|  |  |  |
| --- | --- | --- |
|  | |  |
|  | | Requirements Document |
|  | | Security Monitor subsystem requirements |
|  | Requirement Document  **Version: 1.3**  **Last Revised:** November 19, 2010  **Author: Anatoliy Lokshin** | |

Table of Contents

[Feature Overview 4](#_Toc277857942)

[1 Terminology 4](#_Toc277857943)

[2 Product Requirements 4](#_Toc277857944)

[2.1 Functional Area 6](#_Toc277857945)

Document Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Revision No.** | **Date** | **Description of Change** |
| **Anatoliy Lokshin** | 1.0 | 09/08/2010 | First revision |
| **Anatoliy Lokshin** | 1.1 | 09/10/2010 | Redundant requirements removed. Corrected terminology. |
| **Anatoliy Lokshin** | 1.2 | 11/18/2010 | Added one extra event parameter – input data size (R-102). Added lacking policy engine rules to R-112. Added requirements R-114 and R115. |
| **Anatoliy Lokshin** | 1.3 | 09/23/2011 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Approvers

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Date Approved** |
| **Kyle Quest** | Architect |  |
| **Julia Kuchmai** | QA Representative | 11/18/2010 |
|  |  |  |

Reviewers

| **Name** | **Title** | **Date Reviewed** |
| --- | --- | --- |
| **Kyle Quest** | Architect |  |
| **Julia Kuchmai** | QA Representative | 11/18/2010 |
|  |  |  |

References

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Author** | **Location** |
| |  | | --- | | MetraTech Security Framework Specification | | Kyle Quest | http://seceng.metratech.com/gf/download/docmanfileversion/20/78/MtSecurityFrameworkSpec.doc |
|  |  |  |

# Feature Overview

The Security Monitor subsystem is responsible for providing the infrastructure for collecting and processing security related application events to detect suspicious and malicious activities that can’t be detected when individual data objects or application operations are analyzed. It can also be used to provide policy-based response actions for security exceptions encountered during individual data object or application operation processing.

# Terminology

Requirements that are labeled with an R (such as R-100) are Mandatory (or Required) for the targeted release.

Requirements that are labeled with an O (such as O-100) are Optional for the targeted release.

It is left to Engineering’s discretion on whether these can be met for the targeted release but it is imperative that Engineering take these into consideration when making any design or infrastructure decisions. This holds good for any requirement that might be listed under “Future Requirements” section as well.

**(R-1)** It is a Mandatory requirement that the infrastructure developed shall support the ‘feasibility’ of all feature requirements listed in the document – though the delivery of the features themselves could be scheduled for a later time.

# Product Requirements

Security Monitor subsystem executes two main functions - security event recording and security event analyses.

Event recording allows collecting security events for monitoring by an operator and for subsequent analyses.

The analysis allows making a decision about an action in response on an event without operator’s intrusion (automatically). The making a decision is based on configurable lists of rules (Security Monitor policies).

MetraTech applications can interact with the Security Monitor subsystem either by explicitly creating security events and reporting them through the Security Monitor subsystem API or they can simply invoke the ‘Report’ extension method on exception objects thrown in the application code. If a given exception object was generated by one of the Security Framework subsystems, as a result of a security violation, and then invoking its “Report” extension method will automatically generate a Security Monitor subsystem security event.

The Security Monitor subsystem capabilities are configured using the Security Monitor Policies that control event collection conditions, thresholds, and response actions.

The Security Monitor policies can have the following actions:

* Block operation – to block an individual user operation.
* Block user – to block a specific user.
* Block address – to block a specific IP address.
* Log – to log an application security event.
* Redirect Operation – to redirect an operation to an alternative location.
* Redirect User – to redirect all operations from a specific user to an alternative location.
* Logout user – to logout the current session user.
* User security warning – to send back a visual security warning to the current session user.
* Application administrator notification – to send an admin notification.
* Session parameter change – to change one of the current session parameters (e.g., change caching settings, change timeout settings, change security thresholds, etc).

MetraTech application sets a specific handler for each action type. The handler is specified via the security framework configuration.

The security event categories represented by different event types in the Security Monitor subsystem can be used to report the following types of information:

* HTTP Request Events (Unexpected HTTP Commands; Attempts to Invoke Unsupported HTTP Methods; GET When Expecting POST; POST When Expecting GET; duplicate request parameters; unexpected number of request parameters; bad request parameter names; malformed request parameter values).
* HTTP Response Events (Unexpected response headers; unexpected response data).
* Authentication Events (Use of multiple usernames; Multiple failed passwords; High rate of login attempts; Unexpected quantity of characters in username; Unexpected quantity of characters in password; Unexpected types of characters in username; Unexpected types of characters in password; Providing only the username; Providing only the password).
* Access Control Events (Modifying URL arguments within a GET for Direct Object Access attempts; Modifying parameters within a POST for Direct Object Access attempts; Force browsing attempts; Evading presentation access control through custom posts).
* Session Events (User login; User logout; User operation executed; Modifying existing cookies; Adding new cookies; Deleting existing cookies; Substituting another user's valid session ID or cookie; Source IP address changes during session).
* Ingress Data Processing Events (Cross Site Scripting Attempt; Violations of implemented White Lists; Double encoded characters; Unexpected encoding used; Blacklist inspection for common SQL injection values; NULL byte character in file request; Carriage Return or Line Feed character in file request).
* Egress Data Processing Events (Detect abnormal quantity of returned records; Detect system shell command result information).
* File I/O Events (Detect large individual files; Detect large number of file uploads).
* User Trend Events (enlisted below).
* Application Trend Events (enlisted below).

