New issue

Jump to bottom

# ipf memleak #226

Closed **junka** opened this issue on Sep 29, 2021 · 10 comments

**junka** commented on Sep 29, 2021 • edited

Hi,

This commit introduce a mempool memleak.
640d4db788eda96bb904abcfc7de2327107bafe1

If I keep sending first fragment as an attack, the mempool would be exhausted.

I think revert it and respect the dnsteal flag as @ Wang Liang mentioned in
https://mail.openvswitch.org/pipermail/ovs-dev/2021-April/382098.html
would be better.

```
frag->pkt = dnsteal ? dp_packet_clone(pkt) : pkt;
```

**orgcandman** commented on Sep 29, 2021

No - this leak looks to be a different issue - the ipf expiry timer is taking extremely long to kick in. There is a purge timer, and it should be running at 15s to purge the free list. However, look at what happens:

2021-09-29T17:41:07.491Z|00093|unixctl|DBG|replying with success, id=0: " Fragmentation Module Status
---
v4 enabled: 1
v6 enabled: 1
max num frags (v4/v6): 1000
num frag: 1
min v4 frag size: 1200
v4 frags accepted: 13
v4 frags completed: 12
v4 frags expired: 0
v4 frags too small: 0
v4 frags overlapped: 0
v4 frags purged: 0
min v6 frag size: 1280
v6 frags accepted: 0
v6 frags completed: 0
v6 frags expired: 0
v6 frags too small: 0
v6 frags overlapped: 0
v6 frags purged: 0

Almost 1m later:
2021-09-29T17:42:04.762Z|00111|unixctl|DBG|replying with success, id=0: " Fragmentation Module Status
---
v4 enabled: 1
v6 enabled: 1
max num frags (v4/v6): 1000
num frag: 1
min v4 frag size: 1200
v4 frags accepted: 13
v4 frags completed: 12
v4 frags expired: 0
v4 frags too small: 0
v4 frags overlapped: 0
v4 frags purged: 0
min v6 frag size: 1280
v6 frags accepted: 0
v6 frags completed: 0
v6 frags expired: 0
v6 frags too small: 0
v6 frags overlapped: 0
v6 frags purged: 0

Finally purged:
2021-09-29T17:43:32.555Z|00113|unixctl|DBG|replying with success, id=0: " Fragmentation Module Status
---
v4 enabled: 1
v6 enabled: 1
max num frags (v4/v6): 1000
num frag: 0
min v4 frag size: 1200
v4 frags accepted: 13
v4 frags completed: 12
v4 frags expired: 0
v4 frags too small: 0
v4 frags overlapped: 0
v4 frags purged: 1
min v6 frag size: 1280
v6 frags accepted: 0
v6 frags completed: 0
v6 frags expired: 0
v6 frags too small: 0
v6 frags overlapped: 0
v6 frags purged: 0

Even worse, this timer is not user configurable - so the only way to fix is to adjust the code. I suspect that you see it look worse because now we clone a packet, so under high packet conditions, we hold even bigger buffers for frags. This isn't a leak, but it is a problem (you should see after quiescent period, the number of outstanding frags should go down).

I will look into the purge timeout - it isn't configurable by the user and that would help here as well.

**junka** commented on Sep 29, 2021 — Author

No. I did another test which use iperf to send udp packet over 2000 bytes. My vm's MTU is 1500.

```
iperf -c 192.168.0.3 -u -i 1 -l 2000
```

After a while all frags have be completed.

```
        Fragmentation Module Status
         ---------------------------
         v4 enabled: 1
         v6 enabled: 1
         max num frags (v4/v6): 1000
         num frag: 0
         min v4 frag size: 1200
         v4 frags accepted: 3960
         v4 frags completed: 3960
         v4 frags expired: 0
         v4 frags too small: 0
         v4 frags overlapped: 0
         v4 frags purged: 0
         min v6 frag size: 1280
         v6 frags accepted: 0
         v6 frags completed: 0
         v6 frags expired: 0
         v6 frags too small: 0
         v6 frags overlapped: 0
         v6 frags purged: 0
```

