

Log Montitoring Procedure

Mapped to PCI-DSS v3.2.1

**Document Key Information:**

|  |  |  |
| --- | --- | --- |
| **Title** | : | Log Monitoring Procedure |
| **Topic** | : | Information Security Management System |
| **Classification** | : | Procedure |
| **Status** | : | Draft |
| **Owner Department** | : | Cyber and Information Security Department |
| **Document Approver** | : | Board of Directors |
| **Applicability** | : | [Company Name] |
| **Version** | : | 1.0 |
| **Last Review Date** | : |  |
| **Next Review Date** | : |  |

**Document Change Control:**

To track updates and changes related to the document:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version:** | **Date:** | **Change Description Summary** | **Changed By** |
| 1.0 | xx/xx/xxxx | Initial Version | CISO |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 5](#_Toc114148709)

[2 Scope 5](#_Toc114148710)

[3 Deviations and Waivers 5](#_Toc114148711)

[4 Non-Compliance 5](#_Toc114148712)

[5 Log-Monitoring Requirements in PCI DSS 6](#_Toc114148713)

[5.1 Key Terms 6](#_Toc114148714)

[5.2 Log Monitroing Requirements 7](#_Toc114148715)

[5.3 Other Important PCI DSS Requirements Related to Log Monitoring 9](#_Toc114148716)

[5.4 Section Summary 10](#_Toc114148717)

[6 Planning for Effective Log Monitoring 11](#_Toc114148718)

[6.1 Identify and Map System-Level Event Messages to High-Level Events 13](#_Toc114148719)

[6.2 Prioritize Log Sources 13](#_Toc114148720)

[6.3 Notification and escalation contacts 14](#_Toc114148721)

[6.4 Response to Security Events 14](#_Toc114148722)

[7 Preparing for Effective Log Monitoring 15](#_Toc114148723)

[7.1 Identify the Tools & Resources to be Used for Log Management 15](#_Toc114148724)

[7.2 Establish Central Repository for Log Data 15](#_Toc114148725)

[7.3 Transport Logs to the Centralized Repository 16](#_Toc114148726)

[7.4 Prepare Log Data for Processing 16](#_Toc114148727)

[8 Performing Effective Log Monitoring 17](#_Toc114148728)

[8.1 Collect and Analyze Activity Data 17](#_Toc114148729)

[8.2 Establish a Baseline 18](#_Toc114148730)

[8.3 Configure Automated Alerts 19](#_Toc114148731)

[8.4 Respond to Alerts 19](#_Toc114148732)

[8.5 Validate Events 20](#_Toc114148733)

[8.6 Respond to Incidents 20](#_Toc114148734)

[8.7 Collect and Analyze Incident Data 20](#_Toc114148735)

[8.8 Report on Results 21](#_Toc114148736)

[8.9 Perform Periodic Program Reviews 21](#_Toc114148737)

[9 Applying Effective Log Monitoring 22](#_Toc114148738)

[9.1 Business-as-Usual Activities 22](#_Toc114148739)

[10 Log Monitoring RACI Matrix 24](#_Toc114148740)

[11 Appendix A: List of Mandatory Alerts 25](#_Toc114148741)

# Introduction

For security logs to be useful in defence of information assets, they must be monitored and analyzed—in as close to real-time as possible—so that attacks can be detected quickly and appropriate countermeasures deployed to augment existing defences when and where necessary. This becomes increasingly important as attacks and attackers become more sophisticated.

Without the active monitoring and analysis of security logs, the erosion of information security defences by capable adversaries will likely go undetected and will eventually result in the compromise of the very assets that require protection.

This document should be read in conjunction with [COMPANY NAME] Information Security Policy.

# Scope

This procedure applies to all:

* [COMPANY NAME] units and cloud environments.
* Security solutions logs such as (Antivirus, EDR, Firewall, IPS, etc).
* IT infrastructure logs including Operating system, Databases, Network and Telecom devices.
* Application logs that are required to be clooected to comply with PCI standards or local regulator.

# Deviations and Waivers

If a Unit or Department does not follow or comply with any of the requirements detailed within this procedure, a prior waiver must be sought following the approved Group Standard for Dispensation.

# Non-Compliance

Employees must report any suspected or confirmed violations of this procedure to Cyber and Information Security. Actions that are considered to be non-compliant or a breach of [COMPANY NAME] security policies and procedures will be reviewed on an individual basis. Breaches may result in disciplinary action being undertaken and may lead to termination of employment or criminal charges.

# Log-Monitoring Requirements in PCI DSS

The Payment Card Industry Data Security Standard (PCI DSS) is based on the concept of defence in-depth and includes a variety of preventive, detective, and corrective information security measures (also called “security controls”). Moreover, PCI DSS includes requirements devoted to the use of log monitoring in the ongoing protection of information assets, addressing the need for proactive monitoring of security logs in Requirement 10.6:

***10.6*** *Review logs and security events for all system components to identify anomalies or suspicious activity*

The key elements of PCI DSS Requirement 10.6 are listed in the following three sub-requirements:

|  |
| --- |
| * + 1. *Review the following at least daily:*
			- *All security events*
			- *Logs of all system components that store, process, or transmit CHD and/or SAD*
			- *Logs of all critical system components*
			- *Logs of all servers and system components that perform security functions (for example, firewalls, intrusion-detection systems/intrusion-prevention systems (IDS/IPS), authentication servers, e-commerce redirection servers, etc.)*
 |
| ***10.6.2*** *Review logs from all other system components periodically based on the organization’s policies and risk management strategy, as determined by the organization’s annual risk assessment.* |
| ***10.6.3*** *Follow up exceptions and anomalies identified during the review process* |

## Key Terms

#### Security Event

The security event is “an occurrence considered by an organization to have potential security implications to a system or its environment. In the context of PCI DSS, security events identify suspicious or anomalous activity.” Examples of security events include attempted logons by non- existent user accounts, excessive password authentication failures, or the startup or shutdown of sensitive system processes. Unfortunately, determining the specific types of events and activities that should constitute security events is largely dependent on each individual environment, the systems resident in that environment, and the business processes served by that environment.

