author	Eric Dumazet <edumazet@google.com>	2023-03-20 16:34:27 +0000
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2023-03-30 12:47:48 +0200
commit	b41f37dbd9cdb60000e3b0dfad6df787591c2265 (patch)	
tree	fc3423b9a728951de82bd1adcdc3f19911de2503	
parent	4a8286baf22e5413747064f648462e5593fd353b (diff)	
download	linux-b41f37dbd9cdb60000e3b0dfad6df787591c2265.tar.gz	

diff options

context: 3

space: include

mode: unified

erspan: do not use skb_mac_header() in ndo_start_xmit()

```
[ Upstream commit 8e50ed774554f93d55426039b27b1e38d7fa64d8 ]

Drivers should not assume skb_mac_header(skb) == skb->data in their
ndo_start_xmit().

Use skb_network_offset() and skb_transport_offset() which
better describe what is needed in erspan_fb_xmit() and
ip6erspan_tunnel_xmit()

syzbot reported:
WARNING: CPU: 0 PID: 5083 at include/linux/skbuff.h:2873 skb_mac_header include/linux/skbuff.h:2873 [inline]
WARNING: CPU: 0 PID: 5083 at include/linux/skbuff.h:2873 ip6erspan_tunnel_xmit+0x1d9c/0x2d90 net/ipv6/ip6_gre.c:962
Modules linked in:
CPU: 0 PID: 5083 Comm: syz-executor406 Not tainted 6.3.0-rc2-syzkaller-00866-gd4671cb96fa3 #0
Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 03/02/2023
RIP: 0010:skb_mac_header include/linux/skbuff.h:2873 [inline]
RIP: 0010:ip6erspan_tunnel_xmit+0x1d9c/0x2d90 net/ipv6/ip6_gre.c:962
Code: 04 02 41 01 de 84 c0 74 08 3c 03 0f 8e 1c 0a 00 00 45 89 b4 24 c8 00 00 00 c6 85 77 fe ff ff 01 e9 33 e7 ff ff e8 b4 27 a1 f8 <0f> 0b e9 b6 e7
RSP: 0018:ffffc90003b2f830 EFLAGS: 00010293
RAX: 0000000000000000 RBX: 000000000000ffff RCX: 0000000000000000
RDX: ffff888021273a80 RSI: ffffffff88e1bd4c RDI: 0000000000000003
RBP: ffffc90003b2f9d8 R08: 0000000000000003 R09: 000000000000ffff
R10: 000000000000ffff R11: 0000000000000000 R12: ffff88802b28da00
R13: 00000000000000d0 R14: ffff88807e25b6d0 R15: ffff888023408000
FS: 0000555556a61300(0000) GS:ffff8880b9800000(0000) knlGS:0000000000000000
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 000055e5b11eb6e8 CR3: 0000000027c1b000 CR4: 00000000003506f0
DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
Call Trace:
<TASK>
__netdev_start_xmit include/linux/netdevice.h:4900 [inline]
netdev_start_xmit include/linux/netdevice.h:4914 [inline]
__dev_direct_xmit+0x504/0x730 net/core/dev.c:4300
dev_direct_xmit include/linux/netdevice.h:3088 [inline]
packet_xmit+0x20a/0x390 net/packet/af_packet.c:285
packet_snd net/packet/af_packet.c:3075 [inline]
packet_sendmsg+0x31a0/0x5150 net/packet/af_packet.c:3107
sock_sendmsg_nosec net/socket.c:724 [inline]
sock_sendmsg+0xde/0x190 net/socket.c:747
__sys_sendto+0x23a/0x340 net/socket.c:2142
__do_sys_sendto net/socket.c:2154 [inline]
__se_sys_sendto net/socket.c:2150 [inline]
__x64_sys_sendto+0xe1/0x1b0 net/socket.c:2150
do_syscall_x64 arch/x86/entry/common.c:50 [inline]
do_syscall_64+0x39/0xb0 arch/x86/entry/common.c:80
entry_SYSCALL_64_after_hwframe+0x63/0xcd
RIP: 0033:0x7f123aaa1039
Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 b1 14 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff
RSP: 002b:00007ffc15d12058 EFLAGS: 00000246 ORIG_RAX: 000000000000002c
RAX: ffffffffffffffda RBX: 0000000000000000 RCX: 00007f123aaa1039
RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000003
RBP: 0000000000000000 R08: 0000000020000040 R09: 0000000000000014
R10: 0000000000000000 R11: 0000000000000246 R12: 00007f123aa648c0
R13: 431bde82d7b634db R14: 0000000000000000 R15: 0000000000000000

Fixes: 1baf5ebf8954 ("erspan: auto detect truncated packets.")
Reported-by: syzbot <syzkaller@googlegroups.com>
Signed-off-by: Eric Dumazet <edumazet@google.com>
Reviewed-by: Simon Horman <simon.horman@corigine.com>
Link: https://lore.kernel.org/r/20230320163427.8096-1-edumazet@google.com
Signed-off-by: Jakub Kicinski <kuba@kernel.org>
Signed-off-by: Sasha Levin <sashal@kernel.org>
```

