Subject: [PATCH net-2.6.25 6/6][NETFILTER] Use the ctl paths instead of hand-made analogue
Posted by Pavel Emelianov on Tue, 08 Jan 2008 16:09:41 GMT
View Forum Message <> Reply to Message

The conntracks subsystem has a similar infrastructure
to maintain ctl_paths, but since we already have it
on the generic level, I think it's OK to switch to
using it.

So, basically, this patch just replaces the ctl_table-s
with ctl_path-s, nf_register_sysctl_table with
register_sysctl_paths() and removes no longer needed code.

After this the net/netfilter/nf_sysctl.c file contains
the paths only.

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

---

 include/linux/netfilter.h | 8 -
 include/net/netfilter/nf_conntrack_l3proto.h | 2
 net/netfilter/nf_conntrack_proto.c | 7 -
 net/netfilter/nf_sysctl.c | 127 +-------------------------
 4 files changed, 16 insertions(+), 128 deletions(-)

diff --git a/include/linux/netfilter.h b/include/linux/netfilter.h
index d190d56..c41f643 100644
--- a/include/linux/netfilter.h
+++ b/include/linux/netfilter.h
@@ -120,12 +120,8 @@ void nf_unregister_sockopt(struct nf_sockopt_ops *reg);

 #ifdef CONFIG_SYSCTL
 /* Sysctl registration */
-struct ctl_table_header *nf_register_sysctl_table(struct ctl_table *path,
-	struct ctl_table *table);
-void nf_unregister_sysctl_table(struct ctl_table_header *header,
-	struct ctl_table *table);
-extern struct ctl_table nf_net_netfilter_sysctl_path[];
-extern struct ctl_table nf_net_ipv4_netfilter_sysctl_path[];
+extern struct ctl_path nf_net_netfilter_sysctl_path[];
+extern struct ctl_path nf_net_ipv4_netfilter_sysctl_path[];
 #endif /* CONFIG_SYSCTL */