PCI DSS provides some insight into those activities that might constitute a security event by defining seven high-level activities that must be tracked:

|  |
| --- |
| ***10.2.1*** *All individual user accesses to cardholder data* |
| ***10.2.2*** *All actions taken by any individual with root or administrative privileges* |
| ***10.2.3*** *Access to all audit trails* |
| ***10.2.4*** *Invalid logical access attempts* |
| ***10.2.5*** *Use of and changes to identification and authentication mechanisms—including but not limited to creation of new accounts and elevation of privileges—and all changes, additions, or deletions to accounts with root or administrative privileges* |
| ***10.2.6*** *Initialization, stopping, or pausing of the audit logs* |
| ***10.2.7*** *Creation and deletion of system-level objects* |

Any of the activities described above that are performed without proper authorization would likely constitute a security event. For example, unauthorized individuals accessing cardholder data, audit trails, or making modifications to user settings or system-level objects might reflect an unauthorized elevation of privileges.

Similarly, the frequent and successive occurrence of invalid logical access attempts might also reflect an attempt to brute force passwords.

#### System Components

In this document, we make numerous references to “information systems” or “systems.” It is important to point out that an information system may consist of multiple system components. A system component, however, is not necessarily limited to a physical asset—such as a server or a network device. Software applications are also components of information systems. The system components as “any network devices, servers, computing devices, or applications included in or connected to the cardholder data environment.”

#### Critical System Components

“Critical system components” are those system components that perform functions either vital to the operation or security of an information system or the cardholder data environment, or those components that—if compromised—could result in significant damages (whether financial or reputational) to the company. Examples of critical system components might include databases containing cardholder data, unified storage systems, network authentication systems, or cryptographic systems—such as hardware security modules (HSMs). The system components the company deems “critical” will likely depend on how the company analyzes risk and assigns risk ratings to specific systems.

## Log Monitroing Requirements

#### Requirement 10.6.1

PCI DSS Requirement 10.6.1 provides the foundation for the proactive monitoring of security logs for the occurrence of security events by requiring “daily” reviews of logs for critical system components. The security logs must be reviewed as close to real-time as possible so that attacks can be detected and countermeasures deployed *before* a data breach occurs. If real-time monitoring is not doable, a reasonable timeline must be defined to perform security log reviews while still enabling the company to detect malicious or anomalous activity before it can likely escalate. In the case of Requirement 10.6.1, PCI DSS has determined that timeline to be a maximum of 24 hours or one calendar day.

The note in Requirement 10.6 provides some examples of such automated tools that may be used to meet the intent of this requirement as well as the other sub-requirements under Requirement 10.6:

|  |
| --- |
| ***10.6*** *Review logs and security events for all system components to identify anomalies or suspicious behavior.* |
| ***Note:*** *Log harvesting, parsing, and alerting tools may be used to meet this Requirement.* |

In addition to specifying that “daily” log reviews must be performed, PCI DSS Requirement 10.6.1 identifies what must be reviewed on a daily basis. The best way to clarify what must be reviewed daily is to look at each bullet individually. In the first bullet, the term “security event” is meant to represent those pre-defined events and activities that the company has identified as being potentially “malicious or anomalous,” including those specified in Requirements 10.2.1 through 10.2.7. Consequently, Requirement 10.6.1 requires the review of all security events on a daily basis.

In many environments, “critical system components” might include “all system components that store, process, or transmit CHD and/or SAD.” The same thing is true for “all system components that perform security functions.” The intent of specifying the logs that must be reviewed in this manner is to ensure that regardless of how the company defines critical system components, the company includes logs from such systems in the daily log review.

#### Requirement 10.6.2

Requirement 10.6.2 is complementary to Requirement 10.6.1 and covers all of the other system component logs that are not addressed in Requirement 10.6.1. Like Requirement 10.6.1, the main focus of Requirement

10.6.2 is to review logs for all “security events.” However, the frequency with which all other system component logs are to be reviewed is left up to the company itself to define. Remember, a “system component” is a term used in PCI DSS to denote something that resides in or is connected to the cardholder data environment. Therefore, Requirement 10.6.2 is intended to cover all of the other “in-scope” systems. Systems that are neither located in nor connected to the cardholder data environment—or otherwise are not defined in [COMPANY NAME] PCI DSS scope—are not expected to be reviewed in accordance with Requirement 10.6.2.

PCI DSS Requirement 10.6.2 introduced to provide organizations more flexibility in performing log reviews, by allowing them to define how often logs are reviewed for systems that do not fall under Requirement 10.6.1. This does not absolve organizations from having to review other “less- critical” system logs—as noted in Requirement 10.6. It simply allows the organization to focus efforts on the highest-risk systems first.

#### Requirement 10.6.3

Requirement 10.6.3 is one of the most important requirements in all PCI DSS for the ongoing protection of cardholder data, and is an often-overlooked element of log-monitoring processes. It requires follow-up on all exceptions and anomalies identified during the review processes identified in both Requirement 10.6.1 and 10.6.2. When security events are detected in system component logs, it is essential that those events be investigated further to determine whether the occurrence of each event actually represents something malicious or anomalous.

Deciding whether an event represents normal user activity or potentially malicious activity requires analysis of the context in which the event occurred. The occurrence of an event must be verified to represent something concerning. This is called “event validation.”

