

FW: problem: TCPIP process loops and priority reduced to 1.

# FW: problem: TCPIP process loops and priority reduced to 1.

---

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2006-11/msg00226.html>

---

- *From:* "Hemanth" <[hemanth.thummala@xxxxxxxx](mailto:hemanth.thummala@xxxxxxxx)>
  - *Date:* Fri, 24 Nov 2006 17:18:09 +0530
- 

Some more info on this:

from tcpcb of FIN\_WAIT\_1 end i have seen t\_rxtshift value is 2, means 2 probe requests have gone from this side.

Normally when probe request is going the snd\_nxt value wont be incremented. In our case if the probe request is sent with FIN (dont know the possiblity), then the other side while this packet is processed in tcp\_input.c we can have this following scenario.

```
/*
 * If FIN is received ACK the FIN and let the user know
 * that the connection is closing.
 */
if (tiflags & TH_FIN) {
if (TCPS_HAVERCVDFIN(tp->t_state) == 0) {
socantrcvmore(so);
tp->t_flags |= TF_ACKNOW;
tp->rcv_nxt++; =====>>> here the rcv_nxt is incremented if the probe
contains FIN.
}
}
```

This may be the reason for getting the incremented rcv\_nxt(XXXX13) value than the expected snd\_max(XXXX14).

Now my question is : In what scenarios a FIN can be set in a probe request??

looking for urgent replies..

Regards,  
Hemanth,

---

From: Hemanth [<mailto:hemanth.thummala@xxxxxxxx>]  
Sent: Friday, November 24, 2006 1:35 PM

FW: problem: TCPIP process loops and priority reduced to 1.

FW: problem: TCPIP process loops and priority reduced to 1.

To: 'freebsd-net@xxxxxxxxxxxxx'

Subject: problem: TCPIP process loops and priority reduced to 1.

Importance: High

Hi All,

As mentioned in the subject the TCPIP is looping and its priority is reducing to 1 by the kernel (BSD kind)

From the dump, we were able to identify the inpcb's that are causing the problem.

inpcb:0x222f1a0c, lport:10143, laddr:0x7f000001 (127.0.0.1)

fport: 1378 faddr:0x7f000001 (127.0.0.1)

tcpcb:0x2270760c inpcb:0x222f1a0c flags:0x01e0

and

inpcb:0x224a850c, lport: 1378, laddr:0x7f000001 (127.0.0.1)

fport:10143 faddr:0x7f000001 (127.0.0.1)

tcpcb:0x2268d10c inpcb:0x224a850c flags:0x01e0 flags:0x0000

If we see the tcpcb structure for both structures

\$5 = {

seg\_next\_md = 0xfffc0000,

t\_state = 0x5,====>>> CLOSE\_WAIT state

t\_rxtshift = 0x0,

t\_rxtcur = 0x3fc,

FW: problem: TCPIP process loops and priority reduced to 1.

FW: problem: TCPIP process loops and priority reduced to 1.

t\_dupacks = 0x0,  
t\_maxseg = 0x7d00,  
t\_force = 0x0,  
t\_flags = 0x1e0,  
t\_template = 0x22776bd4,  
t\_inpcb = 0x224a850c,  
snd\_una = 0xbfaeb6d9,  
snd\_nxt = 0xbfaeb6d9,  
snd\_up = 0xbfaeb6d9,  
snd\_wl1 = 0xbfb19e13,  
snd\_wl2 = 0xbfaeb6d9,  
iss = 0xbfaeb678,  
snd\_wnd = 0x7d00,  
rcv\_wnd = 0xf96,  
rcv\_nxt = 0xbfb19e15,  
rcv\_up = 0xbfb19e13,  
irs = 0xbfb0a0a2,  
rcv\_adv = 0xbfb19daa,  
snd\_max = 0xbfaeb6d9,  
inv\_SYN\_seqbase = 0xffaeb678,  
snd\_cwnd = 0x17f10,  
snd\_ssthresh = 0x3fffc000,  
t\_idle = 0x0003934ba0b407c6,  
t\_rtt = 0x0,  
t\_rtseq = 0xbfaeb6c8,  
t\_srtt = 0xba2,

FW: problem: TCPIP process loops and priority reduced to 1.

