CRASHTEST **SECURITY**

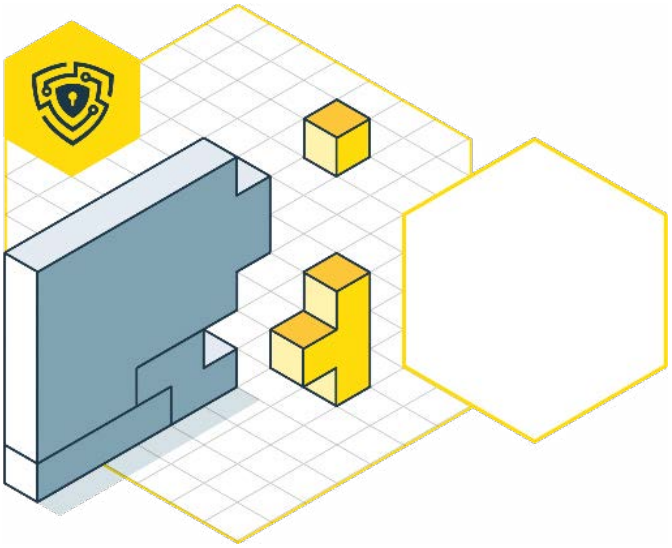
GUIDE FOR

SECURITY LOGGING AND MONITORING FAILURES PREVENTION

WHAT ARE THE STEPS TO KEEP YOUR WEB APP OR API SAFE FROM SUCH VULNERABILITY

GUIDE TO
SECURITY LOGGING AND MONITORING FAILURES PREVENTION

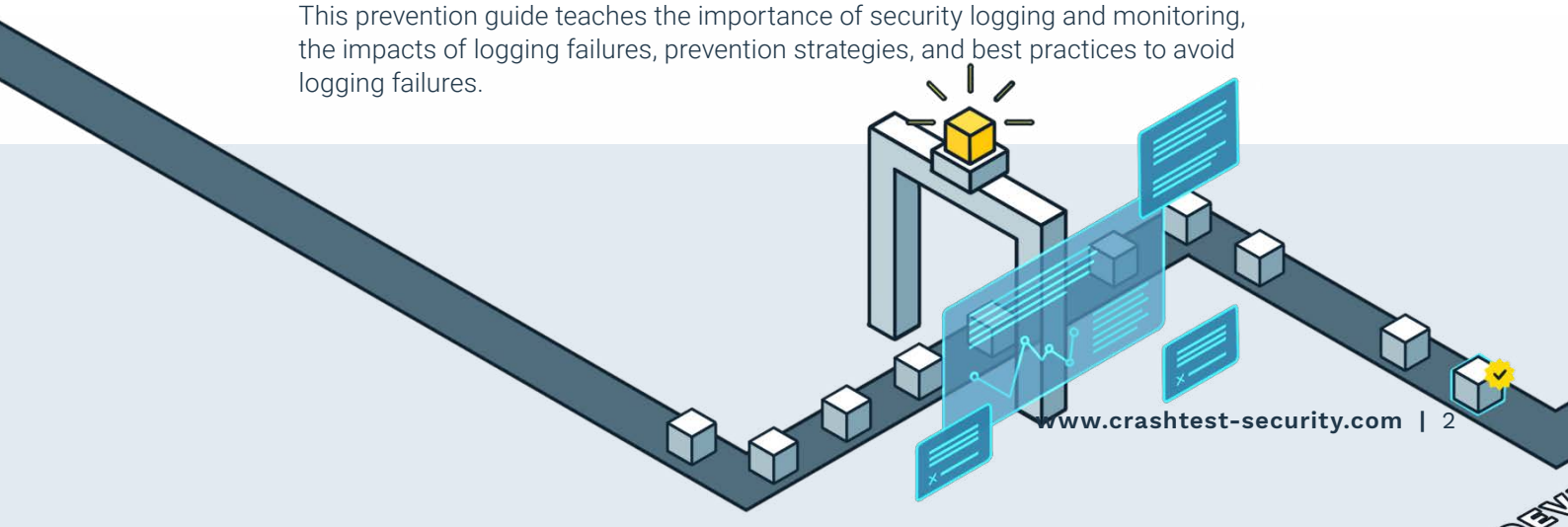
Table of Contents



INTRODUCTION TO THIS GUIDE

Logging is essential to modern application delivery as it helps track data and events throughout an application lifecycle. Logs are categorized into various log formats to help segregate records based on the data they capture. A common approach is to use access logs to contain information about who accessed what resources and when, application logs for information about the application, such as errors and warnings. In contrast, system logs have information about the underlying operating system and hardware, such as boot messages and resource utilization. Logs also offer resourceful insights to detect intrusions, monitor user activity, and track changes to critical files. By regularly reviewing log data, businesses can proactively address potential security threats. While logs are meant to offer granular details, an improper logging mechanism restricts issue resolution and mitigating security attacks. This is often because, without efficiently administered logging, it is difficult to track down and debug errors, understand user behavior, or even know if your system is functioning correctly.

This prevention guide teaches the importance of security logging and monitoring, the impacts of logging failures, prevention strategies, and best practices to avoid logging failures.



WHAT ARE SECURITY LOGGING AND MONITORING?

Although logging and monitoring are two distinct approaches in application delivery, they are often used as critical mechanisms that enforce security by helping software teams identify abnormal patterns and system misconfigurations. Application logs effectively identify attack vectors and correlate them with security events. On the other hand, monitoring enables the automatic, continuous assessment of vulnerabilities by collecting and analyzing key metrics.

Insufficient logging and monitoring are caused when an organization fails to properly configure mechanisms for detecting and identifying security risks. However, not a direct security vulnerability, logging and monitoring failures allow malicious actors to persist attack patterns while remaining unnoticed since the organization lacks indicators of a security breach.

Typical incorrect practices that lead to security logging and monitoring failures include:

- Failure to perform comprehensive security logging for audible events such as access control failures, input validation failures, and unauthorized access of sensitive information, among others.
- Security events that generate unclear, inadequate, or no error messages
- Lack of monitoring application logs for suspicious activity
- Local storage of log files
- Ineffective alerting thresholds and incident response plans
- Failure to detect, escalate and alert for active attacks in real-time

SECURITY LOGGING AND MONITORING FAILURES - SEVERITY LEVEL