 extern struct list_head nf_hooks[NPROTO][NF_MAX_HOOKS];
diff --git a/include/net/netfilter/nf_conntrack_l3proto.h b/include/net/netfilter/nf_conntrack_l3proto.h
index 15888fc..875c6d4 100644

```
--- a/include/net/netfilter/nf_conntrack_l3proto.h
+++ b/include/net/netfilter/nf_conntrack_l3proto.h
@@ -73,7 +73,7 @@ struct nf_conntrack_l3proto

 #ifdef CONFIG_SYSCTL
  struct ctl_table_header *ctl_table_header;
- struct ctl_table *ctl_table_path;
+ struct ctl_path  *ctl_table_path;
  struct ctl_table *ctl_table;
 #endif /* CONFIG_SYSCTL */

diff --git a/net/netfilter/nf_conntrack_proto.c b/net/netfilter/nf_conntrack_proto.c
index 6d94706..8595b59 100644
--- a/net/netfilter/nf_conntrack_proto.c
+++ b/net/netfilter/nf_conntrack_proto.c
@@ -36,11 +36,11 @@ static DEFINE_MUTEX(nf_ct_proto_mutex);

 #ifdef CONFIG_SYSCTL
 static int
-nf_ct_register_sysctl(struct ctl_table_header **header, struct ctl_table *path,
+nf_ct_register_sysctl(struct ctl_table_header **header, struct ctl_path *path,
         struct ctl_table *table, unsigned int *users)
 {
  if (*header == NULL) {
-  *header = nf_register_sysctl_table(path, table);
+  *header = register_sysctl_paths(path, table);
   if (*header == NULL)
    return -ENOMEM;
  }
@@ -55,7 +55,8 @@ nf_ct_unregister_sysctl(struct ctl_table_header **header,
 {
  if (users != NULL && --*users > 0)
   return;
- nf_unregister_sysctl_table(*header, table);
+
+ unregister_sysctl_table(*header);
  *header = NULL;
 }
 #endif
diff --git a/net/netfilter/nf_sysctl.c b/net/netfilter/nf_sysctl.c
index ee34589..d9fcc89 100644
--- a/net/netfilter/nf_sysctl.c
+++ b/net/netfilter/nf_sysctl.c
@@ -7,128 +7,19 @@
 #include <linux/string.h>
 #include <linux/slab.h>

-static void
```

```
-path_free(struct ctl_table *path, struct ctl_table *table)
-{
- struct ctl_table *t, *next;
-
- for (t = path; t != NULL && t != table; t = next) {
-  next = t->child;
-  kfree(t);
- }
-}
-
-static struct ctl_table *
-path_dup(struct ctl_table *path, struct ctl_table *table)
-{
- struct ctl_table *t, *last = NULL, *tmp;
-
- for (t = path; t != NULL; t = t->child) {
-  /* twice the size since path elements are terminated by an
-   * empty element */
-  tmp = kmemdup(t, 2 * sizeof(*t), GFP_KERNEL);
-  if (tmp == NULL) {
-   if (last != NULL)
-    path_free(path, table);
-   return NULL;
-  }
-
-  if (last != NULL)
-   last->child = tmp;
-  else
-   path = tmp;
-  last = tmp;
- }
-
- if (last != NULL)
-  last->child = table;
- else
-  path = table;
-
- return path;
-}
-
-struct ctl_table_header *
-nf_register_sysctl_table(struct ctl_table *path, struct ctl_table *table)
-{
- struct ctl_table_header *header;
-
- path = path_dup(path, table);
- if (path == NULL)
-  return NULL;
```

```
- header = register_sysctl_table(path);
- if (header == NULL)
-  path_free(path, table);
- return header;
-}
-EXPORT_SYMBOL_GPL(nf_register_sysctl_table);
-
-void
-nf_unregister_sysctl_table(struct ctl_table_header *header,
-      struct ctl_table *table)
-{
- struct ctl_table *path = header->ctl_table;
-
- unregister_sysctl_table(header);
- path_free(path, table);
-}
-EXPORT_SYMBOL_GPL(nf_unregister_sysctl_table);
-
 /* net/netfilter */
-static struct ctl_table nf_net_netfilter_table[] = {
- {
-  .ctl_name = NET_NETFILTER,
-  .procname = "netfilter",
-  .mode  = 0555,
- },
- {
-  .ctl_name = 0
- }
-};
-struct ctl_table nf_net_netfilter_sysctl_path[] = {
- {
-  .ctl_name = CTL_NET,
-  .procname = "net",
-  .mode  = 0555,
-  .child  = nf_net_netfilter_table,
- },
- {
-  .ctl_name = 0
- }
+struct ctl_path nf_net_netfilter_sysctl_path[] = {
+ { .procname = "net", .ctl_name = CTL_NET, },
+ { .procname = "netfilter", .ctl_name = NET_NETFILTER, },
+ { }
 };
 EXPORT_SYMBOL_GPL(nf_net_netfilter_sysctl_path);

 /* net/ipv4/netfilter */
-static struct ctl_table nf_net_ipv4_netfilter_table[] = {
```

```
- {
-  .ctl_name = NET_IPV4_NETFILTER,
-  .procname = "netfilter",
-  .mode  = 0555,
- },
- {
-  .ctl_name = 0
- }
-};
-static struct ctl_table nf_net_ipv4_table[] = {
- {
-  .ctl_name = NET_IPV4,
-  .procname = "ipv4",
-  .mode  = 0555,
-  .child  = nf_net_ipv4_netfilter_table,
- },
- {
-  .ctl_name = 0
- }
-};
-struct ctl_table nf_net_ipv4_netfilter_sysctl_path[] = {
- {
-  .ctl_name = CTL_NET,
-  .procname = "net",
-  .mode  = 0555,
-  .child  = nf_net_ipv4_table,
- },
- {
-  .ctl_name = 0
- }
+struct ctl_path nf_net_ipv4_netfilter_sysctl_path[] = {
+ { .procname = "net", .ctl_name = CTL_NET, },
+ { .procname = "ipv4", .ctl_name = NET_IPV4, },
+ { .procname = "netfilter", .ctl_name = NET_IPV4_NETFILTER, },
+ { }
 };
 EXPORT_SYMBOL_GPL(nf_net_ipv4_netfilter_sysctl_path);
```