



author Wang Yufen <wangyufen@huawei.com> 2022-11-01 09:31:36 +0800
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2022-11-16 10:03:51 +0100
commit [14e8bc3bf7bd6af64d7538a0684c8238d96cdfd7](#) (patch)
tree [c9d971196d59232135d5eb4c1928f7458b188fac](#)
parent [3e210891c4a4c2d858cd6f9f61d5809af251d4df](#) (diff)
download [linux-14e8bc3bf7bd6af64d7538a0684c8238d96cdfd7.tar.gz](#)

diff options

context: ▼
space: ▼
mode: ▼

bpf, sockmap: Fix the sk->sk_forward_alloc warning of sk_stream_kill_queues

[Upstream commit 8ec95b94716a1e4d126edc3fb2bc426a717e2dba]

When running `test\_sockmap` selftests, the following warning appears:

```
WARNING: CPU: 2 PID: 197 at net/core/stream.c:205 sk_stream_kill_queues+0xd3/0xf0
Call Trace:
<TASK>
inet_csk_destroy_sock+0x55/0x110
tcp_rcv_state_process+0xd28/0x1380
? tcp_v4_do_rcv+0x77/0x2c0
tcp_v4_do_rcv+0x77/0x2c0
__release_sock+0x106/0x130
__tcp_close+0x1a7/0x4e0
tcp_close+0x20/0x70
inet_release+0x3c/0x80
__sock_release+0x3a/0xb0
sock_close+0x14/0x20
__fput+0xa3/0x260
task_work_run+0x59/0xb0
exit_to_user_mode_prepare+0x1b3/0x1c0
syscall_exit_to_user_mode+0x19/0x50
do_syscall_64+0x48/0x90
entry_SYSCALL_64_after_hwframe+0x44/0xae
```

The root case is in commit 84472b436e76 ("bpf, sockmap: Fix more uncharged while msg has more_data"), where I used msg->sg.size to replace the tosend, causing breakage:

```
if (msg->apply_bytes && msg->apply_bytes < tosend)
    tosend = psock->apply_bytes;
```

Fixes: 84472b436e76 ("bpf, sockmap: Fix more uncharged while msg has more_data")

Reported-by: Jakub Sitnicki <jakub@cloudflare.com>

Signed-off-by: Wang Yufen <wangyufen@huawei.com>

Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>

Acked-by: John Fastabend <john.fastabend@gmail.com>

Acked-by: Jakub Sitnicki <jakub@cloudflare.com>

Link: <https://lore.kernel.org/bpf/1667266296-8794-1-git-send-email-wangyufen@huawei.com>

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

1 files changed, 5 insertions, 3 deletions

diff --git a/net/ipv4/tcp_bpf.c b/net/ipv4/tcp_bpf.c

index c501c329b1dbe4..cf9c3e8f7ccbf9 100644

--- a/net/ipv4/tcp_bpf.c

+++ b/net/ipv4/tcp_bpf.c

@@ -278,7 +278,7 @@ static int tcp_bpf_send_verdict(struct sock *sk, struct sk_psock *psock,
{

bool cork = false, enospc = sk_msg_full(msg);

struct sock *sk_redir;

- u32 tosend, delta = 0;

+ u32 tosend, origsize, sent, delta = 0;

u32 eval = __SK_NONE;

int ret;

@@ -333,10 +333,12 @@ more_data:

cork = true;

psock->cork = NULL;

}

- sk_msg_return(sk, msg, msg->sg.size);

+ sk_msg_return(sk, msg, tosend);

release_sock(sk);

+ origsize = msg->sg.size;

ret = tcp_bpf_sendmsg_redir(sk_redir, msg, tosend, flags);

+ sent = origsize - msg->sg.size;

if (eval == __SK_REDIRECT)

sock_put(sk_redir);

@@ -375,7 +377,7 @@ more_data:

msg->sg.data[msg->sg.start].page_link &&

msg->sg.data[msg->sg.start].length) {

if (eval == __SK_REDIRECT)

- sk_mem_charge(sk, msg->sg.size);

+ sk_mem_charge(sk, tosend - sent);

goto more_data;

}

}