



index : kernel/git/stable/linux.git

master



switch

Linux kernel stable tree

Stable Group

[about](#) [summary](#) [refs](#) [log](#) [tree](#) [commit](#) [diff](#) [stats](#)

log msg



search

author Zhengchao Shao <shaozhengchao@huawei.com> 2022-10-31 20:07:04 +0800
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2022-11-10 17:46:52 +0100
commit [f08ee2aa24c076f81d84e26e213d8c6f4efd9f50](#) (patch)
tree [fb40e774b47d0598eddf57a0500336c68252a446](#)
parent [a969e4473f0f71631a3cceb182acff05fa06f773](#) (diff)
download [linux-f08ee2aa24c076f81d84e26e213d8c6f4efd9f50.tar.gz](#)

diff options

context:

3



space:

include



mode:

unified



ipvs: fix WARNING in __ip_vs_cleanup_batch()

[Upstream commit 3d00c6a0da8ddcf75213e004765e4a42acc71d5d]

During the initialization of ip_vs_conn_net_init(), if file ip_vs_conn or ip_vs_conn_sync fails to be created, the initialization is successful by default. Therefore, the ip_vs_conn or ip_vs_conn_sync file doesn't be found during the remove.

The following is the stack information:

```
name 'ip_vs_conn_sync'
WARNING: CPU: 3 PID: 9 at fs/proc/generic.c:712
remove_proc_entry+0x389/0x460
Modules linked in:
Workqueue: netns cleanup_net
RIP: 0010:remove_proc_entry+0x389/0x460
Call Trace:
<TASK>
__ip_vs_cleanup_batch+0x7d/0x120
ops_exit_list+0x125/0x170
cleanup_net+0x4ea/0xb00
process_one_work+0x9bf/0x1710
worker_thread+0x665/0x1080
kthread+0x2e4/0x3a0
ret_from_fork+0x1f/0x30
</TASK>
```

Fixes: 61b1ab4583e2 ("IPVS: netns, add basic init per netns.")

Signed-off-by: Zhengchao Shao <shaozhengchao@huawei.com>

Acked-by: Julian Anastasov <ja@ssi.bg>

Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>

Signed-off-by: Sasha Levin <sasha@kernel.org>

Diffstat

```
-rw-r--r-- net/netfilter/ipvs/ip_vs_conn.c 26
```

1 files changed, 21 insertions, 5 deletions

diff --git a/net/netfilter/ipvs/ip_vs_conn.c b/net/netfilter/ipvs/ip_vs_conn.c

index 5ebc7998529a58..51679d1e2d7d72 100644

--- a/net/netfilter/ipvs/ip_vs_conn.c

+++ b/net/netfilter/ipvs/ip_vs_conn.c

```
@@ -1378,20 +1378,36 @@ int __net_init ip_vs_conn_net_init(struct netns_ipvs *ipvs)
{
```

```

        atomic_set(&ipvs->conn_count, 0);

-       proc_create_net("ip_vs_conn", 0, ipvs->net->proc_net,
-                       &ip_vs_conn_seq_ops, sizeof(struct ip_vs_iter_state));
-       proc_create_net("ip_vs_conn_sync", 0, ipvs->net->proc_net,
-                       &ip_vs_conn_sync_seq_ops,
-                       sizeof(struct ip_vs_iter_state));
+
+#ifdef CONFIG_PROC_FS
+       if (!proc_create_net("ip_vs_conn", 0, ipvs->net->proc_net,
+                           &ip_vs_conn_seq_ops,
+                           sizeof(struct ip_vs_iter_state)))
+           goto err_conn;
+
+       if (!proc_create_net("ip_vs_conn_sync", 0, ipvs->net->proc_net,
+                           &ip_vs_conn_sync_seq_ops,
+                           sizeof(struct ip_vs_iter_state)))
+           goto err_conn_sync;
+#endif
+
+       return 0;
+
+#ifdef CONFIG_PROC_FS
+err_conn_sync:
+       remove_proc_entry("ip_vs_conn", ipvs->net->proc_net);
+err_conn:
+       return -ENOMEM;
+#endif
    }

void __net_exit ip_vs_conn_net_cleanup(struct netns_ipvs *ipvs)
{
    /* flush all the connection entries first */
    ip_vs_conn_flush(ipvs);
+
+#ifdef CONFIG_PROC_FS
+       remove_proc_entry("ip_vs_conn", ipvs->net->proc_net);
+       remove_proc_entry("ip_vs_conn_sync", ipvs->net->proc_net);
+#endif
    }

int __init ip_vs_conn_init(void)

```