The security logging and monitoring failure vulnerability ranks **#9** on the OWASP Top 10 List of 2021. The exposure has a maximum **incidence rate of 6.51%** and an **average weighted exploit of 6.87**. As such vulnerabilities are difficult to detect without direct access to internal networks, insufficient logging and monitoring failures are typically challenging to exploit for a successful attack.

Insufficient logging and monitoring vulnerabilities have an **average coverage of 39.97%** and an **average weighted impact score of 4.99**. Impacts of a successful attack over insufficient logging and monitoring failures include:

- **BotNet attacks** - Monitoring systems track user behavior to identify a pattern based on the origin, duration, and frequency of requests. Lack of monitoring makes it hard to distinguish between human and non-human network traffic, making it easy to build bots that can overwhelm the application server with requests.

- **File-based attacks** - Attackers can extract file paths and other sensitive information from logs to access restricted information and perform advanced attacks. For instance, server log files provide critical information on the application that writes to the file. This can provide full directory names, files, and system information, which attackers can exploit for code injection and other malicious attacks.
- **System unavailability/Denial of Service** - Attackers can flood a server with requests without appropriate logging and monitoring mechanisms until it crashes or fails to respond to requests. This makes the application's internal services inaccessible to users, impacting the organization's reputation and revenues.
- **Advanced persistent threats** - Logging and monitoring failures make it difficult for security teams to detect and identify active attacks. This allows attackers to compromise the system for a longer time, expanding the scope and impact of the attack.

Logging and monitoring failures are also mapped to four common weaknesses and enumerations. These are:

CWE-778: Insufficient Logging

Insufficient logging weakness is caused when an application fails to record security events or if the log record is missing crucial details. This makes suspicious behavior difficult to detect, consequently hindering the root cause analysis after a successful attack.

CWE-117: Improper Output Neutralization for Logs

This flaw occurs when the application fails to neutralize outputs written to audit logs appropriately. In such instances, threat actors can insert special characters, such as line-ending characters, into data that is written into application logs. Unexpected characters in log entries also cause log-parsing issues, which attackers further target to compromise log management systems.

CWE-223: Omission of Security-relevant Information

An insecure design flaw is caused when the application fails to record or display data that would help to identify the source and nature of cyber attacks. Examples of omission of relevant security information include:

Login attempts are not recorded if the user disconnects before the maximum number of tries
Sender's IP and email address are not recorded in the outgoing email
Failed authentication attempt not recorded if later attempt succeeds.