FW: problem: TCPIP process loops and priority reduced to 1.

```
t_rttvar = 0x288,  
max_rcvd = 0x7d6a,  
max_sndwnd = 0x7d00,  
t_oobflags = 0x0,  
t_iobc = 0x0,  
tm_running = 0x0, No timer running  
tm_q = 0x8003378,  
tm_wait = 0x3fc,  
tm_exp_time = 0x0003934b7cb490dc,  
saved_rtt_clock = 0x0003934b7ca5007c,  
conn_exp_time = 0x0003934b811903db,  
tm_prev = 0x0,  
tm_next = 0x0,  
on_delay_ack_q = 0x0,  
dack_next = 0x0,  
dack_prev = 0x0,  
snd_scale = 0x0,  
rcv_scale = 0x0,  
request_r_scale = 0x0,  
requested_s_scale = 0x0,  
ts_recent = 0x6e49637,  
ts_recent_age = 0x6e49638,  
last_ack_sent = 0xbfb19e15  
}
```

FW: problem: TCPIP process loops and priority reduced to 1.

FW: problem: TCPIP process loops and priority reduced to 1.

And

\$8 = {

seg\_next\_md = 0xfffc0000,

t\_state = 0x6, =====>>> FIN\_WAIT\_1

t\_rxtshift = 0x2,

t\_rxtcur = 0x3eb,

t\_dupacks = 0x0,

t\_maxseg = 0x7d00,

t\_force = 0x0,

t\_flags = 0x1e0,

t\_template = 0x22a1e1d4,

t\_inpcb = 0x222f1a0c,

snd\_una = 0xbfb19e13,

snd\_nxt = 0xbfb19e13,

snd\_up = 0xbfb19e13,

snd\_wl1 = 0xbfaeb6d9,

snd\_wl2 = 0xbfb19e13,

iss = 0xbfb0a0a2,

snd\_wnd = 0x0,

rcv\_wnd = 0x7d00,

rcv\_nxt = 0xbfaeb6d9,

rcv\_up = 0xbfaeb6d9,

irs = 0xbfaeb678,

rcv\_adv = 0xbfaf33d9,

snd\_max = 0xbfb19e14,

FW: problem: TCPIP process loops and priority reduced to 1.

FW: problem: TCPIP process loops and priority reduced to 1.

```
inv_SYN_seqbase = 0xffaeb678,  
snd_cwnd = 0xffff,  
snd_ssthresh = 0x3fffc000,  
t_idle = 0x0003934ba0b407c6,  
t_rtt = 0x0,  
t_rtseq = 0xbfb17daa,  
t_srtt = 0x1f40,  
t_rttvar = 0x3,  
max_rcvd = 0x0,  
max_sndwnd = 0x7d00,  
t_oobflags = 0x0,  
t_iobc = 0x0,  
tm_running = 0x2,===== >> PERSIST TIMER is running on this end.  
tm_q = 0x8004460,  
tm_wait = 0x1388,  
tm_exp_time = 0x0003934ba1005255,  
saved_rtt_clock = 0x0003934b7df4342c,  
conn_exp_time = 0x0003934b811903de,  
tm_prev = 0x0,  
tm_next = 0x0,  
on_delay_ack_q = 0x0,  
dack_next = 0x0,  
dack_prev = 0x0,  
snd_scale = 0x0,  
rcv_scale = 0x0,
```

FW: problem: TCPIP process loops and priority reduced to 1.

