Subject: [PATCH 1/5] netns netfilter: change xt_table_register() return value convention
Posted by Alexey Dobriyan on Mon, 21 Jan 2008 14:52:05 GMT
View Forum Message <> Reply to Message

Switch from 0/-E to ptr/PTR_ERR convention.

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

 include/linux/netfilter/x_tables.h | 6 +++---
 net/ipv4/netfilter/arp_tables.c | 7 ++++---
 net/ipv4/netfilter/ip_tables.c | 7 ++++---
 net/ipv6/netfilter/ip6_tables.c | 7 ++++---
 net/netfilter/x_tables.c | 15 ++++++++-------
 5 files changed, 23 insertions(+), 19 deletions(-)

```
--- a/include/linux/netfilter/x_tables.h
+++ b/include/linux/netfilter/x_tables.h
@@ -335,9 +335,9 @@ extern int xt_check_target(const struct xt_target *target, unsigned short
family
       unsigned int size, const char *table, unsigned int hook,
       unsigned short proto, int inv_proto);

-extern int xt_register_table(struct xt_table *table,
-       struct xt_table_info *bootstrap,
-       struct xt_table_info *newinfo);
+extern struct xt_table *xt_register_table(struct xt_table *table,
+       struct xt_table_info *bootstrap,
+       struct xt_table_info *newinfo);
 extern void *xt_unregister_table(struct xt_table *table);

 extern struct xt_table_info *xt_replace_table(struct xt_table *table,
--- a/net/ipv4/netfilter/arp_tables.c
+++ b/net/ipv4/netfilter/arp_tables.c
@@ -1727,6 +1727,7 @@ int arpt_register_table(struct arpt_table *table,
  struct xt_table_info bootstrap
   = { 0, 0, 0, { 0 }, { 0 }, { } };
  void *loc_cpu_entry;
+ struct xt_table *new_table;

  newinfo = xt_alloc_table_info(repl->size);
  if (!newinfo) {
@@ -1750,10 +1751,10 @@ int arpt_register_table(struct arpt_table *table,
   return ret;
  }

- ret = xt_register_table(table, &bootstrap, newinfo);
```

```
- if (ret != 0) {
+ new_table = xt_register_table(table, &bootstrap, newinfo);
+ if (IS_ERR(new_table)) {
   xt_free_table_info(newinfo);
-  return ret;
+  return PTR_ERR(new_table);
  }

  return 0;
--- a/net/ipv4/netfilter/ip_tables.c
+++ b/net/ipv4/netfilter/ip_tables.c
@@ -2055,6 +2055,7 @@ int ipt_register_table(struct xt_table *table, const struct ipt_replace
*repl)
  struct xt_table_info bootstrap
   = { 0, 0, 0, { 0 }, { 0 }, { } };
  void *loc_cpu_entry;
+ struct xt_table *new_table;

  newinfo = xt_alloc_table_info(repl->size);
  if (!newinfo)
@@ -2074,10 +2075,10 @@ int ipt_register_table(struct xt_table *table, const struct ipt_replace
*repl)
   return ret;
  }

- ret = xt_register_table(table, &bootstrap, newinfo);
- if (ret != 0) {
+ new_table = xt_register_table(table, &bootstrap, newinfo);
+ if (IS_ERR(new_table)) {
   xt_free_table_info(newinfo);
-  return ret;
+  return PTR_ERR(new_table);
  }

  return 0;
--- a/net/ipv6/netfilter/ip6_tables.c
+++ b/net/ipv6/netfilter/ip6_tables.c
@@ -2081,6 +2081,7 @@ int ip6t_register_table(struct xt_table *table, const struct ip6t_replace
*repl)
  struct xt_table_info bootstrap
   = { 0, 0, 0, { 0 }, { 0 }, { } };
  void *loc_cpu_entry;
+ struct xt_table *new_table;

  newinfo = xt_alloc_table_info(repl->size);
  if (!newinfo)
@@ -2100,10 +2101,10 @@ int ip6t_register_table(struct xt_table *table, const struct
ip6t_replace *repl)
```

```
   return ret;
  }

- ret = xt_register_table(table, &bootstrap, newinfo);
- if (ret != 0) {
+ new_table = xt_register_table(table, &bootstrap, newinfo);
+ if (IS_ERR(new_table)) {
   xt_free_table_info(newinfo);
-  return ret;
+  return PTR_ERR(new_table);
  }

  return 0;
--- a/net/netfilter/x_tables.c
+++ b/net/netfilter/x_tables.c
@@ -660,9 +660,9 @@ xt_replace_table(struct xt_table *table,
 }
 EXPORT_SYMBOL_GPL(xt_replace_table);

-int xt_register_table(struct xt_table *table,
-       struct xt_table_info *bootstrap,
-       struct xt_table_info *newinfo)
+struct xt_table *xt_register_table(struct xt_table *table,
+       struct xt_table_info *bootstrap,
+       struct xt_table_info *newinfo)
 {
  int ret;
  struct xt_table_info *private;
@@ -670,7 +670,7 @@ int xt_register_table(struct xt_table *table,

  ret = mutex_lock_interruptible(&xt[table->af].mutex);
  if (ret != 0)
-  return ret;
+  goto out;

  /* Don't autoload: we'd eat our tail... */
  list_for_each_entry(t, &xt[table->af].tables, list) {
@@ -693,11 +693,13 @@ int xt_register_table(struct xt_table *table,
  private->initial_entries = private->number;

  list_add(&table->list, &xt[table->af].tables);
+ mutex_unlock(&xt[table->af].mutex);
+ return table;

- ret = 0;
  unlock:
  mutex_unlock(&xt[table->af].mutex);
- return ret;
```

```
+out:
+ return ERR_PTR(ret);
 }
 EXPORT_SYMBOL_GPL(xt_register_table);
```