Subject: [PATCH net-2.6.25 5/6][NETFILTER] Switch to using ctl_paths in nf_queue and conntrack modules
Posted by Pavel Emelianov on Tue, 08 Jan 2008 16:06:23 GMT
View Forum Message <> Reply to Message

This includes the most simple cases for netfilter.

The first part is tne queue modules for ipv4 and ipv6,
on which the net/ipv4/ and net/ipv6/ paths are reused
from the appropriate ipv4 and ipv6 code.

The conntrack module is also patched, but this hunk is
very small and simple.

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

```
diff --git a/include/net/ip.h b/include/net/ip.h
index 8be48c8..2ad4d2f 100644
--- a/include/net/ip.h
+++ b/include/net/ip.h
@@ -177,6 +177,8 @@ extern void inet_get_local_port_range(int *low, int *high);
 extern int sysctl_ip_default_ttl;
 extern int sysctl_ip_nonlocal_bind;

+extern struct ctl_path net_ipv4_ctl_path[];
+
 /* From ip_fragment.c */
 struct inet_frags_ctl;
 extern struct inet_frags_ctl ip4_frags_ctl;
diff --git a/include/net/ipv6.h b/include/net/ipv6.h
index f2adedf..e371f32 100644
--- a/include/net/ipv6.h
+++ b/include/net/ipv6.h
@@ -112,6 +112,8 @@ struct frag_hdr {
 extern int sysctl_ipv6_bindv6only;
 extern int sysctl_mld_max_msf;

+extern struct ctl_path net_ipv6_ctl_path[];
+
 #define _DEVINC(statname, modifier, idev, field)   \
 ({         \
  struct inet6_dev *_idev = (idev);    \
diff --git a/net/ipv4/netfilter/ip_queue.c b/net/ipv4/netfilter/ip_queue.c
index 68b12ce..7361315 100644
--- a/net/ipv4/netfilter/ip_queue.c
+++ b/net/ipv4/netfilter/ip_queue.c
```

```
@@ -29,6 +29,7 @@
 #include <net/sock.h>
 #include <net/route.h>
 #include <net/netfilter/nf_queue.h>
+#include <net/ip.h>

 #define IPQ_QMAX_DEFAULT 1024
 #define IPQ_PROC_FS_NAME "ip_queue"
@@ -525,26 +526,6 @@ static ctl_table ipq_table[] = {
  { .ctl_name = 0 }
 };

-static ctl_table ipq_dir_table[] = {
- {
-  .ctl_name = NET_IPV4,
-  .procname = "ipv4",
-  .mode  = 0555,
-  .child  = ipq_table
- },
- { .ctl_name = 0 }
-};
-
-static ctl_table ipq_root_table[] = {
- {
-  .ctl_name = CTL_NET,
-  .procname = "net",
-  .mode  = 0555,
-  .child  = ipq_dir_table
- },
- { .ctl_name = 0 }
-};
-
 static int ip_queue_show(struct seq_file *m, void *v)
 {
  read_lock_bh(&queue_lock);
@@ -610,7 +591,7 @@ static int __init ip_queue_init(void)
  }

  register_netdevice_notifier(&ipq_dev_notifier);
- ipq_sysctl_header = register_sysctl_table(ipq_root_table);
+ ipq_sysctl_header = register_sysctl_paths(net_ipv4_ctl_path, ipq_table);

  status = nf_register_queue_handler(PF_INET, &nfqh);
  if (status < 0) {
diff --git a/net/ipv4/sysctl_net_ipv4.c b/net/ipv4/sysctl_net_ipv4.c
index a5a9f8e..45536a9 100644
--- a/net/ipv4/sysctl_net_ipv4.c
+++ b/net/ipv4/sysctl_net_ipv4.c
```

```
@@ -846,17 +846,18 @@ static struct ctl_table ipv4_table[] = {
  { .ctl_name = 0 }
 };

-static __initdata struct ctl_path net_ipv4_path[] = {
+struct ctl_path net_ipv4_ctl_path[] = {
  { .procname = "net", .ctl_name = CTL_NET, },
  { .procname = "ipv4", .ctl_name = NET_IPV4, },
  { },
 };
+EXPORT_SYMBOL_GPL(net_ipv4_ctl_path);

