Subject: [PATCH][IPVS] Fix sched registration race when checking for name collision
Posted by Pavel Emelianov on Mon, 03 Dec 2007 10:10:57 GMT
View Forum Message <> Reply to Message

The register_ip_vs_scheduler() checks for the scheduler with the
same name under the read-locked __ip_vs_sched_lock, then drops,
takes it for writing and puts the scheduler in list.

This is racy, since we can have a race window between the lock
being re-locked for writing.

The fix is to search the scheduler with the given name right under
the write-locked __ip_vs_sched_lock.

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

```
diff --git a/net/ipv4/ipvs/ip_vs_sched.c b/net/ipv4/ipvs/ip_vs_sched.c
index 1602304..4322358 100644
--- a/net/ipv4/ipvs/ip_vs_sched.c
+++ b/net/ipv4/ipvs/ip_vs_sched.c
@@ -183,19 +183,6 @@ int register_ip_vs_scheduler(struct ip_vs_scheduler *scheduler)
  /* increase the module use count */
  ip_vs_use_count_inc();

- /*
-  *  Make sure that the scheduler with this name doesn't exist
-  *  in the scheduler list.
-  */
- sched = ip_vs_sched_getbyname(scheduler->name);
- if (sched) {
-  ip_vs_scheduler_put(sched);
-  ip_vs_use_count_dec();
-  IP_VS_ERR("register_ip_vs_scheduler(): [%s] scheduler "
-     "already existed in the system\n", scheduler->name);
-  return -EINVAL;
- }
-
  write_lock_bh(&__ip_vs_sched_lock);

  if (scheduler->n_list.next != &scheduler->n_list) {
@@ -207,6 +194,20 @@ int register_ip_vs_scheduler(struct ip_vs_scheduler *scheduler)
  }

  /*
+  *  Make sure that the scheduler with this name doesn't exist
```

```
+  *  in the scheduler list.
+  */
+ list_for_each_entry(sched, &ip_vs_schedulers, n_list) {
+  if (strcmp(scheduler->name, sched->name) == 0) {
+   write_unlock_bh(&__ip_vs_sched_lock);
+   ip_vs_use_count_dec();
+   IP_VS_ERR("register_ip_vs_scheduler(): [%s] scheduler "
+     "already existed in the system\n",
+     scheduler->name);
+   return -EINVAL;
+  }
+ }
+ /*
   * Add it into the d-linked scheduler list
   */
  list_add(&scheduler->n_list, &ip_vs_schedulers);
```

Subject: Re: [PATCH][IPVS] Fix sched registration race when checking for name collision
Posted by Simon Horman on Tue, 04 Dec 2007 01:41:43 GMT
View Forum Message <> Reply to Message

On Mon, Dec 03, 2007 at 01:10:57PM +0300, Pavel Emelyanov wrote:
> The register_ip_vs_scheduler() checks for the scheduler with the
> same name under the read-locked __ip_vs_sched_lock, then drops,
> takes it for writing and puts the scheduler in list.
>
> This is racy, since we can have a race window between the lock
> being re-locked for writing.
>
> The fix is to search the scheduler with the given name right under
> the write-locked __ip_vs_sched_lock.

This looks correct to me.

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

Acked-by: Simon Horman <horms@verge.net.au>

> ---
>
> diff --git a/net/ipv4/ipvs/ip_vs_sched.c b/net/ipv4/ipvs/ip_vs_sched.c
> index 1602304..4322358 100644
> --- a/net/ipv4/ipvs/ip_vs_sched.c
> +++ b/net/ipv4/ipvs/ip_vs_sched.c
> @@ -183,19 +183,6 @@ int register_ip_vs_scheduler(struct ip_vs_scheduler *scheduler)
>   /* increase the module use count */

```
>   ip_vs_use_count_inc();
>
> - /*
> -  *  Make sure that the scheduler with this name doesn't exist
> -  *  in the scheduler list.
> -  */
> - sched = ip_vs_sched_getbyname(scheduler->name);
> - if (sched) {
> -  ip_vs_scheduler_put(sched);
> -  ip_vs_use_count_dec();
> -  IP_VS_ERR("register_ip_vs_scheduler(): [%s] scheduler "
> -     "already existed in the system\n", scheduler->name);
> -  return -EINVAL;
> - }
> -
>   write_lock_bh(&__ip_vs_sched_lock);
>
>   if (scheduler->n_list.next != &scheduler->n_list) {
> @@ -207,6 +194,20 @@ int register_ip_vs_scheduler(struct ip_vs_scheduler *scheduler)
>   }
>
>   /*
> +  *  Make sure that the scheduler with this name doesn't exist
> +  *  in the scheduler list.
> +  */
> + list_for_each_entry(sched, &ip_vs_schedulers, n_list) {
> +  if (strcmp(scheduler->name, sched->name) == 0) {
> +   write_unlock_bh(&__ip_vs_sched_lock);
> +   ip_vs_use_count_dec();
> +   IP_VS_ERR("register_ip_vs_scheduler(): [%s] scheduler "
> +     "already existed in the system\n",
> +     scheduler->name);
> +   return -EINVAL;
> +  }
> + }
> + /*
>    * Add it into the d-linked scheduler list
>    */
>   list_add(&scheduler->n_list, &ip_vs_schedulers);

--
Horms
```

Subject: Re: [PATCH][IPVS] Fix sched registration race when checking for name collision
Posted by davem on Tue, 04 Dec 2007 08:45:25 GMT

From: Simon Horman <horms@verge.net.au>
Date: Tue, 4 Dec 2007 10:41:43 +0900

> On Mon, Dec 03, 2007 at 01:10:57PM +0300, Pavel Emelyanov wrote:
> > The register_ip_vs_scheduler() checks for the scheduler with the
> > same name under the read-locked __ip_vs_sched_lock, then drops,
> > takes it for writing and puts the scheduler in list.
> >
> > This is racy, since we can have a race window between the lock
> > being re-locked for writing.
> >
> > The fix is to search the scheduler with the given name right under
> > the write-locked __ip_vs_sched_lock.
>
> This looks correct to me.
>
> > Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
>
> Acked-by: Simon Horman <horms@verge.net.au>

Also applied, thanks a lot.