



# 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 Chuang Wang <nashuiliang@gmail.com> 2022-11-09 17:07:34 +0800  
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2022-11-16 09:57:14 +0100  
commit [956e0216a19994443c90ba2ea6b0b284c9c4f9cb](#) (patch)  
tree [270c6d9eaba0ad63eab34fa4d9eafe1b9f376e5a](#)  
parent [59ec132386a07ac09165f96226fe6fcb44b54b37](#) (diff)  
download [linux-956e0216a19994443c90ba2ea6b0b284c9c4f9cb.tar.gz](#)

## diff options

context:    
space:    
mode:

## net: macvlan: fix memory leaks of macvlan\_common\_newlink

[ Upstream commit 23569b5652ee8e8e55a12f7835f59af6f3cefc30 ]

kmemleak reports memory leaks in macvlan\_common\_newlink, as follows:

```
ip link add link eth0 name .. type macvlan mode source macaddr add
<MAC-ADDR>
```

kmemleak reports:

```
unreferenced object 0xfffff8880109bb140 (size 64):
  comm "ip", pid 284, jiffies 4294986150 (age 430.108s)
  hex dump (first 32 bytes):
    00 00 00 00 00 00 00 00 b8 aa 5a 12 80 88 ff ff  ....Z.....
    80 1b fa 0d 80 88 ff ff 1e ff ac af c7 c1 6b 6b  ....kk
  backtrace:
    [<ffffffffff813e06a7>] kmem_cache_alloc_trace+0x1c7/0x300
    [<ffffffffff81b66025>] macvlan_hash_add_source+0x45/0xc0
    [<ffffffffff81b66a67>] macvlan_changelink_sources+0xd7/0x170
    [<ffffffffff81b6775c>] macvlan_common_newlink+0x38c/0x5a0
    [<ffffffffff81b6797e>] macvlan_newlink+0xe/0x20
    [<ffffffffff81d97f8f>] __rtnl_newlink+0x7af/0xa50
    [<ffffffffff81d98278>] rtnl_newlink+0x48/0x70
    ...
```

In the scenario where the macvlan mode is configured as 'source', macvlan\_changelink\_sources() will be executed to reconfigure list of remote source mac addresses, at the same time, if register\_netdevice() return an error, the resource generated by macvlan\_changelink\_sources() is not cleaned up.

Using this patch, in the case of an error, it will execute macvlan\_flush\_sources() to ensure that the resource is cleaned up.

Fixes: [aa5fd0fb7748](#) ("driver: macvlan: Destroy new macvlan port if macvlan\_common\_newlink failed.")  
Signed-off-by: Chuang Wang <nashuiliang@gmail.com>  
Link: <https://lore.kernel.org/r/20221109090735.690500-1-nashuiliang@gmail.com>  
Signed-off-by: Paolo Abeni <pabeni@redhat.com>  
Signed-off-by: Sasha Levin <sashal@kernel.org>

## Diffstat

-rw-r--r-- [drivers/net/macvlan.c](#) 4

1 files changed, 3 insertions, 1 deletions

```
diff --git a/drivers/net/macvlan.c b/drivers/net/macvlan.c
index c8d803d3616c90..6b269a72388b8d 100644
--- a/drivers/net/macvlan.c
+++ b/drivers/net/macvlan.c
@@ -1509,8 +1509,10 @@ destroy_macvlan_port:
    /* the macvlan port may be freed by macvlan_uninit when fail to register.
     * so we destroy the macvlan port only when it's valid.
     */
-   if (create && macvlan_port_get_rtnl(lowerdev))
+   if (create && macvlan_port_get_rtnl(lowerdev)) {
+       macvlan_flush_sources(port, vlan);
+       macvlan_port_destroy(port->dev);
+   }
    return err;
}
EXPORT_SYMBOL_GPL(macvlan_common_newlink);
```