Subject: [PATCH 2/2][NETFILTER] Use the list_for_each_entry in nf_sockopt.c
Posted by Pavel Emelianov on Thu, 01 Nov 2007 15:58:32 GMT
View Forum Message <> Reply to Message

The list_head pointer, used to iterate over the list, is not used
at all, but to get the struct nf_sockopt_ops pointer (and actually
not in the 100% clean way).

So use the list_for_each_entry, removing one unneeded variable
from each place of use.

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

```
diff --git a/net/netfilter/nf_sockopt.c b/net/netfilter/nf_sockopt.c
index a5e5e30..87bc144 100644
--- a/net/netfilter/nf_sockopt.c
+++ b/net/netfilter/nf_sockopt.c
@@ -23,14 +23,13 @@ static inline int overlap(int min1, int max1, int min2, int max2)
 /* Functions to register sockopt ranges (exclusive). */
 int nf_register_sockopt(struct nf_sockopt_ops *reg)
 {
- struct list_head *i;
+ struct nf_sockopt_ops *ops;
  int ret = 0;

  if (mutex_lock_interruptible(&nf_sockopt_mutex) != 0)
   return -EINTR;

- list_for_each(i, &nf_sockopts) {
-  struct nf_sockopt_ops *ops = (struct nf_sockopt_ops *)i;
+ list_for_each_entry(ops, &nf_sockopts, list) {
   if (ops->pf == reg->pf
       && (overlap(ops->set_optmin, ops->set_optmax,
     reg->set_optmin, reg->set_optmax)
@@ -64,7 +63,6 @@ EXPORT_SYMBOL(nf_unregister_sockopt);
 static struct nf_sockopt_ops *nf_sockopt_find(struct sock *sk, int pf,
   int val, int get)
 {
- struct list_head *i;
  struct nf_sockopt_ops *ops;

  if (sk->sk_net != &init_net)
@@ -73,8 +71,7 @@ static struct nf_sockopt_ops *nf_sockopt_find(struct sock *sk, int pf,
  if (mutex_lock_interruptible(&nf_sockopt_mutex) != 0)
   return ERR_PTR(-EINTR);
```

```
- list_for_each(i, &nf_sockopts) {
-  ops = (struct nf_sockopt_ops *)i;
+ list_for_each_entry(ops, &nf_sockopts, list) {
   if (ops->pf == pf) {
    if (!try_module_get(ops->owner))
     goto out_nosup;
--
1.5.3.4
```