The session and access control events also provide the foundation for the user and application trend monitoring performed by the Security Monitor subsystem. Other application generated security events provide additional information for the trend monitoring.

User trend monitoring tracks the following statistics (can be implemented in future versions):

* Irregular use of application for different users and a common baseline across all users.
* Speed of application use for different users and a common baseline across all users.
* Frequency of site use for different users and a common baseline across all users.
* Frequency of feature use for different users and a common baseline across all users.
* Number of data processing security events for different users and a common baseline.

Application trend monitoring tacks the following statistics:

* Number of logouts.
* Number of logins.
* Number of logins vs. the number of explicit logouts.
* Average transaction/operation count (as a single number and as a range window).
* Normal deviation from a baseline logout counts (to handle short-lived spikes in normal user activity).
* Normal deviation from a baseline transaction/operation counts (to handle short-lived spikes in normal user activity).

The Security Monitor policy rules can be created for each security event category including the internal user and application trend events.

## Functional Area

| Requirement | Story | Constraints |
| --- | --- | --- |
| R-100 – Mandatory requirement | Security Monitor subsystem (Subsystem) provides an API for creating and handling of the security events. |  |
| R-101 – Mandatory requirement | Subsystem provides a simplified API for handling exceptions as the security events. |  |
| R-102 – Mandatory requirement | Subsystem records each handled event. The records will be available to analyses and contain the following information about each event:   * Event category (from the list above) * Event source type (subsystem and its category) * Date and time when the event happened * Operation path (a URL of the page in WEB app) * Server name * Text description of an event * Client IP address * User name (identity) * App session ID * Some info on the user’s client app (WEB browser) * Stack trace * Input data size (for large data, such as files, that we don’t want to record completely) |  |
| R-103 – Mandatory requirement | Subsystem provides the possibility to filter events before recording. The filtering is configurable and provides the following filter types: by security event type (source subsystem and subsystem’s category), the event’s category (from the lists above), and custom filter implemented via delegate – it can be supplied by MetraTech application. |  |
| R-104 – Mandatory requirement | Subsystem can read recorded events at least from one of the storage to analyze it. |  |
| R-105 – Mandatory requirement | Subsystem provides an infrastructure for configuring policies for handling of the security events. The policies are configurable via the Security Framework configuration. |  |
| R-106 – Mandatory requirement | Subsystem allows specifying of the handler for each of the type of a responding action. |  |
| R-107 – Mandatory requirement | An action handler represents an implementation of the ISecurityPolicyActionHandler interface and is provided by the MetraTech application. |  |
| R-108 – Mandatory requirement | Each type of an action has its own attributes those are configurable for each policy. |  |
| R-109 – Mandatory requirement | Each policy can have multiple actions simultaneously. |  |
| R-110 – Mandatory requirement | The multiple policies can be configured simultaneously. |  |
| R-111 – Mandatory requirement | Each policy represents a list of rules. The rules are connected with AND operation. |  |
| R-112 – Mandatory requirement | There would be the following types of the rules:   * EventSourceRule – checks if the event belongs to the specified type (source subsystem and subsystem’s category) * EventTypeRule – checks if the event belongs to the specified category (from the lists above) * EventRepeatThresholdRule – checks if the event of the same type happened some number of times in the current user session * EventTimespanThresholdRule – checks if the event of the same type happened in the specified time window either for entire app or for the current user only * IPChangeRule – checks if the client’s IP address changed in the current session * ActionFrequencyRule – checks how much time passed since an action of the especial type was recommended * EventRatioThresholdRule – checks a ratio between a number of events from one specific source to a number of actions from another source in the specified time window for entire app or for the current user only * InputDataSizeRule – checks an input data size |  |
| R-113 – Mandatory requirement | Each type of the action has some specific configurable properties. Actually we don’t have a complete list of the properties and guided by a common sense. |  |
| R-114 – Mandatory requirement | Subsystem provides the possibility to record each recommended responding action. |  |
| R-115 – Mandatory requirement | Subsystem provides a possibility to track application and user trend events. An especial virtual subsystem – “Statistics” – with a set of categories must be registered for this purpose.  The following categories will be added:   * UserLogin * UserLogout * FeatureUsage * TransactionUsage * FileUpload * IrregularUsage |  |