Subject: Re: vzkernel-3.10.0-x releases stopped since Sept?
Posted by wsap on Fri, 28 Dec 2018 02:30:23 GMT

View Forum Message <> Reply to Message

khorenko wrote on Mon, 03 December 2018 12:10i'm essentially saying that Virtuozzo devs work on Virtuozzo - payed version - and do as much as they can to make OpenVZ users happy, but with no additional devs/QA efforts (which are unpayed, sorry).
And building stable kernels + readykernel patches - are efforts, it cannot be automated.
And TESTING them are BIG efforts, because tests do fail and QA (humans!) have to investigate issues.

I do actually completely understand that you'd want to invest as little time as possible into maintaining the open source base systems. This is why I suggested keeping these types of releases strictly to security patches released to the upstream CentOS kernel, and nothing more.

My suggestion is that *after* these patches have been tested as readykernel patches (which the dev team has to do anyway for commercial customers), the same patches/diffs be applied to the currently stable kernel release, compiled, then quickly tested to confirm the kernel boots successfully, then released to the repo (or perhaps the testing repo for a week before being moved to release).

To further save time investment, every time a ReadyKernel patch is created, the same diffs could be quickly applied to the stable kernel on a test box. Then once a month, a cron script could freshly compile and install the testing kernel and reboot the box. Upon successful reboot it would upload the new kernel release to the repo. If the automated compiling and reboot fails, then it seems plausible the failure could be helpful feedback for devs and likely for ReadyKernel commercial customers as well.

A couple notes on this:

- Doing this should be considerably more stable for end-users than suggesting that they use factory kernel versions. And it would be much more secure for end-users than not getting security patches for 60+ days as it will result in a security patched kernel release roughly once a month. End-users can then choose whether they want to reboot into it immediately or a bit later to keep their reboots to a minimum.
- Ultimately, devs already have to do the work of testing the exact same patches via ReadyKernel, so this work does not need to be duplicated or wasted time. The only additional dev time would be the act of applying the patches to the current release kernel. Even the act of installing the kernel on a test box and rebooting the box to ensure it comes back online could be automated as described above. A seasoned kernel dev would take probably 15 minutes to do this with automation and 60 minutes without automation each month.

Yes, this means a small amount of additional dev time monthly, but it also means you're sending important messages:

1. When users of OpenVZ 6 need to make the decision to choose either OpenVZ 7 or move to a different virtualisation system (which is going to happen en masse in the next year due to OpenVZ6 EOL Nov 2019), that the OpenVZ 7 transition is the simplest and optimal way forward

because it does not present them with new security challenges, as compared to vz6.
2. That Virtuozzo Linux 7 is at least on par security-wise with CentOS 7, which nobody can say is the case at the moment since security patches for Virtuozzo 7 are often delayed by months as compared to CentOS 7.

I've heard that Virtuozzo is preparing for stand-alone release of ReadyKernel and regardless of whether the Virtuozzo devs decide to do any of the above, I'll be first in line to purchase and use ReadyKernel on our OpenVZ 7 nodes. I do, however, still hope that you'll do something like what's described above to ensure that those who do not opt for a ReadyKernel subscription can best maintain the security of their nodes.

While I don't think this is nearly as comprehensive a solution, I'm grateful that @khorenko provided the rough steps to download the kernel sources, apply security patches, and compile the patched kernel. If that could be expanded into a more comprehensive wiki entry, I think that would be a great, albeit somewhat less useful, alternative.