# CWE Detail – CWE-337

## Description

A Pseudo-Random Number Generator (PRNG) is initialized from a predictable seed, such as the process ID or system time.

## Extended Description

The use of predictable seeds significantly reduces the number of possible seeds that an attacker would need to test in order to predict which random numbers will be generated by the PRNG.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2020-7010: Cloud application on Kubernetes generates passwords using a weak random number generator based on deployment time.

**•** CVE-2019-11495: server uses erlang:now() to seed the PRNG, which  
 results in a small search space for potential random  
 seeds

**•** CVE-2008-0166: The removal of a couple lines of code caused Debian's OpenSSL Package to only use the current process ID for seeding a PRNG

**•** CVE-2016-10180: Router's PIN generation is based on rand(time(0)) seeding.

**•** CVE-2018-9057: cloud provider product uses a non-cryptographically secure PRNG and seeds it with the current time

## Modes of Introduction

**•** Implementation: REALIZATION: This weakness is caused during implementation of an architectural security tactic.

## Common Consequences

**•** Impact: Varies by Context — Notes:

## Potential Mitigations

**•** N/A: Use non-predictable inputs for seed generation. (Effectiveness: N/A)

**•** Architecture and Design: Use products or modules that conform to FIPS 140-2 [REF-267] to avoid obvious entropy problems, or use the more recent FIPS 140-3 [REF-1192] if possible. (Effectiveness: N/A)

**•** Implementation: Use a PRNG that periodically re-seeds itself using input from high-quality sources, such as hardware devices with high entropy. However, do not re-seed too frequently, or else the entropy source might block. (Effectiveness: N/A)

## Applicable Platforms

**•** None (Class: Not Language-Specific, Prevalence: Undetermined)

## Demonstrative Examples

**•** An attacker can easily predict the seed used by these PRNGs, and so also predict the stream of random numbers generated. Note these examples also exhibit CWE-338 (Use of Cryptographically Weak PRNG).

## 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.