
Subject: [PATCH 3/6][NET-2.6.24] Make netfilter code use the seq_open_private
Posted by [Pavel Emelianov](#) on Tue, 09 Oct 2007 15:57:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

Just switch to the consolidated calls.

ipt_recent() has to initialize the private, so use
the __seq_open_private() helper.

Signed-off-by: Pavel Emelianov <xemul@openvz.org>
Cc: Patrick McHardy <kaber@trash.net>

```
diff --git a/net/ipv4/netfilter/ipt_recent.c b/net/ipv4/netfilter/ipt_recent.c
index db2a798..11d39fb 100644
--- a/net/ipv4/netfilter/ipt_recent.c
+++ b/net/ipv4/netfilter/ipt_recent.c
@@ -381,25 +381,14 @@ static const struct seq_operations recen
 static int recent_seq_open(struct inode *inode, struct file *file)
 {
     struct proc_dir_entry *pde = PDE(inode);
- struct seq_file *seq;
     struct recent_iter_state *st;
- int ret;

- st = kzalloc(sizeof(*st), GFP_KERNEL);
+ st = __seq_open_private(file, &recent_seq_ops, sizeof(*st));
     if (st == NULL)
         return -ENOMEM;

- ret = seq_open(file, &recent_seq_ops);
- if (ret) {
-     kfree(st);
-     goto out;
- }
-
     st->table = pde->data;
- seq = file->private_data;
- seq->private = st;
-out:
- return ret;
+ return 0;
 }

 static ssize_t recent_proc_write(struct file *file, const char __user *input,
diff --git a/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
b/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
```

index a5ae2ea..741f3df 100644

--- a/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c

+++ b/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c

@@ -174,22 +174,8 @@ static const struct seq_operations ct_se

```
static int ct_open(struct inode *inode, struct file *file)
{
- struct seq_file *seq;
- struct ct_iter_state *st;
- int ret;
-
- st = kzalloc(sizeof(struct ct_iter_state), GFP_KERNEL);
- if (st == NULL)
- return -ENOMEM;
- ret = seq_open(file, &ct_seq_ops);
- if (ret)
- goto out_free;
- seq = file->private_data;
- seq->private = st;
- return ret;
-out_free:
- kfree(st);
- return ret;
+ return seq_open_private(file, &ct_seq_ops,
+ sizeof(struct ct_iter_state));
}
```

static const struct file_operations ct_file_ops = {
@@ -291,22 +277,8 @@ static const struct seq_operations exp_s

```
static int exp_open(struct inode *inode, struct file *file)
{
- struct seq_file *seq;
- struct ct_expect_iter_state *st;
- int ret;
-
- st = kzalloc(sizeof(struct ct_expect_iter_state), GFP_KERNEL);
- if (!st)
- return -ENOMEM;
- ret = seq_open(file, &exp_seq_ops);
- if (ret)
- goto out_free;
- seq = file->private_data;
- seq->private = st;
- return ret;
-out_free:
- kfree(st);
- return ret;
```

```
+ return seq_open_private(file, &exp_seq_ops,
+ sizeof(struct ct_expect_iter_state));
}
```

```
static const struct file_operations ip_exp_file_ops = {
diff --git a/net/netfilter/nf_conntrack_expect.c b/net/netfilter/nf_conntrack_expect.c
index 7a0ae36..175c8d1 100644
--- a/net/netfilter/nf_conntrack_expect.c
+++ b/net/netfilter/nf_conntrack_expect.c
@@ -472,22 +472,8 @@ static const struct seq_operations exp_s
```

```
static int exp_open(struct inode *inode, struct file *file)
{
- struct seq_file *seq;
- struct ct_expect_iter_state *st;
- int ret;
-
- st = kzalloc(sizeof(struct ct_expect_iter_state), GFP_KERNEL);
- if (!st)
- return -ENOMEM;
- ret = seq_open(file, &exp_seq_ops);
- if (ret)
- goto out_free;
- seq = file->private_data;
- seq->private = st;
- return ret;
-out_free:
- kfree(st);
- return ret;
+ return seq_open_private(file, &exp_seq_ops,
+ sizeof(struct ct_expect_iter_state));
}
```

```
static const struct file_operations exp_file_ops = {
diff --git a/net/netfilter/nf_conntrack_standalone.c b/net/netfilter/nf_conntrack_standalone.c
index 2a19c5f..9efdd37 100644
--- a/net/netfilter/nf_conntrack_standalone.c
+++ b/net/netfilter/nf_conntrack_standalone.c
@@ -195,22 +195,8 @@ static const struct seq_operations ct_se
```

```
static int ct_open(struct inode *inode, struct file *file)
{
- struct seq_file *seq;
- struct ct_iter_state *st;
- int ret;
-
- st = kzalloc(sizeof(struct ct_iter_state), GFP_KERNEL);
- if (st == NULL)
```

```

- return -ENOMEM;
- ret = seq_open(file, &ct_seq_ops);
- if (ret)
- goto out_free;
- seq      = file->private_data;
- seq->private = st;
- return ret;
-out_free:
- kfree(st);
- return ret;
+ return seq_open_private(file, &ct_seq_ops,
+   sizeof(struct ct_iter_state));
}

```

```

static const struct file_operations ct_file_ops = {
diff --git a/net/netfilter/nfnetlink_log.c b/net/netfilter/nfnetlink_log.c
index 2135926..2c7bd2e 100644
--- a/net/netfilter/nfnetlink_log.c
+++ b/net/netfilter/nfnetlink_log.c
@@ -951,22 +951,8 @@ static const struct seq_operations nful_

```

```

static int nful_open(struct inode *inode, struct file *file)
{
- struct seq_file *seq;
- struct iter_state *is;
- int ret;
-
- is = kzalloc(sizeof(*is), GFP_KERNEL);
- if (!is)
- return -ENOMEM;
- ret = seq_open(file, &nful_seq_ops);
- if (ret < 0)
- goto out_free;
- seq = file->private_data;
- seq->private = is;
- return ret;
-out_free:
- kfree(is);
- return ret;
+ return seq_open_private(file, &nful_seq_ops,
+   sizeof(struct iter_state));
}

```

```

static const struct file_operations nful_file_ops = {
diff --git a/net/netfilter/nfnetlink_queue.c b/net/netfilter/nfnetlink_queue.c
index 48e095a..49f0480 100644
--- a/net/netfilter/nfnetlink_queue.c
+++ b/net/netfilter/nfnetlink_queue.c

```

@@ -1051,22 +1051,8 @@ static const struct seq_operations nfqnl

```
static int nfqnl_open(struct inode *inode, struct file *file)
```

```
{  
- struct seq_file *seq;  
- struct iter_state *is;  
- int ret;  
-  
- is = kzalloc(sizeof(*is), GFP_KERNEL);  
- if (!is)  
- return -ENOMEM;  
- ret = seq_open(file, &nfqnl_seq_ops);  
- if (ret < 0)  
- goto out_free;  
- seq = file->private_data;  
- seq->private = is;  
- return ret;  
-out_free:  
- kfree(is);  
- return ret;  
+ return seq_open_private(file, &nfqnl_seq_ops,  
+ sizeof(struct iter_state));  
}
```

```
static const struct file_operations nfqnl_file_ops = {
```
