
Subject: [PATCH 5/5] netns netfilter: put table module on netns stop
Posted by [Alexey Dobriyan](#) on Thu, 24 Jan 2008 12:30:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

When number of entries exceeds number of initial entries, foo-tables code will pin table module. But during table unregister on netns stop, that additional pin was forgotten.

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

```
net/ipv4/netfilter/arp_tables.c | 3 +++
net/ipv4/netfilter/ip_tables.c | 3 +++
net/ipv6/netfilter/ip6_tables.c | 3 +++
3 files changed, 9 insertions(+)
```

```
--- a/net/ipv4/netfilter/arp_tables.c
+++ b/net/ipv4/netfilter/arp_tables.c
@@ -1773,6 +1773,7 @@ void arpt_unregister_table(struct arpt_table *table)
{
    struct xt_table_info *private;
    void *loc_cpu_entry;
+ struct module *table_owner = table->me;

    private = xt_unregister_table(table);

@@ -1780,6 +1781,8 @@ void arpt_unregister_table(struct arpt_table *table)
    loc_cpu_entry = private->entries[raw_smp_processor_id()];
    ARPT_ENTRY_ITERATE(loc_cpu_entry, private->size,
        cleanup_entry, NULL);
+ if (private->number > private->initial_entries)
+ module_put(table_owner);
    xt_free_table_info(private);
}
```

```
--- a/net/ipv4/netfilter/ip_tables.c
+++ b/net/ipv4/netfilter/ip_tables.c
@@ -2095,12 +2095,15 @@ void ipt_unregister_table(struct xt_table *table)
{
    struct xt_table_info *private;
    void *loc_cpu_entry;
+ struct module *table_owner = table->me;

    private = xt_unregister_table(table);

    /* Decrease module usage counts and free resources */
    loc_cpu_entry = private->entries[raw_smp_processor_id()];
    IPT_ENTRY_ITERATE(loc_cpu_entry, private->size, cleanup_entry, NULL);
```

```

+ if (private->number > private->initial_entries)
+ module_put(table_owner);
  xt_free_table_info(private);
}

--- a/net/ipv6/netfilter/ip6_tables.c
+++ b/net/ipv6/netfilter/ip6_tables.c
@@ -2120,12 +2120,15 @@ void ip6t_unregister_table(struct xt_table *table)
{
  struct xt_table_info *private;
  void *loc_cpu_entry;
+ struct module *table_owner = table->me;

  private = xt_unregister_table(table);

  /* Decrease module usage counts and free resources */
  loc_cpu_entry = private->entries[raw_smp_processor_id()];
  IP6T_ENTRY_ITERATE(loc_cpu_entry, private->size, cleanup_entry, NULL);
+ if (private->number > private->initial_entries)
+ module_put(table_owner);
  xt_free_table_info(private);
}

```
