CVE-2017-7477

描述：Heap-based buffer overflow in drivers/net/macsec.c in the MACsec module in the Linux kernel through 4.10.12 allows attackers to cause a denial of service or possibly have unspecified other impact by leveraging the use of a MAX\_SKB\_FRAGS+1 size in conjunction with the NETIF\_F\_FRAGLIST feature, leading to an error in the skb\_to\_sgvec function.

软件：MACsec module in the Linux kernel through 4.10.12

源码：drivers/net/macsec.c

出现位置： 2716-2723行

关键源码：

#define MACSEC\_FEATURES \

(NETIF\_F\_SG | NETIF\_F\_HIGHDMA | NETIF\_F\_FRAGLIST)

static struct lock\_class\_key macsec\_netdev\_addr\_lock\_key;

static int macsec\_dev\_init(struct net\_device \*dev)

dev->features = real\_dev->features & MACSEC\_FEATURES;

dev->features |= NETIF\_F\_LLTX | NETIF\_F\_GSO\_SOFTWARE;

dev->needed\_headroom = real\_dev->needed\_headroom +

MACSEC\_NEEDED\_HEADROOM;

dev->needed\_tailroom = real\_dev->needed\_tailroom +

MACSEC\_NEEDED\_TAILROOM;

if (is\_zero\_ether\_addr(dev->dev\_addr))

eth\_hw\_addr\_inherit(dev, real\_dev);

if (is\_zero\_ether\_addr(dev->broadcast))

memcpy(dev->broadcast, real\_dev->broadcast, dev->addr\_len);

Note：Macsec specifically does this:

size += sizeof(struct scatterlist) \* (MAX\_SKB\_FRAGS + 1);

tmp = kmalloc(size, GFP\_ATOMIC);

\*sg = (struct scatterlist \*)(tmp + sg\_offset);

...

sg\_init\_table(sg, MAX\_SKB\_FRAGS + 1);

skb\_to\_sgvec(skb, sg, 0, skb->len);

Specifying MAX\_SKB\_FRAGS + 1 is the right answer usually, but not if you're

using NETIF\_F\_FRAGLIST, in which case the call to skb\_to\_sgvec will

overflow the heap, and disaster ensues.