# CWE Detail – CWE-1241

## Description

The device uses an algorithm that is predictable and generates a pseudo-random number.

## Extended Description

Pseudo-random number generator algorithms are predictable because their registers have a finite number of possible states, which eventually lead to repeating patterns. As a result, pseudo-random number generators (PRNGs) can compromise their randomness or expose their internal state to various attacks, such as reverse engineering or tampering. It is highly recommended to use hardware-based true random number generators (TRNGs) to ensure the security of encryption schemes. TRNGs generate unpredictable, unbiased, and independent random numbers because they employ physical phenomena, e.g., electrical noise, as sources to generate random numbers.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2021-3692: PHP framework uses mt\_rand() function (Marsenne Twister) when generating tokens

## Related Attack Patterns (CAPEC)

* CAPEC-97

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: In many cases, the design originally defines a cryptographically secure random number generator, but is then changed during implementation due to unforeseen constraints.

## Common Consequences

**•** Impact: Read Application Data — Notes:

## Potential Mitigations

**•** Architecture and Design: A true random number generator should be specified for cryptographic algorithms. (Effectiveness: N/A)

**•** Implementation: A true random number generator should be implemented for cryptographic algorithms. (Effectiveness: N/A)

## Demonstrative Examples

**•** During the implementation phase, due to space constraint, a cryptographically secure random-number-generator could not be used, and instead of using a TRNG (True Random Number Generator), a LFSR (Linear Feedback Shift Register) is used to generate a random value. While an LFSR will provide a pseudo-random number, its entropy (measure of randomness) is insufficient for a cryptographic algorithm.

**•**

## Notes

**•** Maintenance: As of CWE 4.5, terminology related to randomness, entropy, and
 predictability can vary widely. Within the developer and other
 communities, "randomness" is used heavily. However, within
 cryptography, "entropy" is distinct, typically implied as a
 measurement. There are no commonly-used definitions, even within
 standards documents and cryptography papers. Future versions of
 CWE will attempt to define these terms and, if necessary,
 distinguish between them in ways that are appropriate for
 different communities but do not reduce the usability of CWE for
 mapping, understanding, or other scenarios.