Therefore, the context in which a security event occurs must be evaluated further to help organizations differentiate between normal and abnormal activity. This is what Requirement 10.6.3 is intended for.

## Other Important PCI DSS Requirements Related to Log Monitoring

#### Requirement 12.10

The whole point of the proactive monitoring of security logs is to facilitate timely response to potentially malicious activities before they become a big problem. How (and how quickly) the company responds to a confirmed malicious event may be the difference between a minor security event and a major breach of cardholder data. [COMPANY NAME] must be prepared for such instances, and have appropriate response procedures and countermeasures prepared—in advance—to respond in a timely and efficient manner. PCI DSS Requirement 12.10 is intended to facilitate the advance planning of such incident response procedures.

***12.10*** *Implement an incident response plan. Be prepared to respond immediately to a system breach.*

Incident response planning is a critical component of any defence-in-depth strategy. Unfortunately, providing guidance on defining incident response procedures would likely command its own Special Interest Group (SIG) and is, therefore, a bit beyond the scope of this document.

#### Requirement 12.5.2

***12.5.2*** *Monitor and analyze security alerts and information, and distribute to appropriate personnel.*

Someone needs to be assigned responsibility for monitoring and analyzing security alerts. Requirement.

12.5.2 is intended to ensure those responsibilities are formally assigned. In addition, Requirement 12.5.2 is intended to facilitate the incident response process in the event suspicious or abnormal behaviour is detected by ensuring appropriate individuals are notified when such events occur.

The “monitoring” is “the use of systems or processes that constantly oversee computer or network resources for the purposes of alerting personnel in the case of outages, alarms, or other predefined events.” Once an event has been confirmed to represent concerning or even known malicious behaviour, it is the responsibility of log-monitoring personnel (as illustrated in the definition above) to alert the appropriate individuals so that suitable countermeasures can be deployed. Whether those individuals are other system or security administrators, management personnel, legal representatives, or even the payment brands will depend on the context and extent to which the malicious activity has occurred. But there needs to be a well-defined “handoff” of responsibility between the log-monitoring function and the incident response function.

Although the intent here is to distinguish between the log-monitoring function and the incident response function, these need not be different individuals or teams. It is entirely possible—even likely in many scenarios—that log-monitoring personnel will be assigned responsibility for incident response in addition to their monitoring duties. The key point here is that in the case of a confirmed malicious event, there will be some point or “trigger” at which time the decision to initiate some type of response will become necessary.

#### Requirement 12.10.3

***12.10.3*** *Designate specific personnel to be available on a 24/7 basis to respond to alerts.*

Attackers do not operate on any particular schedule. Attacks and intrusions can occur at any time. Therefore, it is necessary for the company to have assigned personnel to respond to attacks 24-hours a day, 7 days a week. Requirement 12.10.3 ensures that personnel are available 24/7 to respond to security events and to initiate formal response procedures when required.

#### Requirement 12.10.5

***12.10.5*** *Include alerts from security monitoring systems, including but not limited to intrusion-detection, intrusion-prevention, firewalls, and file-integrity monitoring systems.*

The occurrence of malicious activity does not necessarily mean a data breach has occurred, but formal response procedures are still necessary to ensure a consistent and appropriate response to such instances. Requirement 12.10 is intended to cover multiple scenarios, up to and including a confirmed breach of cardholder data.

Determining whether certain elements of an incident response plan require invocation or whether certain individuals need to be notified will, depend on the context and extent to which malicious activity has occurred. PCI DSS Requirement 12.10.5 is intended to ensure that, in addition to defining processes for responding to a confirmed data breach, incident response procedures2 also include guidance for responding to “lesser” events, such as instances of known malicious activity.

## Section Summary

[COMPANY NAME] recognizes the importance of proactive monitoring of security logs in the detection of attacks on information assets and the protection of those assets from compromise. PCI DSS Requirements for log monitoring are covered under Requirement 10.6. Requirement 10.6 and its sub-requirements call for the review of system security logs to identify the occurrence of security events. Requirement 10.6.1 specifies daily log reviews for critical systems, while 10.6.2 permits more infrequent log reviews for other less-critical systems. Requirement 10.6.3 then stipulates that detection of any security event be investigated further to confirm the occurrence of known malicious activity. Upon detection of known malicious activity, formal response procedures are necessary to respond to such events and to mitigate any further damages.

Requirements 12.5.2 and 12.10.3 ensure that appropriate personnel are assigned responsibility for monitoring alerts and responding to security events on a 24/7 basis. Finally, Requirement 12.10.5 ensures that incident response procedures include guidance on handling instances of known malicious behaviour. These requirements are intended to define a framework for the timely detection of potentially malicious behaviour and the incident response procedures that should be performed to protect cardholder data in the event of an attack.

# Planning for Effective Log Monitoring

Effective log-monitoring practices start with effective planning of log-monitoring needs and activities. To ensure alignment of [COMPANY NAME] with effective log-monitoring (and to meet the intent of PCI DSS Requirements for log monitoring), [COMPANY NAME] must have a thorough understanding of its legal, regulatory, business, and operational requirements. In addition, it must understand the technical capabilities of the systems that need to be monitored, the technologies available to assist with monitoring processes, and the technical capabilities of other individuals and teams within the company who can assist in developing effective and efficient log- monitoring practices. The Security Information and Event Management (SIEM) industry often refer to these activities as “events of interest.”

