



index : kernel/git/stable/linux.git

master

Linux kernel stable tree

Stable Group

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

log msg

author Abdun Nihaal <abdun.nihaal@gmail.com> 2025-04-14 22:36:46 +0530
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2025-04-25 10:47:45 +0200
commit [dafb6e433ab2333b67be05433dc9c6ccbc7b1284](#) (patch)
tree [24a14d7fe35d9a560cf230cb321fc6f5bfd40ae5](#)
parent [cb74d5b83714b5cf0caf53eaa31b6a352d3d3495](#) (diff)
download [linux-dafb6e433ab2333b67be05433dc9c6ccbc7b1284.tar.gz](#)

diff options

context:
space:
mode:

cxgb4: fix memory leak in cxgb4_init_ethtool_filters() error path

[Upstream commit 00ffb3724ce743578163f5ade2884374554ca021]

In the for loop used to allocate the loc_array and bmap for each port, a memory leak is possible when the allocation for loc_array succeeds, but the allocation for bmap fails. This is because when the control flow goes to the label free_eth_finfo, only the allocations starting from (i-1)th iteration are freed.

Fix that by freeing the loc_array in the bmap allocation error path.

Fixes: d915c299f1da ("cxgb4: add skeleton for ethtool n-tuple filters")
Signed-off-by: Abdun Nihaal <abdun.nihaal@gmail.com>
Reviewed-by: Simon Horman <horms@kernel.org>
Reviewed-by: Jacob Keller <jacob.e.keller@intel.com>
Link: <https://patch.msgid.link/20250414170649.89156-1-abdun.nihaal@gmail.com>
Signed-off-by: Jakub Kicinski <kuba@kernel.org>
Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

-rw-r--r-- drivers/net/ethernet/chelsio/cxgb4/cxgb4_ethtool.c 1

1 files changed, 1 insertions, 0 deletions

```
diff --git a/drivers/net/ethernet/chelsio/cxgb4/cxgb4_ethtool.c b/drivers/net/ethernet/chelsio/cxgb4/cxgb4_ethtool.c
index 7f3f5afa864f4a..1546c3db08f093 100644
--- a/drivers/net/ethernet/chelsio/cxgb4/cxgb4_ethtool.c
+++ b/drivers/net/ethernet/chelsio/cxgb4/cxgb4_ethtool.c
@@ -2270,6 +2270,7 @@ int cxgb4_init_ethtool_filters(struct adapter *adap)
     eth_filter->port[i].bmap = bitmap_zalloc(nentries, GFP_KERNEL);
     if (!eth_filter->port[i].bmap) {
         ret = -ENOMEM;
+        kvfree(eth_filter->port[i].loc_array);
         goto free_eth_finfo;
     }
 }
```