Subject: [PATCH 4/5] netns netfilter: propagate netns from userspace
Posted by Alexey Dobriyan on Mon, 21 Jan 2008 14:54:24 GMT
View Forum Message <> Reply to Message

.. all the way down to table searching functions.

Signed-off-by: Alexey Dobriyan <adobriyan@sw.ru>
---

 net/ipv4/netfilter/ip_tables.c | 46 ++----------
 1 file changed, 23 insertions(+), 23 deletions(-)

--- a/net/ipv4/netfilter/ip_tables.c
+++ b/net/ipv4/netfilter/ip_tables.c
@@ -1092,7 +1092,7 @@ static int compat_table_info(const struct xt_table_info *info,
 }
 #endif

-static int get_info(void __user *user, int *len, int compat)
+static int get_info(struct net *net, void __user *user, int *len, int compat)
 {
  char name[IPT_TABLE_MAXNAMELEN];
  struct xt_table *t;
@@ -1112,7 +1112,7 @@ static int get_info(void __user *user, int *len, int compat)
  if (compat)
   xt_compat_lock(AF_INET);
 #endif
- t = try_then_request_module(xt_find_table_lock(&init_net, AF_INET, name),
+ t = try_then_request_module(xt_find_table_lock(net, AF_INET, name),
         "iptable_%s", name);
  if (t && !IS_ERR(t)) {
   struct ipt_getinfo info;
@@ -1152,7 +1152,7 @@ static int get_info(void __user *user, int *len, int compat)
 }

 static int
-get_entries(struct ipt_get_entries __user *uptr, int *len)
+get_entries(struct net *net, struct ipt_get_entries __user *uptr, int *len)
 {
  int ret;
  struct ipt_get_entries get;
@@ -1170,7 +1170,7 @@ get_entries(struct ipt_get_entries __user *uptr, int *len)
   return -EINVAL;
  }