[COMPANY NAME] is specific when it comes to defining the basic high-level activities that must be logged and tracked and lists the below PCI DSS Requirements 10.2.1 through 10.2.7:

|  |
| --- |
| ***10.2.1*** *All individual user accesses to cardholder data* |
| ***10.2.2*** *All actions taken by any individual with root or administrative privileges* |
| ***10.2.3*** *Access to all audit trails* |
| ***10.2.4*** *Invalid logical access attempts* |
| ***10.2.5*** *Use of and changes to identification and authentication mechanisms—including but not limited to creation of new accounts and elevation of privileges—and all changes, additions, or deletions to accounts with root or administrative privileges* |
| ***10.2.6*** *Initialization, stopping, or pausing of the audit logs* |
| ***10.2.7*** *Creation and deletion of system-level objects* |

In addition to the events defined above, PCI DSS identifies other activities that should be logged and tracked. Unfortunately, these are not always explicitly defined in the actual requirement language. For instance, Requirement 5.2 calls for all anti-virus mechanisms to “generate audit logs,” but the language in the requirement does not identify the activities that must be captured in those logs. It is in the guidance for this requirement where the intent to log and track all “virus and malware activity and anti-malware reactions” is defined. Events from IDS/IPS systems (Requirement 11.4) and file-integrity monitoring (FIM) systems (Requirement 11.5) are also expected to be logged and tracked. In the case of Requirement 11.5, any modification (changes, additions, and deletions) of critical system files, configuration files, or content files should be monitored.

There may be other events of interest that [COMPANY NAME] wish to define in addition to those specified in PCI DSS requirements. Examples of such events may include the detection of active malware on a web server, the presence of CHD in an unauthorized location, or multiple attempts to connect to a database server containing CHD from an unauthorized source (such as an external IP address).

For the purposes of PCI DSS compliance, the logs should capture the key data points specified within PCI DSS Requirement 10.3 at a minimum:

|  |
| --- |
| ***10.3*** *Record at least the following audit trail entries for all system components for each event:* |
| ***10.3.1*** *User identification* |
| ***10.3.2*** *Type of event* |
| ***10.3.3*** *Date and time* |
| ***10.3.4*** *Success or failure indication* |
| ***10.3.5*** *Origination of event* |
| ***10.3.6*** *Identify or name of affected data, system component, or resource* |

The actual data fields in each specific log may differ from what is specified in Requirement 10.3. However, most logs should have data fields that generally map to the data points defined in Requirement 10.3. For example, most logs have fields for Source IP and Destination IP. These fields generally map to data points specified in Requirements 10.3.5 and 10.3.6 respectively. The same goes for Timestamp fields and Requirement 10.3.3. Some logs, however, may not have specific fields to map back to those data points in Requirement 10.3. However, that does not mean the information is not available.

That information may be obtained in other fields. One common example involves Success or failure indication in Requirement 10.3.4. Many logs do not have a specific “status” field. However, this information is often provided as part of a System Message field, which may contain information on multiple data points, including the type of event, success or failure, the specific user ID involved, etc. [COMPANY NAME] need to evaluate each log type to determine what information and data points are available.

At a minimum, the following characteristics are captured for each log type:

|  |  |
| --- | --- |
| * Log Name
 | * Application Name (which generates the logs)
 |
| * Description
 | * Application Version
 |
| * Location (file, path, database table, etc.)
 | * Operating System
 |
| * Log Type / Format (text, XML, etc.)
 | * Relevant Data Points
 |

## Identify and Map System-Level Event Messages to High-Level Events

For each of the high-level events of interest, [COMPANY NAME] need to investigate the logs from each log source type to determine how the logs identify those same events within system-level log messages.

This may require [COMPANY NAME] to analyze logs from a range of dates in order to identify the relevant system-level messages. In case of using a third-party tool such as a SIEM solution, many SIEM tools come pre-installed with templates that already map certain system-level events from many common log sources (such as network equipment, operating systems, databases, etc.) to common events of interest, such as “failed user logons” and “user account changes.” Therefore, there may only be limited manual analysis that is required to map system-level events from sources not supported by the SIEM.

[COMPANY NAME] should document the mapping between the high-level events and the corresponding system-level messages for those log source types unique to [COMPANY NAME] environment. For example:

|  |  |  |
| --- | --- | --- |
| **Event Type** | **Message Summary** | **Event Source** |
| **Failed User Logon** | *An account failed to failed logon* | Microsoft Windows |
| *Event ID = 4025* | Microsoft Windows |
| *Authentication failure* | Linux |
| *Failed user authentication* | Microsoft IIS |

Documenting the mapping between high-level events of interest and system-level messages provides other log-monitoring personnel with a valuable reference should new systems be added or upgraded, or when scripts need to be modified to support a new log source or log source type.

## Prioritize Log Sources

[COMPANY NAME] need to focus first on those systems and system components that are the most important and represent the most risk to the company and stakeholders. PCI DSS Requirement 10.6.1 provides some direction in determining the most critical system components by identifying the types of system components that require daily log reviews: “All system components that store, process, or transmit CHD and/or SAD” should probably be considered among the most important. The same likely goes for “all servers and system components that perform security functions (for example, firewalls, intrusion-detection systems/intrusion-prevention systems (IDS/IPS), authentication servers, e-commerce redirection servers, etc.).” [COMPANY NAME] should evaluate the results of their last risk assessment (as specified in PCI DSS

Requirement 12.2).

System component inventories (Requirement 2.4), and network and cardholder data flow diagrams (Requirements 1.1.2 and 1.1.3, respectively) may also help identify those system components that meet the criteria specified above. For all other in-scope systems, [COMPANY NAME] should define criticality based on its risk assessment framework.

Careful consideration for each system component and the relationship it has with other system components is essential in evaluating potential threats to information assets. Additionally, system and system component classifications and justification for those classifications should be documented. One logical place to do so might be in the system component inventories described in Requirement 2.4. Having documented justification for all system classifications can help defend any decisions to exclude certain systems from daily log review requirements.

