systemd-boot-random-seed.service

- Description
- See Also

systemd

systemd-boot-random-seed.service

8

systemd-boot-random-seed.service

Refresh boot loader random seed at boot

`systemd-boot-random-seed.service`

Description

`systemd-boot-random-seed.service` is a system service that automatically refreshes the boot loader random seed stored in the EFI System Partition (ESP), from the Linux kernel entropy pool. The boot loader random seed is primarily consumed and updated by `systemd-boot(7)` from the UEFI environment (or `systemd-stub(7)` if the former is not used, but the latter is), and passed as initial RNG seed to the OS. It is an effective way to ensure the OS comes up with a random pool that is fully initialized.

The service also automatically generates a 'system token' to store in an EFI variable in the system's NVRAM. The boot loader may then combine the on-disk random seed and the system token by cryptographic hashing, and pass it to the OS it boots as initialization seed for its entropy pool. Note: the random seed stored in the ESP is refreshed on *every* reboot ensuring that multiple subsequent boots will boot with different seeds. On the other hand, the system token is generated randomly *once*, and then persistently stored in the system's EFI variable storage, ensuring the same disk image will not result in the same series of boot loader seed values if used on multiple

systems in parallel.

The `systemd-boot-random-seed.service` unit invokes the `bootctl random-seed` command, which updates the random seed in the ESP, and initializes the system token if it is not initialized yet. The service is conditionalized so that it is run only when a boot loader is used that implements the Boot Loader Interface.

For further details see `bootctl(1)`, regarding the command this service invokes.

Note the relationship between `systemd-boot-random-seed.service` and `systemd-random-seed(8)`. The former maintains the random seed consumed and updated by the boot environment (i.e. by `systemd-boot(7)` or `systemd-stub(7)`), the latter maintains a random seed consumed and updated by the OS itself. The former ensures that the OS has a filled entropy pool already during earliest boot when regular disk access is not available yet (i.e. when the OS random seed cannot be loaded yet). The latter is processed much later, once writable disk access is available. Thus it cannot be used to seed the initial boot phase, but typically has much higher quality of entropy. Both files are consumed and updated at boot, but at different times. Specifically:

1. In UEFI mode, the `systemd-boot(7)` or `systemd-stub(7)` components load the boot loader random seed from the ESP, hash it with available entropy and the system token, and then update it on disk. A derived seed is passed to the kernel which writes it to its entropy pool.
2. In userspace the `systemd-random-seed.service` service loads the OS random seed, writes it to the kernel entropy pool, and then updates it on disk with a new value derived from the kernel entropy pool.
3. In userspace the `systemd-boot-random-seed.service` service updates the boot loader random seed with a new value derived from the kernel entropy pool.

This logic should ensure that the kernel's entropy pool is seeded during earliest boot already, if possible, but the highest quality entropy is propagated back to both on-disk seeds.

See Also

`systemd(1)`, `random(4)`, `bootctl(1)`, `systemd-boot(7)`, `systemd-stub(7)`, `systemd-random-seed.service(8)`

□□□ #2
□ □□□ □□□ 2026-10-31 14:59:26 UTC
□ □□□ □□□ 2026-10-31 14:59:26 UTC