expect file.

Although struct nf_conntrack_expect has list_head as the very first element I use list_entry in .show callback to emphasize the fact that *v is the list_head pointer.

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

```
diff --git a/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
b/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
index 89f933e..bec843a 100644
```

```
--- a/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
+++ b/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
@@ -208,35 +208,15 @@ static const struct file_operations ct_f
/* expects */
```

```
static void *exp_seq_start(struct seq_file *s, loff_t *pos)
```

```
{
- struct list_head *e = &nf_conntrack_expect_list;
- loff_t i;
```

```
-
/* strange seq_file api calls stop even if we fail,
 * thus we need to grab lock since stop unlocks */
read_lock_bh(&nf_conntrack_lock);
```

```
-
- if (list_empty(e))
- return NULL;
```

```
-
- for (i = 0; i <= *pos; i++) {
- e = e->next;
- if (e == &nf_conntrack_expect_list)
- return NULL;
```

```
- }
- return e;
+ return seq_list_start(&nf_conntrack_expect_list, *pos);
}
```

```
static void *exp_seq_next(struct seq_file *s, void *v, loff_t *pos)
```

```
{
- struct list_head *e = v;
-
- ++*pos;
```

```

- e = e->next;
-
- if (e == &nf_conntrack_expect_list)
- return NULL;
-
- return e;
+ return seq_list_next(v, &nf_conntrack_expect_list, pos);
}

static void exp_seq_stop(struct seq_file *s, void *v)
@@ -246,7 +226,8 @@ static void exp_seq_stop(struct seq_file

static int exp_seq_show(struct seq_file *s, void *v)
{
- struct nf_conntrack_expect *exp = v;
+ struct nf_conntrack_expect *exp = list_entry(v,
+ struct nf_conntrack_expect, list);

if (exp->tuple.src.l3num != AF_INET)
return 0;
diff --git a/net/netfilter/nf_conntrack_expect.c b/net/netfilter/nf_conntrack_expect.c
index 117cbfd..7ea80e4 100644
--- a/net/netfilter/nf_conntrack_expect.c
+++ b/net/netfilter/nf_conntrack_expect.c
@@ -366,35 +366,15 @@ EXPORT_SYMBOL_GPL(nf_conntrack_expect_re
#ifdef CONFIG_PROC_FS
static void *exp_seq_start(struct seq_file *s, loff_t *pos)
{
- struct list_head *e = &nf_conntrack_expect_list;
- loff_t i;
-
/* strange seq_file api calls stop even if we fail,
 * thus we need to grab lock since stop unlocks */
read_lock_bh(&nf_conntrack_lock);
-
- if (list_empty(e))
- return NULL;
-
- for (i = 0; i <= *pos; i++) {
- e = e->next;
- if (e == &nf_conntrack_expect_list)
- return NULL;
- }
- return e;
+ return seq_list_start(&nf_conntrack_expect_list, *pos);
}

static void *exp_seq_next(struct seq_file *s, void *v, loff_t *pos)

```

```

{
- struct list_head *e = v;
-
- ++*pos;
- e = e->next;
-
- if (e == &nf_conntrack_expect_list)
- return NULL;
-
- return e;
+ return seq_list_next(v, &nf_conntrack_expect_list, pos);
}

static void exp_seq_stop(struct seq_file *s, void *v)
@@ -404,7 +384,8 @@ static void exp_seq_stop(struct seq_file

static int exp_seq_show(struct seq_file *s, void *v)
{
- struct nf_conntrack_expect *expect = v;
+ struct nf_conntrack_expect *expect = list_entry(v,
+ struct nf_conntrack_expect, list);

if (expect->timeout.function)
seq_printf(s, "%ld ", timer_pending(&expect->timeout)

```