- t = xt_find_table_lock(&init_net, AF_INET, get.name);
+ t = xt_find_table_lock(net, AF_INET, get.name);
  if (t && !IS_ERR(t)) {

```
   struct xt_table_info *private = t->private;
   duprintf("t->private->number = %u\n", private->number);
@@ -1191,7 +1191,7 @@ get_entries(struct ipt_get_entries __user *uptr, int *len)
 }

 static int
-__do_replace(const char *name, unsigned int valid_hooks,
+__do_replace(struct net *net, const char *name, unsigned int valid_hooks,
       struct xt_table_info *newinfo, unsigned int num_counters,
       void __user *counters_ptr)
 {
@@ -1208,7 +1208,7 @@ __do_replace(const char *name, unsigned int valid_hooks,
   goto out;
  }

- t = try_then_request_module(xt_find_table_lock(&init_net, AF_INET, name),
+ t = try_then_request_module(xt_find_table_lock(net, AF_INET, name),
         "iptable_%s", name);
  if (!t || IS_ERR(t)) {
   ret = t ? PTR_ERR(t) : -ENOENT;
@@ -1261,7 +1261,7 @@ __do_replace(const char *name, unsigned int valid_hooks,
 }

 static int
-do_replace(void __user *user, unsigned int len)
+do_replace(struct net *net, void __user *user, unsigned int len)
 {
  int ret;
  struct ipt_replace tmp;
@@ -1295,7 +1295,7 @@ do_replace(void __user *user, unsigned int len)

  duprintf("ip_tables: Translated table\n");

- ret = __do_replace(tmp.name, tmp.valid_hooks, newinfo,
+ ret = __do_replace(net, tmp.name, tmp.valid_hooks, newinfo,
       tmp.num_counters, tmp.counters);
  if (ret)
   goto free_newinfo_untrans;
@@ -1331,7 +1331,7 @@ add_counter_to_entry(struct ipt_entry *e,
 }

 static int
-do_add_counters(void __user *user, unsigned int len, int compat)
+do_add_counters(struct net *net, void __user *user, unsigned int len, int compat)
 {
  unsigned int i;
  struct xt_counters_info tmp;
@@ -1383,7 +1383,7 @@ do_add_counters(void __user *user, unsigned int len, int compat)
```

```
   goto free;
  }

- t = xt_find_table_lock(&init_net, AF_INET, name);
+ t = xt_find_table_lock(net, AF_INET, name);
  if (!t || IS_ERR(t)) {
   ret = t ? PTR_ERR(t) : -ENOENT;
   goto free;
@@ -1789,7 +1789,7 @@ out_unlock:
 }

 static int
-compat_do_replace(void __user *user, unsigned int len)
+compat_do_replace(struct net *net, void __user *user, unsigned int len)
 {
  int ret;
  struct compat_ipt_replace tmp;
@@ -1826,7 +1826,7 @@ compat_do_replace(void __user *user, unsigned int len)

  duprintf("compat_do_replace: Translated table\n");

- ret = __do_replace(tmp.name, tmp.valid_hooks, newinfo,
+ ret = __do_replace(net, tmp.name, tmp.valid_hooks, newinfo,
       tmp.num_counters, compat_ptr(tmp.counters));
  if (ret)
   goto free_newinfo_untrans;
@@ -1850,11 +1850,11 @@ compat_do_ipt_set_ctl(struct sock *sk, int cmd, void __user *user,

  switch (cmd) {
  case IPT_SO_SET_REPLACE:
-  ret = compat_do_replace(user, len);
+  ret = compat_do_replace(sk->sk_net, user, len);
   break;

  case IPT_SO_SET_ADD_COUNTERS:
-  ret = do_add_counters(user, len, 1);
+  ret = do_add_counters(sk->sk_net, user, len, 1);
   break;

  default:
@@ -1903,7 +1903,8 @@ compat_copy_entries_to_user(unsigned int total_size, struct xt_table
*table,
 }

 static int
-compat_get_entries(struct compat_ipt_get_entries __user *uptr, int *len)
+compat_get_entries(struct net *net, struct compat_ipt_get_entries __user *uptr,
+      int *len)
```

```
 {
  int ret;
  struct compat_ipt_get_entries get;
@@ -1924,7 +1925,7 @@ compat_get_entries(struct compat_ipt_get_entries __user *uptr, int
*len)
  }

  xt_compat_lock(AF_INET);
- t = xt_find_table_lock(&init_net, AF_INET, get.name);
+ t = xt_find_table_lock(net, AF_INET, get.name);
  if (t && !IS_ERR(t)) {
   struct xt_table_info *private = t->private;
   struct xt_table_info info;
@@ -1960,10 +1961,10 @@ compat_do_ipt_get_ctl(struct sock *sk, int cmd, void __user *user,
int *len)

  switch (cmd) {
  case IPT_SO_GET_INFO:
-  ret = get_info(user, len, 1);
+  ret = get_info(sk->sk_net, user, len, 1);
   break;
  case IPT_SO_GET_ENTRIES:
-  ret = compat_get_entries(user, len);
+  ret = compat_get_entries(sk->sk_net, user, len);
   break;
  default:
   ret = do_ipt_get_ctl(sk, cmd, user, len);
@@ -1982,11 +1983,11 @@ do_ipt_set_ctl(struct sock *sk, int cmd, void __user *user, unsigned
int len)

  switch (cmd) {
  case IPT_SO_SET_REPLACE:
-  ret = do_replace(user, len);
+  ret = do_replace(sk->sk_net, user, len);
   break;

  case IPT_SO_SET_ADD_COUNTERS:
-  ret = do_add_counters(user, len, 0);
+  ret = do_add_counters(sk->sk_net, user, len, 0);
   break;

  default:
@@ -2007,11 +2008,11 @@ do_ipt_get_ctl(struct sock *sk, int cmd, void __user *user, int *len)

  switch (cmd) {
  case IPT_SO_GET_INFO:
-  ret = get_info(user, len, 0);
+  ret = get_info(sk->sk_net, user, len, 0);
```

```
   break;

  case IPT_SO_GET_ENTRIES:
-  ret = get_entries(user, len);
+  ret = get_entries(sk->sk_net, user, len);
   break;

  case IPT_SO_GET_REVISION_MATCH:
```