Subject: Re: figuring out why openvz kills processes
Posted by kir on Fri, 11 Jul 2008 13:38:13 GMT
View Forum Message <> Reply to Message

Geoffrey D. Bennett wrote:
> Hi there,
>
> I'm having an issue with a process dying (being killed by OpenVZ
> limits, presumably), and I can't figure out exactly why it's getting
> killed.
>
> Background info:
> - kernel 2.6.18-53.1.19.el5.028stab053.14
> - CentOS5 host, 2 CentOS5 guests
> - The host has 2GB memory, 0.5GB swap
> - One guest is only running BIND (plus the usual, sshd, syslogd,
> sendmail, crond services). Am having no issues with this guest.
> privvmpages is set to provide it up to 256MB memory, and it's using
> about half that.
> - The other guest is running postgresql, java, apache, and freeradius.
>
> The problem is that freeradius keeps dying. Whenever it dies, failcnt
> on privvmpages goes up and indeed the maxheld privvmpages value is
> above the limit value, so I guess the issue is that OpenVZ thinks that
> something is taking too much memory and is killing radiusd (no other
> failcnt numbers go up -- only privvmpages).
>
> There doesn't appear to be anything logged in the dmesg output on the
> host or the guest to indicate that anything was killed due to a limit
> being exceeded (should there be?).
>
> A few strange things:
>
> - Although the maxheld privvmpages value is above the limit, I've
> never seen the held privvmpages value get anywhere near the limit,
> even checking the value only seconds before radiusd gets killed, the
> held privvmpages value is under half the limit, eg. just before
> radiusd is killed:
>
> 2008-07-11 06:39:24:
> uid resource held maxheld barrier limit failcnt
> privvmpages 224497 581366 506368 557056 486
>
> Then 10 seconds later (radiusd was killed and possibly restarted
> sometime in this interval):
>
> 2008-07-11 06:39:34:
> uid resource held maxheld barrier limit failcnt

> privvmpages 182445 581366 506368 557056 487
>
> (is there any way to reset the maxheld values without restarting the
> guest?)
>
> - Similarly, the output of free doesn't indicate anything wrong:
>
> 2008-07-11 06:39:24:
> total used free shared buffers cached
> Mem: 2071924 898092 1173832 0 0 0
> -/+ buffers/cache: 898092 1173832
>
> 2008-07-11 06:39:34:
> total used free shared buffers cached
> Mem: 2071924 729884 1342040 0 0 0
> -/+ buffers/cache: 729884 1342040
>
> - I've found that I can reproduce the issue on demand by sending many
> RADIUS requests to radiusd at once, but watching what radiusd does
> with ltrace -f doesn't show anything out of the ordinary. I summed
> up all the malloc() requests and saw only 22MB requested.
>
> - Finally, using strace -f to see what radiusd was doing -- there were
> only about 22MB worth of calls to brk() (matching malloc(), as you'd
> expect). And summing the mmap() length parameters (not counting
> munmap() calls) I only came up with 300MB, well within the free
> memory.
>
> Any ideas on debugging this?

OpenVZ doesn't kill anything in this case. It employs killing processes only if there is no any other way to enforce the UBC limits, and there are other ways in this case -- just return ENOMEM from malloc/setbrk.

I guess most probably it's just radiusd calls malloc() which fails (because of privvmpages shortage) and then either radiusd dies explicitly, or it fails to check the error code from malloc and uses the pointer returned by malloc (NULL) and dies with segfault.

free in either VE or on the host system will not help you