**When rndg Will Not Start**

Do you ever have one of those Monday mornings. Yes, one of those! You would believe that with CentOS 7.1 no less little niggles will have been ironed out and the world would be a wonderful place. Apparently not, and we find that on CentOS 7 rngd will not start by default.

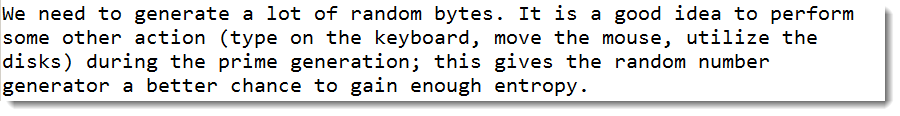
OK, there is a lot to look after and perfection is never there, even with my spelling. So believe me I am not throwing rocks but want to get it out there of how and why we start the rngd service.

**Firstly: The Why**

Many user and system programs in Linux will need **entropy**when working with cryptography. Entropy in Linux is defined as randomness collected by the Operating System. Originally this was collected from the pseudo-device **/dev/random** from data generated by device drivers and services. The data sent to /dev/random is known as the**entropy pool**and when the pool is empty the cryptographic service or user program may stop. This would not be great on your HTTPS enabled web server.

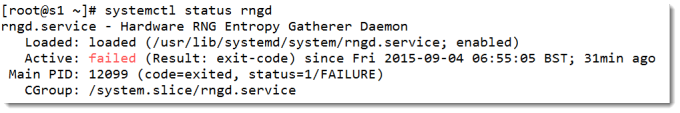
To ensure the entropy pool is not exhausted the device /dev/urandom is now used by default before failing back to /dev/random. Rather than collecting data from device drivers /dev/urandom will have random data directly fed to it from the rngd service. This is part of the **rng-tools** package on both Debian and Red Hat based systems.

A simple demonstration to show the exhaustion of the entropy pool when the service is not running we try to try to generate at new gpg key, (GNU Privacy Guard). If this is executed whilst the service is failed or not running entropy will be gathered from /dev/random and will most often prompt for more random data.

[](http://web.theurbanpenguin.com/wp-content/uploads/2015/09/entropy.png)If the rngd service is running there is always enough entropy in the pool.

**Secondly: The Problem**

OK, so I am sold on the idea of a limitless entropy pool. What is the problem with the service.  It doesn’t start, that is what the problem is!

[](http://web.theurbanpenguin.com/wp-content/uploads/2015/09/rngd.png)The command that the service is running from the service unit is: **rngd -f**

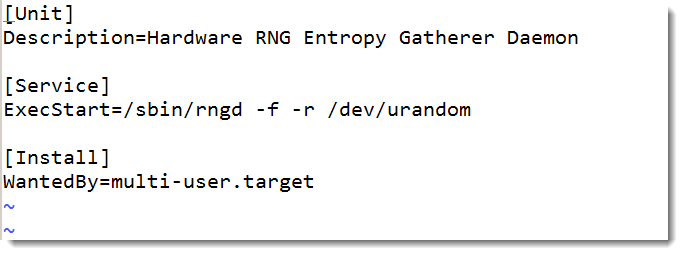
This is just a little wrong. First of all we would like ti to run is the background as a daemon service. The error is that the unit file does not specify the -r option or the path to the device file to use. This will default to **/dev/hwrandom** which does not exist.

**Thirdly: The Fix**

We can easily rectify the problem by editing the service  unit file:**/usr/lib/systemd/system/rngd.service**. The ExecStart line should be edited so that it reads as in the following:

ExecStart=/sbin/rngd -f -r /dev/urandom

This is also shown in the following screenshot:

[](http://web.theurbanpenguin.com/wp-content/uploads/2015/09/rndgservice.png)

We will need to reload the unit file once it has been edited. We can use the following command to achieve this:

# systemctl daemon-reload

With the new unit loaded we can now start the service and check the status:

# systemctl start rngd

# systemctl status rngd

The following video will step you through the process.