author	Gang Yan <yangang@kylinos.cn>	2025-03-28 15:27:16 +0100
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2025-04-20 10:18:12 +0200
commit	efd58a8dd9e7a709a90ee486a4247c923d27296f (patch)	
tree	d9e3f49e9b613fc34b1a61cb7c99c24e7cd3f7c7	
parent	6a95a782507323e563ac6655dd217f025bc9f7cb (diff)	
download	linux-efd58a8dd9e7a709a90ee486a4247c923d27296f.tar.gz	

mptcp: fix NULL pointer in can_accept_new_subflow

```
commit 443041deb5ef6a1289a99ed95015ec7442f141dc upstream.

When testing valkey benchmark tool with MPTCP, the kernel panics in
'mptcp_can_accept_new_subflow' because subflow_req->msk is NULL.

Call trace:

  mptcp_can_accept_new_subflow (./net/mptcp/subflow.c:63 (discriminator 4)) (P)
  subflow_syn_recv_sock (./net/mptcp/subflow.c:854)
  tcp_check_req (./net/ipv4/tcp_minisocks.c:863)
  tcp_v4_rcv (./net/ipv4/tcp_ipv4.c:2268)
  ip_protocol_deliver_rcu (./net/ipv4/ip_input.c:207)
  ip_local_deliver_finish (./net/ipv4/ip_input.c:234)
  ip_local_deliver (./net/ipv4/ip_input.c:254)
  ip_rcv_finish (./net/ipv4/ip_input.c:449)
  ...

According to the debug log, the same req received two SYN-ACK in a very
short time, very likely because the client retransmits the syn ack due
to multiple reasons.

Even if the packets are transmitted with a relevant time interval, they
can be processed by the server on different CPUs concurrently). The
'subflow_req->msk' ownership is transferred to the subflow the first,
and there will be a risk of a null pointer dereference here.

This patch fixes this issue by moving the 'subflow_req->msk' under the
`own_req == true` conditional.

Note that the !msk check in subflow_hmac_valid() can be dropped, because
the same check already exists under the own_req mpj branch where the
code has been moved to.

Fixes: 9466a1ccebbe ("mptcp: enable JOIN requests even if cookies are in use")
Cc: stable@vger.kernel.org
Suggested-by: Paolo Abeni <pabeni@redhat.com>
Signed-off-by: Gang Yan <yangang@kylinos.cn>
Reviewed-by: Matthieu Baerts (NGI0) <matttbe@kernel.org>
Signed-off-by: Matthieu Baerts (NGI0) <matttbe@kernel.org>
Link: https://patch.msgid.link/20250328-net-mptcp-misc-fixes-6-15-v1-1-34161a482a7f@kernel.org
```

Signed-off-by: Jakub Kicinski <kuba@kernel.org>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

Diffstat

-rw-r--r-- net/mptcp/subflow.c 15

1 files changed, 8 insertions, 7 deletions

```
diff --git a/net/mptcp/subflow.c b/net/mptcp/subflow.c
index 9f18217dddc865..a3d3fbb78075b8 100644
--- a/net/mptcp/subflow.c
+++ b/net/mptcp/subflow.c
@@ -754,8 +754,6 @@ static bool subflow_hmac_valid(const struct request_sock *req,

        subflow_req = mptcp_subflow_rsk(req);
        msk = subflow_req->msk;
-       if (!msk)
-               return false;

        subflow_generate_hmac(READ_ONCE(msk->remote_key),
                              READ_ONCE(msk->local_key),
@@ -853,12 +851,8 @@ static struct sock *subflow_syn_recv_sock(const struct sock *sk,

        } else if (subflow_req->mp_join) {
                mptcp_get_options(skb, &mp_opt);
-               if (!(mp_opt.suboptions & OPTION_MPTCP_MPJ_ACK) ||
-                   !subflow_hmac_valid(req, &mp_opt) ||
-                   !mptcp_can_accept_new_subflow(subflow_req->msk)) {
-                       SUBFLOW_REQ_INC_STATS(req, MPTCP_MIB_JOINACKMAC);
+               if (!(mp_opt.suboptions & OPTION_MPTCP_MPJ_ACK))
                        fallback = true;
-               }
        }

 create_child:
@@ -908,6 +902,13 @@ create_child:
                                goto dispose_child;
                        }

+                       if (!subflow_hmac_valid(req, &mp_opt) ||
+                           !mptcp_can_accept_new_subflow(subflow_req->msk)) {
+                               SUBFLOW_REQ_INC_STATS(req, MPTCP_MIB_JOINACKMAC);
+                               subflow_add_reset_reason(skb, MPTCP_RST_EPROHIBIT);
+                               goto dispose_child;
+                       }
+
                        /* move the msk reference ownership to the subflow */
                        subflow_req->msk = NULL;
                        ctx->conn = (struct sock *)owner;
```