CWE-532: Insertion of Sensitive Information into Log File

This flaw occurs when the information recorded in log files is sensitive and can be exposed in a data breach or guide the attacker through a successful attack. An application could write user credentials, session data, or system information into the log file, so anyone accessing logs can read this information. Log files provide an additional, less-secure path for hackers to acquire information needed for an exploit.

IDENTIFY SECURITY LOGGING AND MONITORING FAILURES WITH CRASHTEST SECURITY

Crashtest Security helps eliminate security blind spots that can lead to logging and monitoring failures through automated vulnerability scanning and penetration testing. Crashtest Security offers a suite of automated vulnerability scanners to detect attack vectors across different APIs, web applications, and JavaScript layers.

The platform also outputs actionable security reports that offer comprehensive details of discovered application security risks, including the level of vulnerability severity, impact, and likelihood of exploitation. Security teams and developers can correlate this information with application logs to perform contextual assessments for maintaining a robust security posture.

Try Crashtest Security for a free, 14-day demo to know how the platform can help administer effective monitoring and logging mechanisms for enhanced security.

SECURITY LOGGING AND MONITORING PREVENTION TECHNIQUES

Security measures to prevent logging and monitoring failures include:

SECURITY INFORMATION AND EVENT MANAGEMENT (SIEM)

SIEM platforms collect audit logs from several resources, identify any suspicious behavior and take appropriate action. By collecting and analyzing security-relevant information, SIEM tools help security teams recognize potential threats before they can misuse the organization‘s internal services. Besides efficient logging, SIEM tools can also be used for event correlation and monitoring for incident response. These systems are recommended to be used to enhance existing cybersecurity plans through the detection of malicious activities, user behavior monitoring, compliance report generation, and enforcement of access controls.

AUDIT TRAILS

An audit trail is a detailed, chronological record showing application events‘ date, time, and destination/source. Audit trails enable comprehensive security assessments since they provide help with a contextual correlation of event cycles for early detection and mitigation of attack vectors. Beyond insufficient logging and monitoring flaws, audit trails also enforce compliance with industry regulations that require verifying and validating auditable events.

INCIDENT RESPONSE AND RECOVERY PLANS

An incident response and resolution plan operate on bench-marked standards for identifying security events and quickly mitigating risks. An incident response and recovery plan strengthen logging and monitoring systems by providing a standard format for reporting incidents while enabling security teams to react quickly to active attacks.

BEST PRACTICES IN PREVENTING SECURITY LOGGING AND MONITORING FAILURES

Best practices to prevent security logging and monitoring failures include:

ENFORCE MULTI-FACTOR AUTHENTICATION (MFA)

Organizations should consider implementing MFA for all users, particularly those with privileged accounts with access to critical data or internal services. In addition, MFA should be enforced for specific administrative actions, such as creating or modifying firewall rules or accessing customer data. Enforcing MFA helps ensure that only authorized individuals can access security-relevant information of log entries, which further restricts attackers from orchestrating DNS and file-based attacks.

ENCODE LOG DATA

Proper encoding of log data prevents attackers from compromising log management systems through malicious code injection attacks and other exploits that target log files. Log injection attacks often lead to the entry of malicious code and the forging of bogus log events. As encoded log outputs restrict attackers from accessing or altering log data, the practice helps preserve the integrity of the security and event information collected.

USE STANDARD LOG FORMATS

Standard log formats, such as Common Event Format (CEF), help ensure that logs are uniformly readable and understandable by machines. The approach makes it easier to aggregate, search, and analyze log data. Additionally, standard log formats can help prevent logging and monitoring failures by ensuring that logs are properly formatted and, therefore, usable.

IMPLEMENT EFFECTIVE LOG MONITORING WITH CRASHTEST SECURITY

Crashtest Security Suite enables organizations to boost their functional logging and monitoring efforts through automated security checks. The platform integrates seamlessly with most modern frameworks and can be set up to initiate scans within minutes.

To know more about how Crashtest Security can scan logging and monitoring misconfigurations, try a free, 14-day demo here.

Start 2-Week Trial for Free

CRASHTEST **SECURITY**

WWW.CRASHTEST-SECURITY.COM