# CWE Detail – CWE-1204

## Description

The product uses a cryptographic primitive that uses an Initialization
 Vector (IV), but the product does not generate IVs that are
 sufficiently unpredictable or unique according to the expected
 cryptographic requirements for that primitive.

## Extended Description

By design, some cryptographic primitives
 (such as block ciphers) require that IVs
 must have certain properties for the
 uniqueness and/or unpredictability of an
 IV. Primitives may vary in how important
 these properties are. If these properties
 are not maintained, e.g. by a bug in the
 code, then the cryptography may be weakened
 or broken by attacking the IVs themselves.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2020-1472: ZeroLogon vulnerability - use of a static IV of all zeroes in AES-CFB8 mode (KEV)

**•** CVE-2011-3389: BEAST attack in SSL 3.0 / TLS 1.0. In CBC mode, chained initialization vectors are non-random, allowing decryption of HTTPS traffic using a chosen plaintext attack.

**•** CVE-2001-0161: wireless router does not use 6 of the 24 bits for WEP encryption, making it easier for attackers to decrypt traffic

**•** CVE-2001-0160: WEP card generates predictable IV values, making it easier for attackers to decrypt traffic

**•** CVE-2017-3225: device bootloader uses a zero initialization vector during AES-CBC

**•** CVE-2016-6485: crypto framework uses PHP rand function - which is not cryptographically secure - for an initialization vector

**•** CVE-2014-5386: encryption routine does not seed the random number generator, causing the same initialization vector to be generated repeatedly

**•** CVE-2020-5408: encryption functionality in an authentication framework uses a fixed null IV with CBC mode, allowing attackers to decrypt traffic in applications that use this functionality

**•** CVE-2017-17704: messages for a door-unlocking product use a fixed IV in CBC mode, which is the same after each restart

**•** CVE-2017-11133: application uses AES in CBC mode, but the pseudo-random secret and IV are generated using math.random, which is not cryptographically strong.

**•** CVE-2007-3528: Blowfish-CBC implementation constructs an IV where each byte is calculated modulo 8 instead of modulo 256, resulting in less than 12 bits for the effective IV length, and less than 4096 possible IV values.

## Related Attack Patterns (CAPEC)

* CAPEC-20
* CAPEC-97

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Read Application Data — Notes: If the IV is not properly initialized, data that is encrypted can be compromised and information about the data can be leaked. See [REF-1179].

## Potential Mitigations

**•** Implementation: Different cipher
 modes have different requirements for
 their IVs. When choosing and implementing
 a mode, it is important to understand
 those requirements in order to keep
 security guarantees intact. Generally, it
 is safest to generate a random IV, since
 it will be both unpredictable and have a
 very low chance of being non-unique. IVs
 do not have to be kept secret, so if
 generating duplicate IVs is a concern, a
 list of already-used IVs can be kept and
 checked against. NIST offers recommendations on generation of IVs for modes of which they have approved. These include options for when random IVs are not practical. For CBC, CFB, and OFB, see [REF-1175]; for GCM, see [REF-1178]. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** In both of these examples, the initialization vector (IV) is always a block of zeros. This makes the resulting cipher text much more predictable and susceptible to a dictionary attack.

**•** N/A

## 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.