## Notification and escalation contacts

Once [COMPANY NAME] have defined the key events of interest, and have identified and prioritized the systems it wish to monitor, [COMPANY NAME] need to determine which individuals and/or teams should be notified when such events occur. PCI DSS Requirement 12.5.2 ensures that appropriate individuals are formally assigned responsibility for monitoring and analyzing security alerts and information, and Requirement 12.10.3 ensures such personnel are made available on a 24/7 basis. In many cases the personnel that are assigned responsibility for responding to alerts will depend on the specific event, and/or the system(s) on which the event(s) occurred.

For each of the events identified by [COMPANY NAME] and those defined in Requirement 10.2, consider the potential impact of that event occurring on each system and system component. Consideration may need to be given to each individual system’s classification, as those resources that require notification may differ from system to system based on the criticality of that system. Additional considerations may need to be given to the “owner” of the system (business owner vs. administrative personnel), the time the event occurs (and whether or not separate “after hours” personnel should be notified), and the location of the system (and whether local or remote personnel require notification).

## Response to Security Events

The response procedures for each high-level event should also be specified (see Requirement 10.6.3 and 12.10.5). This will depend largely on the event, the type of system the event occurred on, and the criticality of that system. Likewise, response procedures may also depend on other factors, including the time of day the event occurred as well as the location of the event. Response procedures need not be highly detailed, as a general description of the procedures should suffice. However, the more detailed the procedure, the less likelihood responsible personnel may misunderstand those procedures. So it may be in the company’s best interest to be as detailed as possible.

# Preparing for Effective Log Monitoring

Another key component of an effective log-monitoring program is effective log-management practices. [COMPANY NAME] need to define the processes, tools, and infrastructure that will be used to manage logs in accordance with those requirements.

## Identify the Tools & Resources to be Used for Log Management

Although it’s not required by PCI DSS, it is becoming increasingly infeasible to be effective at log monitoring and management without some form of automation. Whether using scripts, native alerting mechanisms, or third-party Log Management, SIEM, or Advanced Analysis solutions, some level of automation is necessary to process, transmit, analyze, and alert on security event information. Even in small environments, the speed and volume with which log data is generated today makes manual processing and analysis completely impractical.

[COMPANY NAME] need to assess the tools and resources and depending upon the log-monitoring needs and implement the best solution. Or, outsource the service to a trusted capable SIEM/SOC provide.

## Establish Central Repository for Log Data

As per PCI DSS Requirement 10.7, it is often necessary to retain log data for historical purposes:

***10.7*** *Retain audit trail history for at least one year, with a minimum of three months immediately available for analysis (for example, online, archived, or restorable from backup).*

PCI DSS Requirement 10.5 and its sub-requirements describe the methods that must be employed to protect logs:

|  |
| --- |
| ***10.5*** *Secure audit trails so they cannot be altered* |
| ***10.5.1*** *Limit viewing of audit trails to those with a job-related need* |
| ***10.5.2*** *Protect audit trails from unauthorized modifications* |
| ***10.5.3*** *Promptly back up audit trail files to a centralized log server or media that is difficult to alter* |
| ***10.5.4*** *Write logs for external-facing technologies onto a secure, centralized, internal log server or media device* |
| ***10.5.5*** *Use file-integrity monitoring or change-detection software on logs to ensure that existing log data cannot be changed without generating alerts (although new data being added should not cause an alert).* |

There are two key reasons for leveraging a central repository for log data storage. First, is for the long-term storage of log data for historical purposes. In the event of a breach, access to the original logs (or unmodified copies of the original logs) is critical to determining the root cause of the breach as well as identifying the potential culprit(s). Centralizing of logs simplifies the management of those logs.

The second reason for having a central repository for the storage of logs is so that operations can be performed on the logs in order to extract the information needed to meet log-monitoring requirements. Strict protection mechanisms, including those specified in Requirement 10.5, are necessary. Depending upon the tools and technologies in use to process log data, it may also be necessary to create a second repository where operations can be performed on the logs in order to extract relevant event information, separate from the long-term storage of the original logs.

## Transport Logs to the Centralized Repository

How frequently logs are transported to the central repository is an important consideration as well. It must be updated as close to real-time as possible. The criticality of the system may also factor into the frequency with which logs are polled or transported to the central repository, giving priority to the most critical systems first. For example, an authentication server within the CDE may have its log data harvested every 10-15 seconds, while the log data of an internal storage management console may only be harvested every 60 seconds.

## Prepare Log Data for Processing

[COMPANY NAME] need to implement mechanisms or processes to prepare the log data to be processed and analyzed as quickly (and as accurately) as possible. One way to achieve this is to filter (or parse) the logs to focus on only the information that is needed to satisfy operational, security, and compliance needs for log monitoring.

For efficient filtering to take place, log data needs to be converted to a common format such that filtering mechanisms can be applied quickly and consistently. Trying to process and analyze data in different formats and in different locations creates inefficiencies and often requires multiple, redundant processes or mechanisms in order to extract relevant security data.

After log data has been normalized, [COMPANY NAME] need to parse or filter the log data for information related to the specific event messages [COMPANY NAME] have identified. Filtered events should then be captured into a single repository that represents all events that have occurred on all systems. Centralization of relevant event messages allows for faster processing of event data as well as more complex operations like event correlation.

# Performing Effective Log Monitoring

Effective log monitoring is not a tool or a technology, but rather a process that requires continuous improvement. The key activities associated with that process are described in the following Figure.

**Log-Monitoring Process Cycle**



Continuous improvement of the log-monitoring process defined above is achieved through the implementation of its key activities as part of a virtual cycle. Each individual activity provides input into subsequent activities, which, in turn, feedback into the consideration of log-monitoring requirements. Additional inputs are received from parallel or supporting activities such as the risk assessment and PCI DSS scoping exercises.