But the mempool did not get released.
This can be confirmed using `dpdkprocinfo`

```
./dpdk-proc-info -a 0000:17:00.0 --file-prefix=`pidof ovs-vswitchd`- -- --show-mempool=ovsd78f4ce801021580262144
```

And the result is like

```
EAL: No legacy callbacks, legacy socket not created
========== show - MEMPOOL ==========
  - Name: ovsd78f4ce801021580262144 on socket 1
  - flags:
         -- No spread (n)
         -- No cache align (n)
         -- SP put (n), SC get (n)
         -- Pool created (y)
         -- No IOVA config (n)
  - Size 262144 Cache 512 element 2880
  - header 64 trailer 64
  - private data size 64
  - memezone - socket 1
  - Count: avail (258196), in use (3948)
  - ops_index 1 ops_name ring_mp_mc
```

The `in use` numbers increased as packets send out. And eventually it reached total size.

**orgcandman** commented on Sep 30, 2021

What is the output from ovs-appctl netdev-dpdk/get-mempool-info [netdev] ?

**orgcandman** commented on Oct 4, 2021

Perhaps the issue is here:

```
@@ -948,6 +948,8 @@ ipf_extract_frags_from_batch(struct ipf *ipf, struct dp_packet_batch *pb,
             if (!ipf_handle_frag(ipf, pkt, dl_type, zone, now, hash_basis,
                                  pb->do_not_steal)) {
```

```
                dp_packet_batch_refill(pb, pkt, pb_idx);
+           } else {
+               dp_packet_delete(pkt);
            }
            ovs_mutex_unlock(&ipf->ipf_lock);
        } else {
```

Please try with this patch and let me know

**orgcandman** commented on Oct 4, 2021

Note - I applied this to a pre-liang version of the code, because it seems this leak has been present for a very long time (and is independent of the dnsteal flag - it is a completely missed branch and would happen anyway).

**orgcandman** commented on Oct 5, 2021

https://patchwork.ozlabs.org/project/openvswitch/patch/20211005181844.734362-1-aconole@redhat.com/

This patch should address the outstanding buffer. Please confirm it works in your environment.

**ovsrobot** pushed a commit to ovsrobot/ovs that referenced this issue on Oct 5, 2021

`ipf: release unhandled packets from the batch` ··· ✓ `201fd93`

**junka** commented on Oct 6, 2021 Author

Hi Aaron,
My apologies for the late reply.
I was on my vacation since last week. I'll try your patch and reach out to you with the result asap.

**junka** commented on Oct 8, 2021 Author

Hi Aaron,
I've tested your patch, no leak here observed for now.

Close it now.

**junka** closed this as completed on Oct 8, 2021

---

**orgcandman** commented on Oct 11, 2021

Thanks so much - would you consider replying upstream to the mailing list with a "tested-by" tag?

**junka** commented on Oct 12, 2021 • edited Author

> Thanks so much - would you consider replying upstream to the mailing list with a "tested-by" tag?

Ah yes.

Done.

**aserdean** pushed a commit to openvswitch/ovs that referenced this issue on Oct 12, 2021

`ipf: release unhandled packets from the batch` ··· ✓ `803ed12`

**aserdean** pushed a commit to openvswitch/ovs that referenced this issue on Oct 12, 2021

`ipf: release unhandled packets from the batch` ··· ✓ `39c4269`

**aserdean** pushed a commit to openvswitch/ovs that referenced this issue on Oct 12, 2021

`ipf: release unhandled packets from the batch` ··· ✓ `ec44c50`

**aserdean** pushed a commit to openvswitch/ovs that referenced this issue on Oct 12, 2021

`ipf: release unhandled packets from the batch` ··· ✓ `4873b77`

**aserdean** pushed a commit to openvswitch/ovs that referenced this issue on Oct 12, 2021

`ipf: release unhandled packets from the batch` ··· ✓ `9efa2ea`

**aserdean** pushed a commit to openvswitch/ovs that referenced this issue on Oct 12, 2021

`ipf: release unhandled packets from the batch` ··· ✓ `0fd17fb`

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants