# CWE Detail – CWE-778

## Description

When a security-critical event occurs, the product either does not record the event or omits important details about the event when logging it.

## Extended Description

When security-critical events are not logged properly, such as a failed login attempt, this can make malicious behavior more difficult to detect and may hinder forensic analysis after an attack succeeds. As organizations adopt cloud storage resources, these technologies often require configuration changes to enable detailed logging information, since detailed logging can incur additional costs. This could lead to telemetry gaps in critical audit logs. For example, in Azure, the default value for logging is disabled.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2008-4315: server does not log failed authentication attempts, making it easier for attackers to perform brute force password guessing without being detected

**•** CVE-2008-1203: admin interface does not log failed authentication attempts, making it easier for attackers to perform brute force password guessing without being detected

**•** CVE-2007-3730: default configuration for POP server does not log source IP or username for login attempts

**•** CVE-2007-1225: proxy does not log requests without "http://" in the URL, allowing web surfers to access restricted web content without detection

**•** CVE-2003-1566: web server does not log requests for a non-standard request type

## Modes of Introduction

**•** Operation: COMMISSION: This weakness refers to an incorrect design related to an architectural security tactic.

## Common Consequences

**•** Impact: Hide Activities — Notes: If security critical information is not recorded, there will be no trail for forensic analysis and discovering the cause of problems or the source of attacks may become more difficult or impossible.

## Potential Mitigations

**•** Architecture and Design: Use a centralized logging mechanism that supports multiple levels of detail. (Effectiveness: N/A)

**•** Implementation: Ensure that all security-related successes and failures can be logged. When storing data in the cloud (e.g., AWS S3 buckets, Azure blobs, Google Cloud Storage, etc.), use the provider's controls to enable and capture detailed logging information. (Effectiveness: N/A)

**•** Operation: Be sure to set the level of logging appropriately in a production environment. Sufficient data should be logged to enable system administrators to detect attacks, diagnose errors, and recover from attacks. At the same time, logging too much data (CWE-779) can cause the same problems, including unexpected costs when using a cloud environment. (Effectiveness: N/A)

**•** Operation: To enable storage logging using Azure's Portal, navigate to the name of the Storage Account, locate Monitoring (CLASSIC) section, and select Diagnostic settings (classic). For each of the various properties (blob, file, table, queue), ensure the status is properly set for the desired logging data. If using PowerShell, the Set-AzStorageServiceLoggingProperty command could be called using appropriate -ServiceType, -LoggingOperations, and -RetentionDays arguments. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** The previous configuration file has effectively disabled the recording of security-critical events, which would force the administrator to look to other sources during debug or recovery efforts.

**•** It is recommended to log the failed login action. Note that unneutralized usernames should not be part of the log message, and passwords should never be part of the log message.

**•** The "--log d" portion of the command says to log deletes. However, the argument does not include the logging of writes and reads. Adding the "rw" arguments to the -log parameter will fix the issue: