

Packages - Bug #3127

[lvm2 2.03.14-1] Dependency on systemd on non-systemd (OpenRC) systems prevents lvm volumes from activating

2021-11-17 04:55 PM - DragOnFly

Status: unconfirmed	% Done: 0%
Priority: bug	
Assignee:	
Category:	
Description	
<p>The hook which is run when triggering automatic lvm activation for new volumes relies on "systemd-run", thus preventing LVM volumes from being set active at boot-time. This should be modified to openrc-run, or equivalent, in nonsystemd repos.</p> <p>Specifically this is done in the 69-dm-lvm-rules file –</p> <pre>/usr/lib/udev/rules.d/69-dm-lvm.rules:ENV{LVM_VG_NAME_COMPLETE}=="?*", RUN+="/usr/bin/systemd-run -r --no-block --property DefaultDependencies=no --unit lvm-activate-\$env{LVM_VG_NAME_COMPLETE} /usr/bin/lvm vgchange -aay --nohints \$env{LVM_VG_NAME_COMPLETE}"</pre> <p>For someone who has gone to great lengths to get rid of any-and-all-things systemd, it is unfortunate to see these things continue to crop up.</p>	

History

#1 - 2021-12-02 04:13 AM - bill-auger

FWIW, openrc (or any other init) does not eliminate the need for systemd - all standard parabola openrc systems include parts of systemd; because there are many programs which will not run otherwise - openrc can even be configured to use systemd as it's init

parts of systemd are effectively core system libraries now, required by essential programs, for example via 'device-mapper' alone:

```
'cryptsetup' 'grub' 'lvm2'
```

to "get rid of any-and-all-things systemd", you would need to maintain a system without 'systemd', 'nonsystemd', or 'elogind' (eg: by replacing 'elogind' with something else which provides 'systemd', 'systemd-libs', and 'libsystemd')

#2 - 2021-12-04 03:35 PM - DragOnFly

That would be too optimistic, utopian even. ;) I should have worded it differently, as I realize there are packages which provide a systemd-like layer for compatibility. What I meant was simply that "systemd-run" does not belong on a non-systemd system.

E.g, that the line above in:

```
/usr/lib/udev/rules.d/69-dm-lvm.rules
```

Is changed from–

```
ENV{LVM_VG_NAME_COMPLETE}=="?*", RUN+="/usr/bin/systemd-run -r --no-block --property DefaultDependencies=no --unit lvm-activate-$env{LVM_VG_NAME_COMPLETE} /usr/bin/lvm vgchange -aay --nohints $env{LVM_VG_NAME_COMPLETE}"
```

to–

```
ENV{LVM_VG_NAME_COMPLETE}=="?*", RUN+="/usr/bin/lvm vgchange -aay --nohints $env{LVM_VG_NAME_COMPLETE}"
```

or something similar so that LVM volumes activate.

The reason for routing this via systemd is apparently in the event there are hundreds of LVs to set active, as that will take "a couple of seconds" (which apparently is too long in the systemd-world...)

```
# When the VG is complete it can be activated, so  
# vgchange -aay <vgname> is run. It is run via
```

```
# systemd since it can take longer to run than
# udev wants to block when processing rules.
# (if there are hundreds of LVs to activate,
# the vgchange can take many seconds.)
```

#3 - 2021-12-04 03:52 PM - Drag0nFly

btw.—I am not sure if the abovementioned suggestion is the “correct” way to do this, as there is also an “IMPORT{program}” notation being used in the (Red Hat-provided) script, in addition to the “RUN+=” notation which runs vgchange via systemd-run.