 static __init int sysctl_ipv4_init(void)
 {
  struct ctl_table_header *hdr;

- hdr = register_sysctl_paths(net_ipv4_path, ipv4_table);
+ hdr = register_sysctl_paths(net_ipv4_ctl_path, ipv4_table);
  return hdr == NULL ? -ENOMEM : 0;
 }

diff --git a/net/ipv6/netfilter/ip6_queue.c b/net/ipv6/netfilter/ip6_queue.c
index e5b0059..a20db0b 100644
--- a/net/ipv6/netfilter/ip6_queue.c
+++ b/net/ipv6/netfilter/ip6_queue.c
@@ -529,26 +529,6 @@ static ctl_table ipq_table[] = {
  { .ctl_name = 0 }
 };

-static ctl_table ipq_dir_table[] = {
- {
-  .ctl_name = NET_IPV6,
-  .procname = "ipv6",
-  .mode  = 0555,
-  .child  = ipq_table
- },
- { .ctl_name = 0 }
-};
-
-static ctl_table ipq_root_table[] = {
- {
-  .ctl_name = CTL_NET,
-  .procname = "net",
-  .mode  = 0555,
-  .child  = ipq_dir_table
- },
- { .ctl_name = 0 }
-};
```

```
-
 static int ip6_queue_show(struct seq_file *m, void *v)
 {
  read_lock_bh(&queue_lock);
@@ -614,7 +594,7 @@ static int __init ip6_queue_init(void)
  }

  register_netdevice_notifier(&ipq_dev_notifier);
- ipq_sysctl_header = register_sysctl_table(ipq_root_table);
+ ipq_sysctl_header = register_sysctl_paths(net_ipv6_ctl_path, ipq_table);

  status = nf_register_queue_handler(PF_INET6, &nfqh);
  if (status < 0) {
diff --git a/net/ipv6/sysctl_net_ipv6.c b/net/ipv6/sysctl_net_ipv6.c
index 0b5bec3..4ad8d9d 100644
--- a/net/ipv6/sysctl_net_ipv6.c
+++ b/net/ipv6/sysctl_net_ipv6.c
@@ -82,17 +82,19 @@ static ctl_table ipv6_table[] = {
  { .ctl_name = 0 }
 };

-static struct ctl_path ipv6_ctl_path[] = {
+struct ctl_path net_ipv6_ctl_path[] = {
  { .procname = "net", .ctl_name = CTL_NET, },
  { .procname = "ipv6", .ctl_name = NET_IPV6, },
  { },
 };
+EXPORT_SYMBOL_GPL(net_ipv6_ctl_path);

 static struct ctl_table_header *ipv6_sysctl_header;

 void ipv6_sysctl_register(void)
 {
- ipv6_sysctl_header = register_sysctl_paths(ipv6_ctl_path, ipv6_table);
+ ipv6_sysctl_header = register_sysctl_paths(net_ipv6_ctl_path,
+   ipv6_table);
 }

 void ipv6_sysctl_unregister(void)
diff --git a/net/netfilter/nf_conntrack_standalone.c b/net/netfilter/nf_conntrack_standalone.c
index 9efdd37..2ad4933 100644
--- a/net/netfilter/nf_conntrack_standalone.c
+++ b/net/netfilter/nf_conntrack_standalone.c
@@ -383,15 +383,11 @@ static ctl_table nf_ct_netfilter_table[] = {
  { .ctl_name = 0 }
 };

-static ctl_table nf_ct_net_table[] = {
```

```
- {
-  .ctl_name = CTL_NET,
-  .procname = "net",
-  .mode  = 0555,
-  .child  = nf_ct_netfilter_table,
- },
- { .ctl_name = 0 }
+struct ctl_path nf_ct_path[] = {
+ { .procname = "net", .ctl_name = CTL_NET, },
+ { }
 };
+
 EXPORT_SYMBOL_GPL(nf_ct_log_invalid);
 #endif /* CONFIG_SYSCTL */

@@ -418,7 +414,8 @@ static int __init nf_conntrack_standalone_init(void)
  proc_stat->owner = THIS_MODULE;
 #endif
 #ifdef CONFIG_SYSCTL
- nf_ct_sysctl_header = register_sysctl_table(nf_ct_net_table);
+ nf_ct_sysctl_header = register_sysctl_paths(nf_ct_path,
+    nf_ct_netfilter_table);
  if (nf_ct_sysctl_header == NULL) {
   printk("nf_conntrack: can't register to sysctl.\n");
   ret = -ENOMEM;
```