# CWE Detail – CWE-203

## Description

The product behaves differently or sends different responses under different circumstances in a way that is observable to an unauthorized actor, which exposes security-relevant information about the state of the product, such as whether a particular operation was successful or not.

## Extended Description

Discrepancies can take many forms, and variations may be detectable in timing, control flow, communications such as replies or requests, or general behavior. These discrepancies can reveal information about the product's operation or internal state to an unauthorized actor. In some cases, discrepancies can be used by attackers to form a side channel.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2020-8695: Observable discrepancy in the RAPL interface for some Intel processors allows information disclosure.

**•** CVE-2019-14353: Crypto hardware wallet's power consumption relates to total number of pixels illuminated, creating a side channel in the USB connection that allows attackers to determine secrets displayed such as PIN numbers and passwords

**•** CVE-2019-10071: Java-oriented framework compares HMAC signatures using String.equals() instead of a constant-time algorithm, causing timing discrepancies

**•** CVE-2002-2094: This, and others, use ".." attacks and monitor error responses, so there is overlap with directory traversal.

**•** CVE-2001-1483: Enumeration of valid usernames based on inconsistent responses

**•** CVE-2001-1528: Account number enumeration via inconsistent responses.

**•** CVE-2004-2150: User enumeration via discrepancies in error messages.

**•** CVE-2005-1650: User enumeration via discrepancies in error messages.

**•** CVE-2004-0294: Bulletin Board displays different error messages when a user exists or not, which makes it easier for remote attackers to identify valid users and conduct a brute force password guessing attack.

**•** CVE-2004-0243: Operating System, when direct remote login is disabled, displays a different message if the password is correct, which allows remote attackers to guess the password via brute force methods.

**•** CVE-2002-0514: Product allows remote attackers to determine if a port is being filtered because the response packet TTL is different than the default TTL.

**•** CVE-2002-0515: Product sets a different TTL when a port is being filtered than when it is not being filtered, which allows remote attackers to identify filtered ports by comparing TTLs.

**•** CVE-2002-0208: Product modifies TCP/IP stack and ICMP error messages in unusual ways that show the product is in use.

**•** CVE-2004-2252: Behavioral infoleak by responding to SYN-FIN packets.

**•** CVE-2001-1387: Product may generate different responses than specified by the administrator, possibly leading to an information leak.

**•** CVE-2004-0778: Version control system allows remote attackers to determine the existence of arbitrary files and directories via the -X command for an alternate history file, which causes different error messages to be returned.

**•** CVE-2004-1428: FTP server generates an error message if the user name does not exist instead of prompting for a password, which allows remote attackers to determine valid usernames.

**•** CVE-2003-0078: SSL implementation does not perform a MAC computation if an incorrect block cipher padding is used, which causes an information leak (timing discrepancy) that may make it easier to launch cryptographic attacks that rely on distinguishing between padding and MAC verification errors, possibly leading to extraction of the original plaintext, aka the "Vaudenay timing attack."

**•** CVE-2000-1117: Virtual machine allows malicious web site operators to determine the existence of files on the client by measuring delays in the execution of the getSystemResource method.

**•** CVE-2003-0637: Product uses a shorter timeout for a non-existent user than a valid user, which makes it easier for remote attackers to guess usernames and conduct brute force password guessing.

**•** CVE-2003-0190: Product immediately sends an error message when a user does not exist, which allows remote attackers to determine valid usernames via a timing attack.

**•** CVE-2004-1602: FTP server responds in a different amount of time when a given username exists, which allows remote attackers to identify valid usernames by timing the server response.

**•** CVE-2005-0918: Browser allows remote attackers to determine the existence of arbitrary files by setting the src property to the target filename and using Javascript to determine if the web page immediately stops loading, which indicates whether the file exists or not.

## Related Attack Patterns (CAPEC)

* CAPEC-189

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: N/A

## Common Consequences

**•** Impact: Read Application Data, Bypass Protection Mechanism — Notes: An attacker can gain access to sensitive information about the system, including authentication information that may allow an attacker to gain access to the system.

**•** Impact: Read Application Data — Notes: When cryptographic primitives are vulnerable to side-channel-attacks, this could be used to reveal unencrypted plaintext in the worst case.

## Potential Mitigations

**•** Architecture and Design: Compartmentalize the system to have "safe" areas where trust boundaries can be unambiguously drawn. Do not allow sensitive data to go outside of the trust boundary and always be careful when interfacing with a compartment outside of the safe area. Ensure that appropriate compartmentalization is built into the system design, and the compartmentalization allows for and reinforces privilege separation functionality. Architects and designers should rely on the principle of least privilege to decide the appropriate time to use privileges and the time to drop privileges. (Effectiveness: N/A)

**•** Implementation: Ensure that error messages only contain minimal details that are useful to the intended audience and no one else. The messages need to strike the balance between being too cryptic (which can confuse users) or being too detailed (which may reveal more than intended). The messages should not reveal the methods that were used to determine the error. Attackers can use detailed information to refine or optimize their original attack, thereby increasing their chances of success. If errors must be captured in some detail, record them in log messages, but consider what could occur if the log messages can be viewed by attackers. Highly sensitive information such as passwords should never be saved to log files. Avoid inconsistent messaging that might accidentally tip off an attacker about internal state, such as whether a user account exists or not. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** In the above code, there are different messages for when an incorrect username is supplied, versus when the username is correct but the password is wrong. This difference enables a potential attacker to understand the state of the login function, and could allow an attacker to discover a valid username by trying different values until the incorrect password message is returned. In essence, this makes it easier for an attacker to obtain half of the necessary authentication credentials.

**•** When the attacker tries their own values, they can first try strings of various length. When they find a string of the right length, the computation will take a bit longer, because the for loop will run at least once. Additionally, with this code, the attacker can possibly learn one character of the password at a time, because when they guess the first character right, the computation will take longer than a wrong guesses. Such an attack can break even the most sophisticated password with a few hundred guesses.

**•** In the example above, an attacker may vary the inputs, then observe differences between processing times (since different plaintexts take different time). This could be used to infer information about the key.

**•** An attacker can recover the key by knowing if specific memory locations have been accessed or not. The value stored at those memory locations is irrelevant. The encryption routine's memory accesses will affect the state of the processor cache. If cache resources are shared across contexts, after the encryption routine completes, an attacker in different execution context can discover which memory locations the routine accessed by measuring the time it takes for their own memory accesses to complete.