/dev/random and /dev/urandom

Description

Both **/dev/random** and **/dev/urandom** are special files in Unix-like operating systems (like Linux) that serve as interfaces to the kernels's random number generator. They provide a way to access unpredictable random data that is generated from mouse movements, keyboard timing, device drivers, and other hardware events. They are critical for various cryptographic and security-related functions.

Key Differences

/dev/random generates random bytes using unpredictable sources like mouse movements, keyboard timings, and other hardware events. It pauses (blocks) if there is not enough randomness (called entropy) available, and it waits until it can produce high-quality random data. This makes it slower but faster for tasks that need high security consideration, like generating cryptographic keys. On the other hand, **/dev/urandom** or (**unlimited random**) also uses the same entropy pool but doesn't pause when the pool is low. Instead, it keeps providing random data by reusing what is available, even if there is less randomness at the moment. While this might make the data slightly less unpredictable during low entropy, it is still secure enough for most purposes. Therefore, /dev/urandom is faster and provides data instantly, making it a better choice for situations where random data is needed quickly.

Security Considerations

/dev/random is highly preferred for critical security tasks, such as generating cryptographic keys, because it waits until there is enough randomness to produce unpredictable data. However, modern systems consider **/dev/urandom** secure enough for general use. It strikes a good balance between security and performance, making it suitable for many applications where absolute unpredictability is less critical.

Common Uses

/dev/random is generally used for generating cryptographic keys and in high-security applications where the highest level of unpredictability is required. Conversely, **/dev/urandom** is used for generating session identifiers, salt values for hashing functions, and general-purpose random number generation in software where blocking would pose a problem.

Practical Usage Examples

1. Generate a random string of 15 characters using alphanumeric characters and specified special symbols from the /dev/urandom file: `sudo tr -dc 'A-Za-z0-9~!@$%^&*()_+-}{[]?><' < /dev/urandom | head -c 55; echo`
2. Generate 32 bytes of random data from /dev/random: `head -c 32 /dev/random`

3. Generate 24 bytes of random data from /dev/urandom, encodes it in Base64, and saves it to password.txt: `openssl rand -base64 24 > password.txt`

References

[1] IBM, "uRandom and random devices," IBM Documentation, AIX 7.2. [Online]. Available: https://www.ibm.com/docs/ru/aix/7.2?topic=files-urandom-random-devices. [Accessed: Aug. 14, 2024].

[2] T. Pornin, "On Linux's Random Number Generation," NCC Group, Dec. 19, 2019. [Online]. Available: https://research.nccgroup.com/2019/12/19/on-linuxs-random-number-generation/. [Accessed: Aug. 15, 2024].

Prepared by: Zahid Muhammed
Date: August 21, 2024 - 16:42