## Collect and Analyze Activity Data

The key to effective analysis is to look at log data within a specific time frame. Keep the time window manageable. Too large of a population will exhaust the limited of available resources to perform analysis. Too small of a population may not reflect some activity persistent in the environment. Appropriate time frames will differ from organization to organization and will be somewhat dependent on the number of systems in the environment and the amount of log data generated by those systems.

## Establish a Baseline

#### Look for Patterns

Activity patterns can be good indicators of normal as well as abnormal activity. Log data analysis is intended, in part, to help [COMPANY NAME] differentiate between normal and abnormal behaviour. If [COMPANY NAME] have the ability to sort or filter log data by certain data fields within the centralized event repository, [COMPANY NAME] may want to start by sorting activity by Source IP/port and Destination IP/port. Look at the frequency in which certain systems are accessed. Also, look for common activities that take place and then factor in where those activities originate and the systems that are targeted by those activities. Some key metrics to consider include:

|  |  |
| --- | --- |
| * Avg # of user logins to a specific IP per day
 | * Avg bytes of data transferred per destination IP per day
 |
| * Avg # of user logins from a specific IP per day
 | * Avg # failed logins per user per day
 |
| * Avg # of times a specific IP/port is hit per day
 | * Avg # failed logins per user per day per source IP
 |
| * Avg # of user logins during off hours per host per day
 | * Avg # of times processes stopped/started per day
 |
| * Avg bytes of data transferred per source IP per day
 | * Avg # error messages per system per day
 |
| * Typical geographical region data is sent to
 | * Avg # of log messages per day
 |

In the case of using a SIEM or other data analytics tool, activity summary reports are a great source for this kind of information. Scripting languages can be used to parse log data to obtain this type of information as well. The context in which certain events occurred must be considered. For example, multiple invalid logon attempts using an administrator account followed by a successful logon using the same account is certainly one pattern that may warrant investigation.

Additionally, the creation of a non-admin user account followed immediately by admin-level privilege escalation is another example of activity patterns that may reflect potentially malicious behaviour. The ability to identify these types of patterns depends greatly on the tools and methods used to parse log data. Leveraging SIEM or SOC services will likely find it much easier to identify these types of patterns.

#### Consider Other Data to Baseline

In addition to activity captured within logs, it may also be helpful to capture some baseline information about the operating environment in general. Systems that serve a common function (such as webservers) should have almost identical configurations and operating characteristics. The processes running on those servers should be consistent across all like systems. The same is true for all users authorized to access those systems.

Any activity or circumstances that extend or contradict the baseline characteristics of any system should be investigated. Examples of potentially suspicious activity include the startup of a new process uncommon to those systems. A new FTP daemon running on a database server might be cause for concern. The creation of a new administrative user on a single system within a collection of like systems is another example of activity that has potentially malicious intentions.

#### Define Common Activity Constraints and Rules

For those activities that are common, [COMPANY NAME] need to determine at which point those activities become suspicious. In many cases, activities become suspicious if the frequency in which they occur exceeds average frequencies. Using frequency to define activity thresholds as indicators of suspicious activity is one good method for differentiating between normal and potentially abnormal activity.

Time and location constraints should also be considered. For instance, changes to key systems—such as the shutdown of audit logging mechanisms—outside of well-defined maintenance windows could be an indication of potentially malicious behaviour. Additionally, activity originating from an IP address located in region of the world where the organization has no business interests or personnel might also indicate suspicious behaviour.

## Configure Automated Alerts

[COMPANY NAME] need to create automated alerts to notify appropriate personnel of activity or behaviour that exceeds those baselines. [COMPANY NAME] will also want appropriate log monitoring personnel to be alerted of any occurrence of certain events, not just those that exceed baselines.

Automation is a critical component of the notification process and the overarching log monitoring process as well. For logs to be useful in the defence of information assets, they must be monitored and analyzed—in as close to real-time as possible—so that attacks can be detected quickly and appropriate countermeasures deployed to augment existing defences when and where necessary. Without automated alerting mechanisms, it is almost impossible to identify and alert about such events in near real-time. And without near real-time notifications, the ability to deploy countermeasures to protect information assets before they can be compromised is greatly diminished.

Despite the availability of point solutions, centralization of alerting mechanisms is preferred. Centralized alerting simplifies alert management and allows for more complex rules and alerts to be created to take into account how events may transcend multiple systems (e.g. event correlation).

## Respond to Alerts

When alerts are generated, they must be responded to in a timely manner in order to be useful. The whole point of log monitoring is to detect malicious behaviour *before* it becomes a major issue. PCI DSS Requirements 12.5.2, 10.6.3, and 12.10.3 support the need for timely response to events by requiring that personnel be assigned responsibility for monitoring and analyzing security alerts and information; that exceptions and anomalies identified during the review be followed up on; and that personnel be made available on a 24x7 basis to respond to such events, or this responsibility needs to be outsourced to 24/7 SOC service provider.

However, in addition to ensuring they are followed upon in a timely manner, alerts must also be based on relevant information, meaning they must pertain directly to the risks and concerns of the company. Alerts created based on non-essential or irrelevant events are just noisemakers and can be detrimental to the overall efficiency and effectiveness of the log-monitoring program. Alerts should only be configured on events that *require* follow up. In other words, alerts should only be created for those events that the company deems important enough to warrant an active (and timely) response. Alerts based in “informational” events serve no practical purpose other than to consume resources that should otherwise be allocated to more critical activities.

