# CWE Detail – CWE-779

## Description

The product logs too much information, making log files hard to process and possibly hindering recovery efforts or forensic analysis after an attack.

## Extended Description

While logging is a good practice in general, and very high levels of logging are appropriate for debugging stages of development, too much logging in a production environment might hinder a system administrator's ability to detect anomalous conditions. This can provide cover for an attacker while attempting to penetrate a system, clutter the audit trail for forensic analysis, or make it more difficult to debug problems in a production environment.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2007-0421: server records a large amount of data to the server log when it receives malformed headers

**•** CVE-2002-1154: chain: application does not restrict access to front-end for updates, which allows attacker to fill the error log

## Modes of Introduction

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

## Common Consequences

**•** Impact: DoS: Resource Consumption (CPU), DoS: Resource Consumption (Other) — Notes: Log files can become so large that they consume excessive resources, such as disk and CPU, which can hinder the performance of the system.

**•** Impact: Hide Activities — Notes: Logging too much information can make the log files of less use to forensics analysts and developers when trying to diagnose a problem or recover from an attack.

**•** Impact: Hide Activities — Notes: If system administrators are unable to effectively process log files, attempted attacks may go undetected, possibly leading to eventual system compromise.

## Potential Mitigations

**•** Architecture and Design: Suppress large numbers of duplicate log messages and replace them with periodic summaries. For example, syslog may include an entry that states "last message repeated X times" when recording repeated events. (Effectiveness: N/A)

**•** Architecture and Design: Support a maximum size for the log file that can be controlled by the administrator. If the maximum size is reached, the admin should be notified. Also, consider reducing functionality of the product. This may result in a denial-of-service to legitimate product users, but it will prevent the product from adversely impacting the entire system. (Effectiveness: N/A)

**•** Implementation: Adjust configurations appropriately when the product is transitioned from a debug state to production. (Effectiveness: N/A)

## Applicable Platforms

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