

Subject: [PATCH net-next 5/9] netns: make rt_secret_rebuild timer per namespace
Posted by [den](#) on Fri, 04 Jul 2008 13:17:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
include/net/netns/ipv4.h | 2 ++
net/ipv4/route.c          | 40 ++++++
2 files changed, 32 insertions(+), 10 deletions(-)
```

```
diff --git a/include/net/netns/ipv4.h b/include/net/netns/ipv4.h
```

```
index a29adf1..356617f 100644
```

```
--- a/include/net/netns/ipv4.h
```

```
+++ b/include/net/netns/ipv4.h
```

```
@@ -46,5 +46,7 @@ struct netns_ipv4 {
    int sysctl_icmp_ratelimit;
    int sysctl_icmp_ratemask;
    int sysctl_icmp_errors_use_inbound_ifaddr;
```

```
+
```

```
+ struct timer_list rt_secret_timer;
```

```
};
```

```
#endif
```

```
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
```

```
index 6fe799d..f99d9db 100644
```

```
--- a/net/ipv4/route.c
```

```
+++ b/net/ipv4/route.c
```

```
@@ -132,7 +132,6 @@ static int ip_rt_secret_interval __read_mostly = 10 * 60 * HZ;
```

```
static void rt_worker_func(struct work_struct *work);
```

```
static DECLARE_DELAYED_WORK(expires_work, rt_worker_func);
```

```
-static struct timer_list rt_secret_timer;
```

```
/*
```

```
 * Interface to generic destination cache.
```

```
@@ -801,10 +800,11 @@ void rt_cache_flush(struct net *net, int delay)
```

```
/*
```

```
 * We change rt_genid and let gc do the cleanup
```

```
*/
```

```
-static void rt_secret_rebuild(unsigned long dummy)
```

```
+static void rt_secret_rebuild(unsigned long __net)
```

```
{
```

```
+ struct net *net = (struct net *)__net;
```

```
    rt_cache_invalidate();
```

```
- mod_timer(&rt_secret_timer, jiffies + ip_rt_secret_interval);
```

```
+ mod_timer(&net->ipv4.rt_secret_timer, jiffies + ip_rt_secret_interval);
```

```
}
```

```
/*
```

```

@@ -3072,6 +3072,31 @@ static __net_initdata struct pernet_operations sysctl_route_ops = {
};
#endif

+
+static __net_init int rt_secret_timer_init(struct net *net)
+{
+ net->ipv4.rt_secret_timer.function = rt_secret_rebuild;
+ net->ipv4.rt_secret_timer.data = (unsigned long)net;
+ init_timer_deferrable(&net->ipv4.rt_secret_timer);
+
+ net->ipv4.rt_secret_timer.expires =
+ jiffies + net_random() % ip_rt_secret_interval +
+ ip_rt_secret_interval;
+ add_timer(&net->ipv4.rt_secret_timer);
+ return 0;
+}
+
+static __net_exit void rt_secret_timer_exit(struct net *net)
+{
+ del_timer_sync(&net->ipv4.rt_secret_timer);
+}
+
+static __net_initdata struct pernet_operations rt_secret_timer_ops = {
+ .init = rt_secret_timer_init,
+ .exit = rt_secret_timer_exit,
+};
+
+
+
+#ifdef CONFIG_NET_CLS_ROUTE
+struct ip_rt_acct *ip_rt_acct __read_mostly;
+#endif /* CONFIG_NET_CLS_ROUTE */
@@ -3124,19 +3149,14 @@ int __init ip_rt_init(void)
devinet_init();
ip_fib_init();

- rt_secret_timer.function = rt_secret_rebuild;
- rt_secret_timer.data = 0;
- init_timer_deferrable(&rt_secret_timer);
-
/* All the timers, started at system startup tend
to synchronize. Perturb it a bit.
*/
schedule_delayed_work(&expires_work,
net_random() % ip_rt_gc_interval + ip_rt_gc_interval);

- rt_secret_timer.expires = jiffies + net_random() % ip_rt_secret_interval +
- ip_rt_secret_interval;

```

```
- add_timer(&rt_secret_timer);
+ if (register_pernet_subsys(&rt_secret_timer_ops))
+ printk(KERN_ERR "Unable to setup rt_secret_timer\n");

if (ip_rt_proc_init())
    printk(KERN_ERR "Unable to create route proc files\n");
--
1.5.3.rc5
```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
