Entropy

Entropy in cryptography refers to **the measure of randomness or unpredictability within a system**, which is critical for generating secure cryptographic keys, nonces, and initialization vectors. **The higher the entropy, the more secure** and less predictable the generated values are. Here's an overview of entropy, pseudo-random number generators (PRNGs), entropy buffer draining, and methods for filling an entropy buffer.

1. Entropy

- Definition: Entropy is **a measure of randomness in a system**, particularly relevant **for generating secure cryptographic values**. High entropy in key generation ensures that values are unpredictable and, thus, harder for attackers to guess or reproduce.
- Importance in Cryptography: Without sufficient entropy, cryptographic systems can become vulnerable to attacks. **Low entropy makes it easier for attackers to predict or brute-force cryptographic values**, such as keys or session tokens.
- Sources of Entropy: Entropy can come from various sources in a computer system, including **hardware-based random events**, **operating system processes**, or even **user-driven actions** (e.g., mouse movements or keystrokes).

2. PRNG (Pseudo-Random Number Generators)

- Definition: **A PRNG is an algorithm that generates sequences of numbers that approximate randomness**. These numbers are "pseudo-random" because they are derived from an initial value, or "seed," which is often generated from an entropy source.
- How It Works
    - PRNGs **use a seed value** (often derived from an entropy source) to start generating numbers. From that seed, they produce a sequence of numbers that seem random but are deterministic.
    - Common algorithms for PRNGs include Linear Congruential Generators (LCG), Mersenne Twister, and Xorshift.
- Applications: PRNGs are used in a variety of applications, including **simulations, games, and cryptographic functions**. However, in cryptography, cryptographically secure PRNGs (CSPRNGs) are preferred as they ensure stronger unpredictability.
- Security Implications: **PRNGs without sufficient entropy can be vulnerable to prediction attacks**. Cryptographically secure PRNGs (e.g., Fortuna, Yarrow) are specifically designed to be resilient against these attacks by relying on strong, entropy-rich seeds.

3. Entropy Buffer Draining

- Definition: **An entropy buffer is a storage area in an operating system or hardware where random bits (or entropy) are collected for use by PRNGs or other cryptographic functions**.
- Entropy Buffer Draining
    - When cryptographic operations or PRNGs frequently request entropy, they draw from this buffer, which can **become depleted or "drained" if there isn't enough entropy available**.
    - When the entropy buffer is drained, the system may not be able to provide true randomness, potentially falling back on less secure sources or becoming slow as it waits for the buffer to refill.

- Implications
    - Low entropy in the buffer can lead to predictable PRNG output, weakening cryptographic security.
    - Drained buffers can cause delays in applications waiting for secure random data, especially in systems that rely heavily on cryptographic operations.

4. Methods of Filling the Entropy Buffer

- **Operating System Sources**: Most operating systems **gather entropy from a variety of sources to keep the entropy buffer full**.
    - Linux: Uses **/dev/random** and **/dev/urandom** as entropy sources. **/dev/random blocks if the entropy pool is low**, while **/dev/urandom doesn't block, providing potentially less secure data when entropy is low**.
    - Windows: Uses the CryptGenRandom function, which gathers entropy from system states, events, and hardware-based random number generators.
- **Hardware Random Number Generators (HRNGs)**
    - Dedicated hardware (such as Intel's RDSEED and RDRAND instructions) provides high-quality randomness by gathering entropy directly from physical processes, like electrical noise.
- **User-Driven Entropy**
    - Random events driven by the user, such as **mouse movements, keyboard strokes, and disk activity**, can contribute to entropy.
    - User-driven entropy is often used on systems where cryptographic operations demand a significant amount of randomness, as it provides a steady source.
- **Environmental Sources**
    - Environmental factors like **timing variances, CPU temperatures, and other subtle hardware variations** can also be used to generate entropy.
    - These sources are often slow but useful as supplementary sources in entropy-starved environments.

Summary

- **Entropy**: The measure of randomness in a system, crucial for cryptographic security. Low entropy makes systems more vulnerable to attacks.
- **PRNGs**: Generate sequences of pseudo-random numbers based on a seed, which ideally comes from a high-entropy source. Secure PRNGs (CSPRNGs) are designed to be unpredictable.
- **Entropy Buffer Draining**: Occurs when cryptographic operations deplete available entropy, which can lead to slower operations and potentially less secure random data.
- **Filling the Entropy Buffer**
    - **OS Sources**: Use system events, environmental sources, and user actions.
    - **Hardware RNGs**: Provide dedicated, high-quality entropy.
    - **User Actions**: Mouse movements, keyboard strokes, and other interactions help supplement entropy.

In summary, **maintaining high entropy is critical in cryptography**. It ensures that random numbers used in key generation, session tokens, and other security-sensitive tasks are unpredictable. A robust mix of entropy sources, including hardware, user interactions, and OS-managed sources, helps ensure that entropy buffers are sufficiently filled, supporting secure cryptographic operations.