

author Eric Dumazet <edumazet@google.com> 2022-06-20 01:35:06 -0700
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2022-06-29 09:04:32 +0200
commit 2c8aeffc7c586d53e1d380f010bdca4f710f2480 (patch)
tree 96f4d7c5847cc8a3b66a7b131c911800c10b15e4
parent 0d79a2d5816df5059d17e2fcee128c153e2abd63 (diff)
download linux-2c8aeffc7c586d53e1d380f010bdca4f710f2480.tar.gz

diff options

context: 3

space: include

mode: unified

erspan: do not assume transport header is always set

[Upstream commit 301bd140ed0b24f0da660874c7e8a47dad8c8222]

Rewrite tests in ip6erspan_tunnel_xmit() and
erspan_fb_xmit() to not assume transport header is set.

syzbot reported:

WARNING: CPU: 0 PID: 1350 at include/linux/skbuff.h:2911 skb_transport_header include/linux/skbuff.h:2911 [inline]
WARNING: CPU: 0 PID: 1350 at include/linux/skbuff.h:2911 ip6erspan_tunnel_xmit+0x15af/0x2eb0 net/ipv6/ip6_gre.c:963
Modules linked in:
CPU: 0 PID: 1350 Comm: aoe_tx0 Not tainted 5.19.0-rc2-syzkaller-00160-g274295c6e53f #0
Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.14.0-2 04/01/2014
RIP: 0010:skb_transport_header include/linux/skbuff.h:2911 [inline]
RIP: 0010:ip6erspan_tunnel_xmit+0x15af/0x2eb0 net/ipv6/ip6_gre.c:963
Code: 0f 47 f0 40 88 b5 7f fe ff ff e8 8c 16 4b f9 89 de bf ff ff ff ff e8 a0 12 4b f9 66 83 fb ff 0f 85 1d f1 ff ff e8 71 16 4b f9 <0f> 0b e9 43 f0
RSP: 0018:ffffc90005daf910 EFLAGS: 00010293
RAX: 0000000000000000 RBX: 000000000000ffff RCX: 0000000000000000
RDX: ffff88801f032100 RSI: ffffffff882e8d3f RDI: 0000000000000003
RBP: fffffc90005dafab8 R08: 0000000000000003 R09: 000000000000ffff
R10: 000000000000ffff R11: 0000000000000000 R12: ffff888024f21d40
R13: 0000000000000a288 R14: 00000000000000b0 R15: ffff888025a2e000
FS: 0000000000000000(0000) GS:ffff88802c800000(0000) knlGS:0000000000000000
CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 0000001b2e425000 CR3: 000000006d099000 CR4: 0000000000152ef0
Call Trace:
<TASK>
__netdev_start_xmit include/linux/netdevice.h:4805 [inline]
netdev_start_xmit include/linux/netdevice.h:4819 [inline]
xmit_one net/core/dev.c:3588 [inline]
dev_hard_start_xmit+0x188/0x880 net/core/dev.c:3604
sch_direct_xmit+0x19f/0xbe0 net/sched/sch_generic.c:342
__dev_xmit_skb net/core/dev.c:3815 [inline]
__dev_queue_xmit+0x14a1/0x3900 net/core/dev.c:4219
dev_queue_xmit include/linux/netdevice.h:2994 [inline]
tx+0x6a/0xc0 drivers/block/aoe/aoenet.c:63
kthread+0x1e7/0x3b0 drivers/block/aoe/aoecmd.c:1229
kthread+0x2e9/0x3a0 kernel/kthread.c:376
ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:302
</TASK>

Fixes: d5db21a3e697 ("erspan: auto detect truncated ipv6 packets.")
Reported-by: syzbot <syzkaller@googlegroups.com>
Signed-off-by: Eric Dumazet <edumazet@google.com>
Cc: William Tu <u9012063@gmail.com>
Signed-off-by: David S. Miller <davem@davemloft.net>
Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

-rw-r--r--	net/ipv4/ip_gre.c	15
-rw-r--r--	net/ipv6/ip6_gre.c	15

2 files changed, 20 insertions, 10 deletions

diff --git a/net/ipv4/ip_gre.c b/net/ipv4/ip_gre.c
index bc8dfdf1c48ad5..31867351797669 100644
--- a/net/ipv4/ip_gre.c
+++ b/net/ipv4/ip_gre.c
@@ -524,7 +524,6 @@ static void erspan_fb_xmit(struct sk_buff *skb, struct net_device *dev)
int tunnel_hlen;
int version;
int nhoff;
- int thoff;

tun_info = skb_tunnel_info(skb);
if (unlikely(!tun_info || !(tun_info->mode & IP_TUNNEL_INFO_TX) ||
@@ -558,10 +557,16 @@ static void erspan_fb_xmit(struct sk_buff *skb, struct net_device *dev)
(ntohs(ip_hdr(skb)->tot_len) > skb->len - nhoff))
truncate = true;

- thoff = skb_transport_header(skb) - skb_mac_header(skb);
- if (skb->protocol == htons(ETH_P_IPV6) &&
- (ntohs(ipv6_hdr(skb)->payload_len) > skb->len - thoff))
- truncate = true;

```

+         if (skb->protocol == htons(ETH_P_IPV6)) {
+             int thoff;
+
+             if (skb_transport_header_was_set(skb))
+                 thoff = skb_transport_header(skb) - skb_mac_header(skb);
+             else
+                 thoff = nhoff + sizeof(struct ipv6hdr);
+             if (ntohs(ipv6_hdr(skb)->payload_len) > skb->len - thoff)
+                 truncate = true;
+         }
+
+         if (version == 1) {
+             erspan_build_header(skb, ntohl(tunnel_id_to_key32(key->tun_id)),
+
diff --git a/net/ipv6/ip6_gre.c b/net/ipv6/ip6_gre.c
index 5136959b3dc5d6..b996ccaff56e3f 100644
--- a/net/ipv6/ip6_gre.c
+++ b/net/ipv6/ip6_gre.c
@@ -944,7 +944,6 @@ static netdev_tx_t ip6erspan_tunnel_xmit(struct sk_buff *skb,
     __be16 proto;
     __u32 mtu;
     int nhoff;
-    int thoff;

     if (!pskb_inet_may_pull(skb))
         goto tx_err;
@@ -965,10 +964,16 @@ static netdev_tx_t ip6erspan_tunnel_xmit(struct sk_buff *skb,
     (ntohs(ip_hdr(skb)->tot_len) > skb->len - nhoff))
         truncate = true;

-    thoff = skb_transport_header(skb) - skb_mac_header(skb);
-    if (skb->protocol == htons(ETH_P_IPV6) &&
-        (ntohs(ipv6_hdr(skb)->payload_len) > skb->len - thoff))
-        truncate = true;
+    if (skb->protocol == htons(ETH_P_IPV6)) {
+        int thoff;
+
+        if (skb_transport_header_was_set(skb))
+            thoff = skb_transport_header(skb) - skb_mac_header(skb);
+        else
+            thoff = nhoff + sizeof(struct ipv6hdr);
+        if (ntohs(ipv6_hdr(skb)->payload_len) > skb->len - thoff)
+            truncate = true;
+    }

     if (skb_cow_head(skb, dev->needed_headroom ?: t->hlen))
         goto tx_err;

```