Overview
qRand augments a computer’s entropy pool with full entropy random bits, solving the problem of “entropy starvation”. This prevents performance degradation for applications that use entropy, or the security compromise of using low entropy pseudo-randomness.

The Challenge of Entropy Limitations
qRand solves the problem of “entropy starvation”, by augmenting a computer’s entropy pool when it falls below a lower bound.

Entropy starvation is a major concern, especially in environments using virtual machines, including in cloud infrastructure. It degrades performance, with applications failing to respond due to a lack of randomness for cryptographic operations. Equally worrisome is the fact that many applications use a “non-blocking” source of pseudo randomness to overcome this first issue. This can compromise security, resulting in vulnerabilities, including duplicate cryptographic keys.

RNG in Linux
Any process that needs random numbers can get them from /dev/random. However, /dev/random will only return random numbers if there is enough entropy available. If not, /dev/random simply blocks, resulting in performance degradation. Many applications remedy this using “non-blocking” sources of randomness such as /dev/urandom. This degrades security, resulting in potential vulnerabilities such as duplicated cryptographic keys. Other Linux packages that provide random numbers like “rngd” and “haveged” can also result in entropy dilution if insufficient entropy is available, with the potential for security risks.

This is particularly challenging in environments where normal entropy gathering does not yield enough entropy, for example in virtual machines or embedded devices.

qRand addresses these issues by feeding entropy into the entropy-pool of a computer. The entropy provided is delivered from QuintessenceLabs’ QRNG installed in a qCrypt appliance.

How qRand Works
qRand monitors entropy status on a computer, and when it falls below a defined lower bound, augments it with entropy from QuintessenceLabs’ quantum random number generator (QRNG), embedded in the qCrypt appliance. This enables applications on the computer to generate and use high quality cryptographic keys without any changes to the application itself.

Users can configure the behavior of qRand in several ways, such as setting the lower limit of the entropy status, the order of qCrypt devices from which to get entropy, and logging options.

QuintessenceLabs’ QRNG
QuintessenceLabs uses groundbreaking quantum technology to deliver random numbers with full-entropy at 1 Gbit/s. The QRNG is available as a qStream stand-alone appliance or PCIe card, or as part of the qCrypt product suite.
## SPECIFICATIONS

### qRand™

Quantum-Powered Entropy

### Key Features

- Linux daemon, running as a native system service, that monitors entropy status in system
- When entropy levels fall below lower limit, qRand retrieves entropy from the quantum random number generator embedded in qCrypt 300R or 350TSF
- User configurable

### User settings

- Lower bound of entropy (in bits)
- Entropy fill watermark (in bits)
- Enable/disable use of deterministic entropy sources
- Select from any available qCrypt random objects
- Enable/disable audit logging, log verbosity level

### Supported OS

- Ubuntu (64-bit) 16.04, 18.04
- RHEL (64-bit) 6.10, 7.3, 7.6
- Support for more Linux distributions planned

### qCrypt 300R or 350TSF

Refer to qCrypt product sheet for full specifications. Includes quantum random number generator with the following features:

- Quantum random number generator delivering 100% entropy
- 8 Gbit/s quantum entropy source, 1Gbit/s conditioned entropy
- Meets all requirements of NIST SP 800-90A, 90B and 90C (draft) standards for Non-Deterministic Random Bit Generators
- Satisfies NIST SP 800-22 (NIST STS) and Dieharder tests