Systemd

Page describes steps to verify systemd functionality and possible fixes for troublesome parts.

To check if everything works correctly use

```
systemctl status -a
```

and look for failed services if any.

Required overrides

Some systems require systemd override files to disable security hardening not supported by our OpenVZ kernel - this includes seccomp filter and memory deny write execute.

These can be disabled for specific service with following override file:

```
[Service]
SystemCallFilter=
MemoryDenyWriteExecute=no
```

File needs to be placed in

```
/etc/systemd/system/${SVC_NAME}.service.d/override.conf
```

To create it manually for e.g. systemd-journald.service use

```
systemctl edit systemd-journald
# paste override code from above
systemctl daemon-reload
systemctl start systemd-journald
```

Knot DNS

With systemd and OpenVZ, Knot DNS fails to start with the following error message:

```
... systemd[22357]: knot.service: Failed at step CAPABILITIES spawning
/usr/sbin/knotc: Invalid argument
-- Subject: Process /usr/sbin/knotc could not be executed
-- Defined-By: systemd
-- Support: https://www.debian.org/support
--
-- The process /usr/sbin/knotc could not be executed and failed.
--
-- The error number returned by this process is 22.
```

```
... systemd[1]: knot.service: Control process exited, code=exited status=218
... systemd[1]: Failed to start Knot DNS server.
```

The reason is that Knot DNS systemd unit specifies a few required capabilities which vpsFree does not support under OpenVZ, namely:

```
# cat /lib/systemd/system/knot.service

[...]
CapabilityBoundingSet=CAP_NET_BIND_SERVICE CAP_SETPCAP
AmbientCapabilities=CAP_NET_BIND_SERVICE CAP_SETPCAP
[...]
```

For explanation of what these capabilities mean visit this link.

The first step is either commenting these out or even better, overriding the settings.

```
systemctl edit knot.service

[Service]
CapabilityBoundingSet=~
AmbientCapabilities=
```

(note the ~ as a value for `CapabilityBoundingSet`).

and reload with `systemctl daemon-reload`.

Now Knot DNS starts but fails to bind to port `53/TCP` and `53/UDP` because without the capability `CAP_NET_BIND_SERVICE` Knot DNS can't bind to system ports (<1024) as a user `knot`. The solution is to let Knot DNS know to run as root first, bind the necessary ports and then switch to user `knot` afterwards.

So first we override once more and add `User=` and `Group=`:

```
systemctl edit knot.service

[Service]
User=
Group=
CapabilityBoundingSet=~
AmbientCapabilities=
```

Then we edit the Knot DNS configuration itself and specify user and group `knot` for it:

```
nano /etc/knot/knot.conf

server:
    [...]
    user: knot:knot

[...]
```

Reload once again with `systemctl daemon-reload` and you should be good to go.