FW: problem: TCPIP process loops and priority reduced to 1.

```
request_r_scale = 0x0,  
requested_s_scale = 0x0,  
ts_recent = 0x6e4917e,  
ts_recent_age = 0x6e4917e,  
last_ack_sent = 0xbfaeb6d9  
}
```

If we closely look into this structure, the `snd_max` of one end (which is in `FIN_WAIT_1` state) is less than the `rcv_nxt` of another end (which is in `CLOSE_WAIT`). When the end (present in `CLOSE_WAIT`) sends any frame it will put `rcv_nxt` field in the `ti_ack` field.

Because of the below code, a check `SEQ_GT(ti->ti_ack, tp->snd_max)` at the end which is in `FIN_WAIT_1` will fail resulting in it to drop the packet and send an ack with ack sequence number it is expecting.

Snippet from `tcp_input()`

```
1523 tp->t_dupacks = 0;  
1524 if (SEQ_GT(ti->ti_ack, tp->snd_max)) {  
1525     tcpstat.tcps_rcvacktoomuch++;  
1526     goto dropafterack;  
1527 }
```

If we dump the `tcpstat` information from the online `cpu dump`:

FW: problem: TCPIP process loops and priority reduced to 1.

FW: problem: TCPIP process loops and priority reduced to 1.

\$2 = {

...

...

...

tcps\_connattempt = 382657891,

tcps\_rcvtotal = 272941912,

tcps\_sndtotal = 272848321,

tcps\_noport = 13535,

tcps\_rcvurp = 0,

...

tcps\_sndacks = -394474409,

tcps\_rcvacktoomuch = 34255849,

tcps\_rcvackpack = 425941141,

tcps\_rcvackbyte = 603776015,

tcps\_rcvwinupd = 651289,

tcps\_predack = 2280435060,

..

tcps\_sc\_aborted = 0,

tcps\_sc\_dupesyn = 0,

tcps\_sc\_dropped = 0

and the tcpstat captured in the saveabend

tcpstat =

...

FW: problem: TCPIP process loops and priority reduced to 1.

FW: problem: TCPIP process loops and priority reduced to 1.

...

tcps\_connattempt = 382657891

tcps\_rcvtotal = 421811861

tcps\_sndtotal = 421718270

tcps\_noport = 13535

tcps\_rcvurp = 0

...

...

tcps\_sndacks = -245604460

tcps\_rcvacktoomuch = 108690823

tcps\_rcvackpack = 425941141

tcps\_rcvackbyte = 603776015

tcps\_rcvwinupd = 651289

tcps\_predack = 2280435060

...

tcps\_sc\_aborted = 0

tcps\_sc\_dupesyn = 0

tcps\_sc\_dropped = 0

We see lot of difference in three values, tcps\_rcvtotal,tcps\_sndtotal and tcps\_rcvacktoomuch and rest of the values remains the same. These values have increased ie almost doubled between time when cpu dump was taken ( Priority was 9) and the save abend (priority is 1). We presume that this observation is important clue for the problem and will work further by looking at code in what scenarios these values will increase.

This will cause another end to send ack again with incorrect sequence number. These events ie of repeated acks continue which led the tcpip proc to loop and hence priority was reduced to 1. This explains the reason behind

FW: problem: TCPIP process loops and priority reduced to 1.

FW: problem: TCPIP process loops and priority reduced to 1.

why only `tcps_rcvtotal`, `tcps_sndtotal`, `tcps_rcvduppack` and `tcps_rcvacktoomuch` are getting incremented in `tcpstats`.

We simulated the problem by making one end to send ACK frame with ack seq number higher than the expected. This resulted in loop and priority was reduced to one.

We are now looking at the scenarios when an ACK packet can come with ack sequence number higher than expected.

We are wondering in which scenarios this type of problem can occur in BSD.

I am looking for urgent feedbacks.

Regards,

Hemanth.

---

freebsd-net@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-net>

To unsubscribe, send any mail to "freebsd-net-unsubscribe@xxxxxxxxxxx"