



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 20:07:05 +0800
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2022-11-10 18:17:19 +0100
commit [97f872b00937f2689bffa2dab4ad9ed259482840f](#) (patch)
tree [d2cbf5a2a6e1d9f9b4408e4f12be5a62eab2844c](#)
parent [e724220b826e008764309d2a1f55a9434a4e1530](#) (diff)
download [linux-97f872b00937f2689bffa2dab4ad9ed259482840f.tar.gz](#)

diff options

context:

3

space:

include

mode:

unified

ipvs: fix WARNING in ip_vs_app_net_cleanup()

[Upstream commit 5663ed63adb9619c98ab7479aa4606fa9b7a548c]

During the initialization of ip_vs_app_net_init(), if file ip_vs_app fails to be created, the initialization is successful by default. Therefore, the ip_vs_app file doesn't be found during the remove in ip_vs_app_net_cleanup(). It will cause WRNING.

The following is the stack information:

name 'ip_vs_app'

WARNING: CPU: 1 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>

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: 457c4cbc5a3d ("[NET]: Make /proc/net per network namespace")

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 <sashal@kernel.org>

Diffstat

-rw-r--r-- [net/netfilter/ipvs/ip_vs_app.c](#) 10

1 files changed, 8 insertions, 2 deletions

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

index f9b16f2b221918..fdacbc3c15bef9 100644

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

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

@@ -599,13 +599,19 @@ static const struct seq_operations ip_vs_app_seq_ops = {

int __net_init ip_vs_app_net_init(struct netns_ipvs *ipvs)

{

INIT_LIST_HEAD(&ipvs->app_list);

```
-      proc_create_net("ip_vs_app", 0, ipvs->net->proc_net, &ip_vs_app_seq_ops,  
-                      sizeof(struct seq_net_private));  
  
+#ifdef CONFIG_PROC_FS  
+    if (!proc_create_net("ip_vs_app", 0, ipvs->net->proc_net,  
+                          &ip_vs_app_seq_ops,  
+                          sizeof(struct seq_net_private)))  
+        return -ENOMEM;  
+#endif  
  
    return 0;  
}  
  
void __net_exit ip_vs_app_net_cleanup(struct netns_ipvs *ipvs)  
{  
    unregister_ip_vs_app(ipvs, NULL /* all */);  
+#ifdef CONFIG_PROC_FS  
    remove_proc_entry("ip_vs_app", ipvs->net->proc_net);  
+#endif  
}
```