## Validate Events

Upon notification that a suspicious event has occurred, the context in which that event occurred needs to be analyzed to confirm whether or not the event indeed represents abnormal or malicious activity. This may require that other data or events related to the system that generated the original event message be analyzed. [COMPANY NAME] should look for similar activity being performed on the system in question.

If the initial event that triggered the validation process is associated with a particular user, [COMPANY NAME] should look at other activities associated with that up to and around the time frame in which the initial event occurred. [COMPANY NAME] should also look at what other non-user associated activities may have been going on at the time the initial event was generated, such as system error messages or system-level processes that may have been stopped/started in and around the same time frame.

Event validation may also require that events from other systems also be analyzed to correlate event activity across systems. This is called “event correlation.” Event correlation is a complex exercise that, while still technically possible to do manually, generally requires the use of third-party tools and solutions to be done effectively. Additionally, third-party tools and solutions may employ other methods, such as the use of signatures or automated heuristic analysis mechanisms to help confirm the occurrence of abnormal or malicious activities.

## Respond to Incidents

When event activity is confirmed to be—or associated with—known malicious activity, appropriate countermeasures need to be deployed as quickly as possible to isolate or stop the activity and address any potential residual effects or risks presented by the occurrence of that activity. Rapid response requires that procedures be defined in advance of the incident such that any delays in deploying appropriate countermeasures can be minimized.

PCI DSS Requirement 12.10.5 requires [COMPANY NAME] to implement incident response procedures to respond to alerts from security monitoring systems. PCI DSS Requirement 12.10 covers the need to have such incident response procedures formally documented (in advance), and Requirement 12.5.3 requires appropriate personnel be assigned responsibility for ensuring the timely and effective handling of security incidents to meet forensics cyber review requirements.

This section should be read in conjunction with [COMPANY NAME] Incident Response Management Plan.

## Collect and Analyze Incident Data

Incident feedback is a critical component of an effective log-monitoring program. Any lessons learned from the handling and response to information security events should be formally documented and used as input when it is time to conduct a periodic review of incident response processes. PCI DSS Requirement

12.10.6 explicitly requires such improvements be made for incident response processes, and the same should be made applicable to other log monitoring processes as well. Any gaps or areas of improvement should be identified, and recommendations for addressing those gaps should be documented.

Metrics should also be developed to qualify and/or quantify program performance. Some sample metrics that could be used to describe the effectiveness of the log-monitoring program includes the following:

|  |  |
| --- | --- |
| * % systems with logging enabled
 | * Avg # of confirmed events per day/week/month
 |
| * % systems with alerting enabled
 | * # of unknown events per day/week/month
 |
| * Top 10 events detected
 | * # of baseline violations per day/week/month
 |
| * Top 10 systems affected (destination IP)
 | * # of detected events vs. # of alerts generated
 |
| * Top 10 activity sources (source IP)
 | * # of false positives
 |
| * Top 10 users associated with confirmed events
 | * # incidents vs. # of resolved incidents
 |
| * Avg # of alerts per system per day/week/month
 | * Avg response delay (from detection to response) in minutes
 |

## Report on Results

Management must be kept aware of the performance of the log-monitoring program to ensure the program continues to receive the oversight and resources necessary to continue to operate effectively. Many third-party monitoring tools and solutions come pre-packaged with report modules that provide log-monitoring performance data, including metrics.

Where third-party tools are unavailable, scripts can be written to extract similar information from log data repositories. Reports and/or presentations should be created to describe the program’s performance to management using metrics as well as other quantitative and qualitative measures. These reports then become a primary source of input when log-monitoring requirements and use cases are re-evaluated to ensure ongoing alignment with updated risk analysis.

## Perform Periodic Program Reviews

The ongoing effectiveness of any log-monitoring program is contingent on its ability to keep up with changing business risks and needs. Therefore, to keep the log-monitoring program operating effectively and efficiently, log-monitoring requirements should be frequently re-evaluated and take into consideration the latest risk analysis information as well as overall program performance information to identify any potential gaps.

It is possible, even likely, that updated risk analysis and log monitoring performance data will uncover new events and activities that require monitoring.

Periodic Review of log-monitoring requirements should also be performed. Security threats and protection mechanisms evolve rapidly. If log-monitoring requirements do not keep up with the evolving risk landscape, detection mechanisms may not pick up on new attacks or events that are potential indicators of new attacks.

Periodic reviews of log-monitoring requirements should be performed using intelligence from numerous sources. One such source is the results from the annual risk assessment. Additional sources of intelligence should include log monitoring program performance metrics as well as intelligence information from external sources. The use of third-party tools and solutions, particularly those solutions with advanced data analytics and “machine learning” capabilities, can greatly help with the performance of periodic program reviews and ongoing threat identification and analysis.

# Applying Effective Log Monitoring

While suspicious events and activity need to be monitored in near real-time, so should the process and controls that are employed to protect information assets from intruders.

## Business-as-Usual Activities

PCI DSS introduced the concept of business-as-usual (BAU) activities as a strategy for ensuring the ongoing effectiveness of PCI DSS security controls. The primary objective of implementing security into BAU activities is to detect the failure of any critical security controls. Some security controls are systems themselves, such as firewalls, intrusion-detection/intrusion-prevention systems, anti-malware, and access control mechanisms. Other security controls are not systems, and detecting potential failures in those controls isn’t as simple as monitoring the up/down status of a system or process.