Diffstat

-rw-r--r-- net/ipv4/ip_gre.c	4
-rw-r--r-- net/ipv6/ip6_gre.c	4

2 files changed, 4 insertions, 4 deletions

```
diff --git a/net/ipv4/ip_gre.c b/net/ipv4/ip_gre.c
index 454c4357a29797..c094963a86f1e0 100644
```

```
--- a/net/ipv4/ip_gre.c
+++ b/net/ipv4/ip_gre.c
@@ -552,7 +552,7 @@ static void erspan_fb_xmit(struct sk_buff *skb, struct net_device *dev)
                truncate = true;
        }

-       nhoff = skb_network_header(skb) - skb_mac_header(skb);
+       nhoff = skb_network_offset(skb);
        if (skb->protocol == htons(ETH_P_IP) &&
            (ntohs(ip_hdr(skb)->tot_len) > skb->len - nhoff))
                truncate = true;
@@ -561,7 +561,7 @@ static void erspan_fb_xmit(struct sk_buff *skb, struct net_device *dev)
                int thoff;

                if (skb_transport_header_was_set(skb))
-                       thoff = skb_transport_header(skb) - skb_mac_header(skb);
+                       thoff = skb_transport_offset(skb);
                else
                        thoff = nhoff + sizeof(struct ipv6hdr);
                if (ntohs(ipv6_hdr(skb)->payload_len) > skb->len - thoff)

diff --git a/net/ipv6/ip6_gre.c b/net/ipv6/ip6_gre.c
index 13b1748b8b4659..a91f93ec7d2b4e 100644
--- a/net/ipv6/ip6_gre.c
+++ b/net/ipv6/ip6_gre.c
@@ -959,7 +959,7 @@ static netdev_tx_t ip6erspan_tunnel_xmit(struct sk_buff *skb,
                truncate = true;
        }

-       nhoff = skb_network_header(skb) - skb_mac_header(skb);
+       nhoff = skb_network_offset(skb);
        if (skb->protocol == htons(ETH_P_IP) &&
            (ntohs(ip_hdr(skb)->tot_len) > skb->len - nhoff))
                truncate = true;
@@ -968,7 +968,7 @@ static netdev_tx_t ip6erspan_tunnel_xmit(struct sk_buff *skb,
                int thoff;

                if (skb_transport_header_was_set(skb))
-                       thoff = skb_transport_header(skb) - skb_mac_header(skb);
+                       thoff = skb_transport_offset(skb);
                else
                        thoff = nhoff + sizeof(struct ipv6hdr);
                if (ntohs(ipv6_hdr(skb)->payload_len) > skb->len - thoff)
```