author	Dan Carpenter <dan.carpenter@oracle.com>	2022-10-28 18:05:00 +0300
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2022-11-10 17:46:52 +0100
commit	e877f8fa49fbccc63cb2df2e9179bddc695b825a (patch)	
tree	42d2962ad0acc61db8585a11a68f647d8e16f6da	
parent	81c3e3efbc44aae1172a68e74c11dd5fe493075a (diff)	
download	linux-e877f8fa49fbccc63cb2df2e9179bddc695b825a.tar.gz	

net: sched: Fix use after free in red_enqueue()

```
[ Upstream commit 8bdc2acd420c6f3dd1f1c78750ec989f02a1e2b9 ]

We can't use "skb" again after passing it to qdisc_enqueue().  This is
basically identical to commit 2f09707d0c97 ("sch_sfb: Also store skb
len before calling child enqueue").

Fixes: d7f4f332f082 ("sch_red: update backlog as well")
Signed-off-by: Dan Carpenter <dan.carpenter@oracle.com>
Reviewed-by: Eric Dumazet <edumazet@google.com>
Signed-off-by: David S. Miller <davem@davemloft.net>
Signed-off-by: Sasha Levin <sashal@kernel.org>
```

Diffstat

-rw-r--r-- net/sched/sch_red.c 4

1 files changed, 3 insertions, 1 deletions

```
diff --git a/net/sched/sch_red.c b/net/sched/sch_red.c
index 0424aa747c341c..afe0c2d689b176 100644
--- a/net/sched/sch_red.c
+++ b/net/sched/sch_red.c
@@ -63,6 +63,7 @@ static int red_enqueue(struct sk_buff *skb, struct Qdisc *sch,
 {
        struct red_sched_data *q = qdisc_priv(sch);
        struct Qdisc *child = q->qdisc;
+       unsigned int len;
        int ret;

        q->vars.qavg = red_calc_qavg(&q->parms,
@@ -98,9 +99,10 @@ static int red_enqueue(struct sk_buff *skb, struct Qdisc *sch,
                break;
        }

+       len = qdisc_pkt_len(skb);
        ret = qdisc_enqueue(skb, child, to_free);
        if (likely(ret == NET_XMIT_SUCCESS)) {
-               qdisc_qstats_backlog_inc(sch, skb);
+               sch->qstats.backlog += len;
                sch->q.qlen++;
        } else if (net_xmit_drop_count(ret)) {
                q->stats.pdrop++;
```