The ability to track the status and effectiveness of security controls is yet another benefit of an effective log monitoring program. The table below provides a sample of information that can be obtained from log data to monitor the state and effectiveness of PCI DSS security controls:

|  |  |  |
| --- | --- | --- |
| **Req. No.** | **Requirement** | **Potential Verification Method** |
| 1.3 | Prohibit direct public access between the Internet and any system component in the CDE. | Set up alerts/reports to identify activity between IP addresses located in the CDE and IP address ranges not employed by the organization. |
| 2.2.2 | Enable only necessary services, protocols, daemons, etc. as required for the function of the system. | Set up alerts/reports to detect when non-approved services, protocols, and daemons are activated. |
| 3.6.4 | Replace cryptographic keys that have reached the end of their crypto period. | Set up alerts/reports to detect cryptographic keys that have exceeded their defined crypto period. |
| 5.2 | Ensure that all anti-virus mechanisms are kept current. | Set up alerts/reports to detect outdated anti-virus signatures. |
| 6.2 | Install critical security patches within one month of release. | Set up alerts/reports to identify systems that have outdated versions of software. |
| 7.1 | Limit access to system components and cardholder data to only those individuals whose job requires such access | Set up alerts/reports to identify when users are assigned to groups with access to cardholder data. |
| 8.1.5 | Monitor activity of user IDs used by vendors | Set up alerts/reports to identify when vendor user IDs are logged on/off. |
| 9.1.1 | Use access control mechanisms to monitor individual physical access to sensitive areas. | Set up alerts to identify when users access sensitive areas. |

Log-monitoring functions themselves can be “monitored” as well to ensure their ongoing effectiveness. One way to do this is to review system status/performance information. For example, system-level processes associated with logging mechanisms should be monitored (using alerts) to detect whether or not those processes are actively running. This is slightly different than what is required by PCI DSS Requirement 10.2.6.

Active monitoring or polling of a process’ status (e.g., an active PID) can help detect—in near real-time—when such processes fail or are shut down. Additionally, it may be possible to create and automate “fake” events to check the status of detection and alerting mechanisms. One example of this approach in action is to run a scheduled task to periodically (for example, every 1-5 minutes) generate a log message that can be used to verify event detection and alerting mechanisms are operating effectively.

Other methods may include leveraging some of the advanced capabilities of third-party tools and solutions, including security information and event management systems and advanced data analytics solutions.

# Log Monitoring RACI Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Cyber and Information Security** | **Security Operations Center** | **IT Department** |
| Manage Overall SIEM Solution | A | R |   |
| Vendor Escalation Requests | R/A | I |   |
| Vendor Management | R/A | I |   |
| Internal SIEM Service Escalations | R/A |   |   |
| Develop and maintain logging configuration standards | R/A | C | C |
| Test logging configuration standards | R/A | C | C |
| Support SIEM Application |   | R/A |   |
| Install SIEM agent | I |  | R/A |
| Manage Parsers |   | R/A |   |
| Prioritize parsing issues | C | R/A | C |
| Maintain Data Source Integration | I | R/A | C |
| Support SIEM Operating System | I | C | R/A |
| Support SIEM Servers/Hardware | I | C | R/A |
| Support SIEM Storage | I | C | R/A |
| Respond to SIEM Application alerts | I | R/A |   |
| Patch SIEM Application | I | R/A | C |
| Manage certificates | I | R/A | I |
| Create New Rules | I | R/A | C |
| Update Rules | I | R/A | C |
| Manage intake of tuning requests | I | R/A | C |
| Plan New Use Case Development | A | R | C |
| Create Reports | A | R | C |
| Update Reports | A | R | C |
| Develop and Tune Correlation Rules | A | R | C |
| Develop and Tune Reports | A | R | C |
| Manage User Accounts | I/C | R/A |   |

# Appendix A: List of Mandatory Alerts

|  |  |  |
| --- | --- | --- |
| **PCI DSS Reference** | **Coverage** | **[COMPANY NAME] Asset Types** |
| **Windo ws Server 2008** | **Windo ws Server 2016** | **Windo ws Server 2012 R2** | **Linux** | **AIX** | **Network Switch** | **Network Router** | **Check point Firewall** | **For tiga te Firewall** | **Juni per Firewall** |
| **PCI 10.2.x****(Capture log events)** | Alert: User Access to Sensitive Data | X | X | X | X | X | X | X | X | X | X |
| Alert: Actions taken by root or admin access | X | X | X | X | X | X | X | X | X | X |
| Alert: Failed login attempts | X | X | X | X | X | X | X | X | X | X |
| Alert: Access to audit trails | X | X | X | X | X | X | X | X | X | X |
| Alert: Privilege elevation / escalation by users | X | X | X | X | X | X | X | X | X | X |
| Alert: Changes, creation and deletions to root or administrative accounts. | X | X | X | X | X | X | X | X | X | X |
| Alert: Initialization, stopping, or pausing of the audit logs | X | X | X | X | X | X | X | X | X | X |
| Alert: Creation and deletion of system-level objects. | X | X | X | X | X | X | X | X | X | X |
| Alert: List of malware infected systems | X | X | X | X | X | X | X | X | X | X |
| **PCI 10.5 (Secure Audit Trails)** | Maintain integrity of logs on centralised log repository | X | X | X | X | X | X | X | X | X | X |
| **PCI 10.6.1 (Daily log review report)** | Creation of daily log review report | X | X | X | X | X | X | X | X | X | X |
| **PCI 10.7 (Log retention)** | Centralized Log Management | X | X | X | X | X | X | X | X | X | X |
| On-demand supply logs for specific period. | X | X | X | X | X | X | X | X | X | X |
| **PCI 10.8, 12.10.5****(failures of criticalsecurity control****systems)** | Monitor surge in log traffic | Consolidated log traffic surge monitoring |
| Monitoring of Disconnections for log systems | X | X | X | X | X | X | X | X | X | X |
| Failures of critical security control systems | X | X | X | X | X | X | X | X | X | X |
| **PCI 11.5 (FIM)** | File Integrity Monitoring | X | X | X |  |  |  |  |  |  |  |