# CIS 751 Lecture Assignment 9

Chuck Zumbaugh

October 24, 2025

The /dev/random and /dev/urandom interfaces are kernel interfaces that generate random numbers when read. Both of these gather environmental noise (entropy) from device drivers and other sources into an entropy pool that feeds the pseudorandom number generator (PRNG).

## 1 Linux /dev/urandom

/dev/urandom provides random bytes using a PRNG seeded from the entropy pool. This is non-blocking, and in the event of low entropy, this will still return a random number, relying on the strength of the PRNG.

## 2 Linux /dev/random

This also provides random bytes using a PRNG seeded from the entropy pool. Unlike /dev/urandom, historically /dev/random was blocking. That is, reads from /dev/random will block in the event entropy is low and will continue to do so until sufficient environmental noise is gathered.

## 3 Differences between random and urandom

Historically, random was considered more secure than urandom because it would wait for additional environmental noise before providing a random number. Thus, it was generally used for very sensitive operations such as cryptographic keys. On the other hand, urandom was generally considered acceptable for most normal applications and could provide a stream

of random numbers because it was non-blocking. However, since Linux 5.6 /dev/random is also non-blocking except early in the boot process. The /dev/random interface is now considered a legacy interface, with /dev/urandom being preferred and sufficient for all use cases. Additionally, /dev/urandom can generate a larger number of random bytes (up to 32 MB) compared to /dev/random (up to 512 bytes).